Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

William Emanuel Skreien Kaaby
Stig André Rosenlund

Structured Object Detection

Detecting Objects in Images Through a Structural Model of Constituent Object Parts

Master's thesis in Datateknologi (MIDT)
Supervisor: Rudolf Mester
Co-supervisor: Annette Stahl
January 2022

NTNU
Kunnskap for en bedre verden

William Emanuel Skreien Kaaby
Stig André Rosenlund

Structured Object Detection

Detecting Objects in Images Through a Structural Model of Constituent Object Parts

Master's thesis in Datateknologi (MIDT)
Supervisor: Rudolf Mester
Co-supervisor: Annette Stahl
January 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

Abstract

CNNs are frequently used to solve visual tasks such as image classification or object detection. However, explaining why the internal structures produce a particular hypothesis remains challenging. This thesis aims to explore and develop a system capable of performing object detection while increasing the explainability of the network's hypotheses. The majority of work is oriented toward the classification aspect of object detection, with a lesser focus on localization. The increased explainability is achieved by leveraging the compositionality between an object and its constituent parts. To that end, we have endeavored to develop two subsystems that target different tasks associated with using compositionality: part detectors that can detect the constituent parts of an object and a joint structure system that takes the part detections as inputs and outputs a class hypothesis.

Our approach represents, to our knowledge, a novel method of creating part detectors. The part detectors are created from the filters of a pre-trained VGG16. The selection of filters is made by looking at the activation intensity distributions that the filters produce from instances of different classes. We select those filters that have the highest distribution separation. The selection process was developed based on previous research on network dissection. We fit a logistic regressor to the filter outputs, which produces a probability map that contains the probability of a part being present at each location of an input.

When evaluating our part detectors using a Bag of Words classification model based on part occurrences, we find that we can achieve comparable performance to VGG16 despite using only a subset of filters from VGG16. In other words, we prune the base model while maintaining good performance. Furthermore, using probabilistic part detectors means that our system has acquired increased explainability compared to VGG16. Finally, we conducted some preliminary experiments on object detection using the developed classification systems to localize objects.

Sammendrag

CNNer er ofte brukt til å løse visuelle oppgaver slik som bildeklassifikasjon og objektdeteksjon. Imidlertid er det utfordrende å forklare hvorfor de interne strukturene fører til en bestemt hypotese. Denne avhandlingen har som mål å utforske og utvikle et system som kan utføre objektdeteksjon i tillegg til å øke tolkbarheten av nettverkets hypoteser. Hoveddelen av arbeidet er rettet mot klassifikasjonsaspektet ved objektdeteksjon med et mindre fokus på lokalisering. Denne økte tolkbarheten oppnås ved å utnytte sammensetningen mellom et objekt og dets tilhørende deler. Med dette i mente har vi utviklet to ulike undersystemer som utfører ulike oppgaver tilknyttet bruken av objektets delsammensetning: deldetektorer som detekterer et objekts bestanddeler og et fellesstruktursystem som tar inn deldeteksjonene og gir ut en klassehypotese.

Vår tilnærming representerer, så vidt vi vet, en ny metode for å lage deldetektorer. Deldetektorene er laget fra filtrene til en forhåndstrent VGG16. Valget av filtre er gjort ved å se på distribusjonen av aktiveringsintensiteter som er produsert av filtrene gitt ulike klasseinstanser. Vi velger de filtrene som har størst separasjon av distribusjoner. Valgprosessen er utviklet på bakgrunn av eksisterende forskning på nettverksdisseksjon. Vi tilpasser en logistisk regresjonsmodell til filtrenes utdata, som igjen produserer et sannsynlighetskart over sannsynligheten for at en del er i en gitt posisjon.

Gjennom en evaluering av deldetektorene, som ble gjort ved hjelp av en Bag of Words klassifikasjonsmodell basert på delforekomster, oppnår vi en tilsvarende ytelse som VGG16 til tross for at vi kun bruker en delmengde av filtrene fra VGG16. Med andre ord reduseres grunnmodellen samtidig som vi opprettholder god ytelse. Dessuten fører bruken av probabilistiske deldetektorer til at vårt system har økt tolkbarhet sammenlignet med VGG16. Avslutningsvis utførte vi noen innledende forsøk på objekt deteksjon ved å bruke de utviklede klassifikasjonssystemene til å lokalisere objekter.

Preface And Acknowledgments

This master thesis was conducted as part of the master's degree program "Computer Science" at the Norwegian University of Science and Technology. We thank our supervisor Rudolf Mester for his continuously valuable feedback during the pre-project and thesis work. We must commend him for his rapid responses to queries and his general enthusiasm for the work. We also thank Ernesto Lopez for allowing us to use the boat dataset he created.

Contents

Chapter 1

Thesis Outline

This thesis is about detecting objects in images through a structural model of constituent object parts. Object detection is the task of localizing and categorizing objects in an image. Since the thesis aims to contribute to object detection by building part-based detectors, the tasks can be said to be two-fold. The first task is to detect where parts are located and what parts are found, and the second is to determine if these parts indicate an object's presence.

We believe that this structured form of object detection has inherent benefits over monolithic, end-to-end, object detection approaches. One potential benefit is the possibility of generating an explanation of why the system generates a specific detection hypothesis, which is crucial for applying a detection system to failure-sensitive environments. Another potential benefit is increased robustness to specific visual problems like occlusion and context sensitivity. A last potential benefit is the possibility for rule generalization, i.e., learning the part configuration of one object and extending this to also handle unfamiliar objects.

It should be noted that the idea of structured detection through parts is not new and was, in fact, one of the most advanced forms of object detection before the advent of the deep learning revolution[1]. Recently a trend has appeared of adapting the idea of structural models to the context of deep learning. We wish to contribute and intend our approach to automatically learn which parts are essential for detecting a particular class of images through leveraging the strengths of deep learning.

Our approach requires two separate systems; one that can generate part detections and another that can detect objects by reasoning about the structural composition of the detected parts. In order to discover how such systems can be created, we perform multiple literature searches, which are divided across multiple chapters. A high-level overview of related work is given in chapter 3. Additionally, we start both chapter 4 and chapter 5 with a literature review as well, with the mentioned chapters being focused explicitly on part detectors and methodologies to combine part detections, respectively.

[1] This refers to the part based model [FMR08], which is talked about further in chapter 3

In chapter 4 we find a wide range of approaches related to generating part detections, and due to this we also re-implement specific papers to test their viability. We implement a total of four different part detector approaches:

A) Feature vectors as part detectors, section 4.4

B) Feature map pattern mining, section 4.5

C) Interpretable Convolutional Neural Network, section 4.6

D) Filter-Based part detectors, section 4.7

Our unique contribution to generating part detections is presented in section 4.7. Our approach focuses on dissecting a pre-trained CNN to find part detector filters, inspired by papers from Bau et al. [Bau+17; Bau+20]. To our knowledge, the proposed method of selecting and applying the filters represents a novel approach to creating part detectors. Bau et al.'s papers focus on mapping concepts such as objects, parts, textures, and colors to filters using segmentation masks. Our work differs from Bau et al. since we only need image-level labeling and exclusively look for parts. Our work also further expands on how these filters can be used in decisions by transforming their outputs into probabilities.

In chapter 5 we also find several approaches for modeling the relationships between parts with a primary focus on spatial relationships. However, due to time constraints, we have only been able to implement three basic approaches for classification. It should be noted that our goal is detection, so we also discuss how our classifiers can be extended for detection in chapter 7.

A) Bag of words, section 5.3

B) Bag of words extended with spatial neighborhood, section 5.4

C) Bag of words extended with spatial neighborhood configuration, section 5.5

The evaluation of our classifiers requires part detections. We use the part detectors from the approach in section 4.7 for this. The results from the evaluation is found in section 6.1. Afterward, in section 6.2, we describe and evaluate different aspects of our work and make suggestions to solve the remaining issues. Finally, in section 6.3, we present the direction of future work which we believe would yield the best results. The thesis ends with chapter 8 where we restate the findings and give some closing remarks.

Due to the thesis work's sequential nature, we tried a large number of different approaches to the given problem. To avoid confusion about our work, we illustrate the relationship between the approaches from part detection and the simple classifiers in Figure 1.1. The part detector approaches A, B, and C, and the classifier approaches A, B and C are based on previous work. In other words, we did not come up with these approaches, we only investigated and evaluated them. Therefore, we consider the thesis's primary contribution to be approach D, i.e., our method of finding part detector filters and then using them in a classification process.

We would also like to mention that this thesis builds on a pre-project [KR21] which was conducted in the previous semester. A minor amount of the text and figures in this thesis originate in the pre-project. However, it has substantially changed over the past months, so we do not consider most of it citable reuse. For clarity, we will still insert footnotes in chapters containing content from the pre-project.

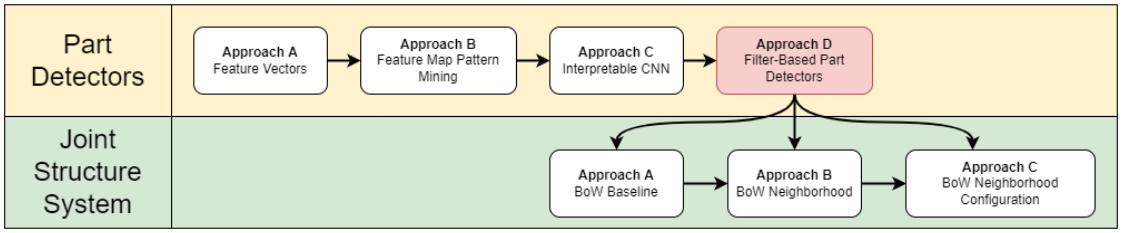


Figure 1.1: An illustration of the chronological implementation of approaches and the dependencies between them. The primary contribution of this thesis is Approach D, which is highlighted in red. The horizontal direction represents chronology, while the vertical direction represents dependency. The vertical direction takes precedence over the horizontal direction. For instance, the BoW baseline was developed after the filter-based part detectors.

Chapter 2

Introduction

[1]In this thesis, we investigate whether a *compositional architecture* based on deep learning can detect objects in images. By compositionality, we mean the concept of defining a whole in terms of its constituents using a set of rules. In the context of detecting objects in images, this means an object (the whole) can be seen as a spatial configuration (the rule) of individual parts (the constituents). We define compositional architectures as consisting of two main components, *part detectors* and a *joint-structure system*. We define a part detector as a detector that detects the constituent parts of an object and a joint-structure system as a system that combines multiple part detector outputs into an overall hypothesis.

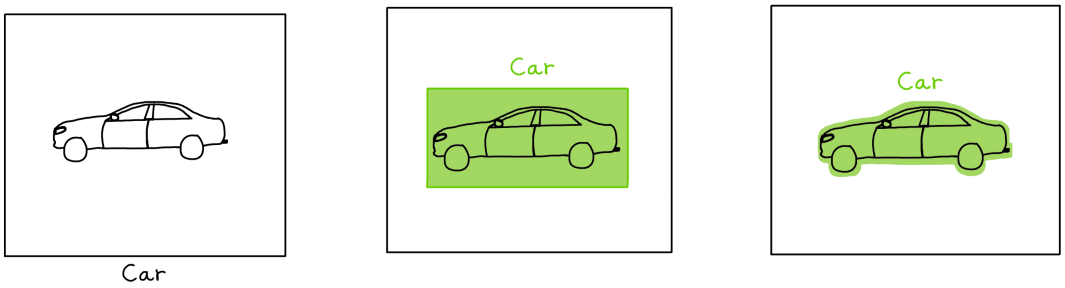
The current object detection paradigm uses deep artificial neural networks to generate object detections. Using deep networks consistently beats other approaches, but they also suffer from a lack of explainability due to their black-box[2] nature. On the other hand, a compositional architecture contrasts deep networks by having an inherent ability to be explainable by construction. In our thesis, we want to use the impressive power of deep learning while generating explainable decisions through a compositional architecture.

2.1 Visual Tasks

We focus on a specific subset of visual recognition tasks in images: classification, detection, and segmentation. These three visual tasks are illustrated in Figure 2.1. The most straightforward of the visual tasks is classification. *Image classification* is the task of assigning a category to an image, and typical applications are recognizing the presence of an object or scene in an image. When a joint-structure system performs classification, we define it as a *joint-structure classifier*. A simple illustration of a compositional architecture that performs classification can be seen in Figure 2.2. It should be noted that classification as a task is usually of limited practical utility due to two things. (1) The task does not localize objects, a common requirement in real-world applications. (2) The task has issues related to the mixing of background and object information in images, i.e., information

[1]The first paragraph of the introduction, Figure 2.2 and Figure 2.3 all has evolved from the pre-project
[2]Black-box is a term used for a system which we can only observe through its input and outputs

calculated across the entire image will contain a mixture of unrelated information and information from the object of interest.



(a) Example of classification of an image where a car is found to be present. The car is not localized in a classification task.

(b) Example of bounding box generation. A bounding box is generated on the location of an object and classified as containing a car.

(c) Example of segmentation. Each image pixel is classified as belonging to a class, which gives a mask around the car.

Figure 2.1: Example of three common visual task.

When a visual task, in addition to classification, incorporates localization, it is considered *Object detection*. It is essential to highlight that classification in the detection context is of objects at specific locations in an image, not the whole image. We define a joint-structure system that performs detection as a *joint-structure detector*. There are different forms of detection, the most simple form being localization via the computation of bounding boxes around objects in an image. A more advanced version of detection is (pixel level) segmentation[3]. *Object segmentation* is a more fine-grained form of localization in which each pixel of an image is classified, effectively generating a mask over each object's location. Other visual tasks exist, such as keypoint detection, motion prediction, pose estimation, and so on, but we will not elaborate on these further as they are outside the scope of the thesis.

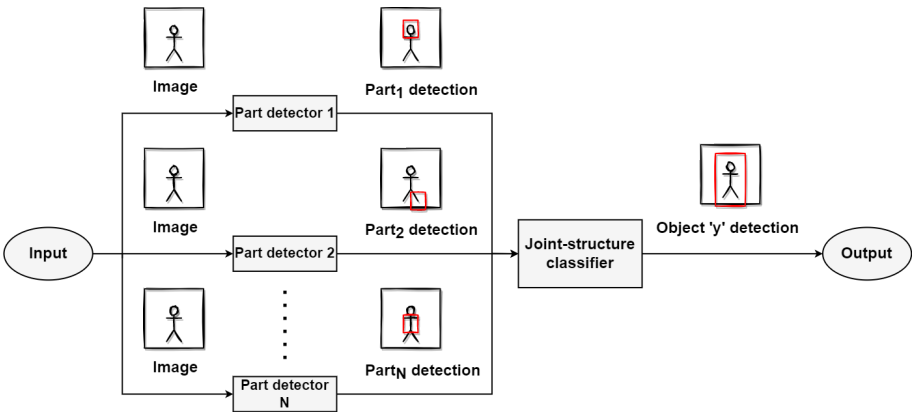


Figure 2.2: An example of a compositional architecture that performs classification. This figure is a slightly altered version of figure 1.1 from [KR21].

[3]Some works also denote the determination of bounding boxes or other approximate object masks as "segmentation", but we will use the term in the following for pixel-level segmentation.

2.2 Joint-Structure Systems

There are different ways a joint-structure system can be implemented. We present advanced approaches in chapter 5, but more straightforward approaches can also be used. For example, suppose one chooses to ignore spatial relationships between parts. In that case, one could implement a Bag of Words (BoW) approach, a group of methods based on learning to classify entities by counting tokens.

BoW methods were first applied in a natural language context for classifying the topic of documents by counting the presence of keywords. This form of classification works because texts related to specific themes often have some repeating words that can be distinguished using word histograms. For example, imagine the scenario of classifying political and scientific documents. The texts related to politics would be more likely to repeat words related to names of well-known political parties and persons. In contrast, scientific texts would be more likely to repeat certain words related to the scientific method, such as ”hypothesis” or ”probability”. The BoW methods were later applied in a visual context by considering features extracted from images as a form of visual words, which gave them the name Bag of Visual Words (BoVW). Through visual words, one can classify images in a similar way to natural language.

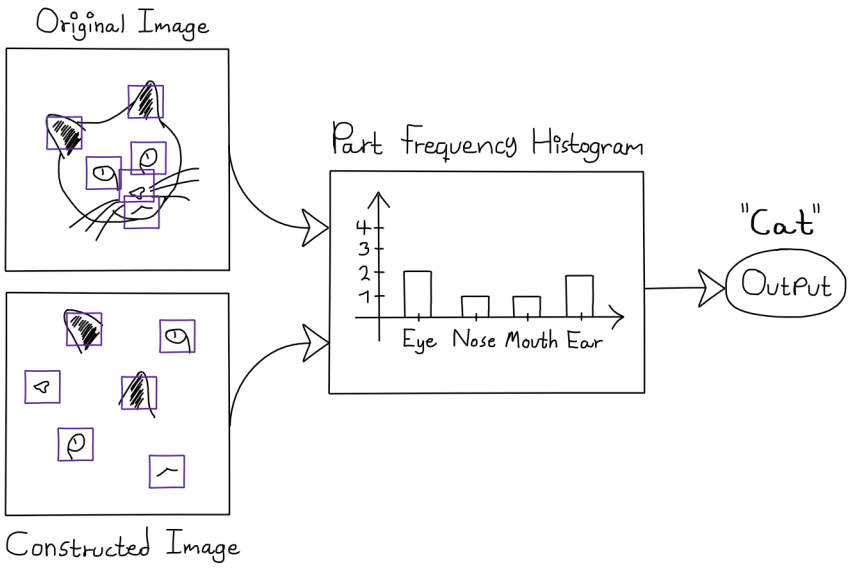


Figure 2.3: This illustration shows an issue that can arise if one only considers part counts and not the spatiality of the parts. By only considering the part counts, a random configuration of parts can be classified as an instance of some object class of interest. This illustration is based on figure 1.2 from [KR21].

One critical flaw of the BoW methods is their inability to consider the spatial arrangement of words. In the natural language case, this means it does not consider word order, while in the visual case, it does not consider the position of visual words in the image. One of the issues associated with this approach is illustrated in Figure 2.3. It should be noted that extensions to fix the spatial flaw of BoW exist. The paper by Lazebnik et al. [LSP06] for example, suggests a spatial pyramid, which is a collection of features from sub-regions of images at different scales (i.e., in a pyramid pattern, a global feature first covers the

entire image, then smaller sub-region features, and so on). This spatial extensions, and others, are described in further detail in section 5.1.2.

2.3 Background Of Object Recognition In Images

One should recognize that objects in images fundamentally can be understood as the projection of three-dimensional entities onto a plane. The projection generates a pattern representation that can only represent the visible surface of an object, which means that the shape (contours) of the projected object depends on the object's pose. Therefore, methods that solve the aforementioned visual tasks using images are constrained to operate on the features that form the image's patterns. *Features* are the fundamental structured representation of patterns in an image. In other words, features are classifiable, localizable, and quantifiable entities that describe the contents of an image. Features can range from low-level concepts such as edges to high-level concepts such as windows. Any attempt to solve visual problems has to do so by targeting the features characteristic of the relevant visual domain.

2.3.1 Hand-Crafted Features

We have already mentioned that extracting "features" is crucial for object recognition, localization, and classification. In the early days of the computer vision field, the primary paradigm was to handcraft feature extractors, but handcrafting was labor-intensive and introduced a problem of generalization. Since humans constructed features, the features are burdened with assumptions the creators made about the data to which the features were applied. These assumptions lead to a lack of generalizability of the feature extractors, which causes failure cases, e.g., some extractors were utterly reliant on finding particular features such as corners, edges, or blobs.

Some well-known examples of handcrafted features that have been used in early object detectors are SIFT [Low04], and HOG [DT05]. In general, the evaluation and selection of suitable handcrafted approaches require a deep understanding of the mathematics of pattern formation and description and knowledge of the application domain. Therefore, it is desirable to use a method that automatically adjusts to the problem challenges so that domain expertise becomes less of a concern.

2.3.2 Learned Features

The emergence of cheap computing, large public datasets, and powerful machine learning approaches produced a new avenue of approach. These developments led to a paradigm of using deep artificial neural networks. This paradigm shift meant that handcrafting was no longer a stringent necessity since features could now be automatically learned from large datasets. Furthermore, high accuracy on large datasets gave confidence regarding the generalization of the learned features. Although the new paradigm of using deep artificial neural networks outperformed previous methods, it also introduced a problem of explainability.

Since deep network features are automatically learned and encoded as high-dimensional representations, understanding what they represent is very difficult. The effect of using incomprehensible features is that the deep networks effectively function as black boxes, meaning explaining why they make certain decisions or how they process inputs is very challenging. This lack of explainability severely impacts the level of trust deep networks can be afforded in safety-critical applications, limiting their potential.

2.4 Convolutional Neural Networks

It should be noted that when we talk about deep learning and deep networks, we specifically refer to Convolutional Neural Networks (CNN). This specificity is because we operate in the context of object detection in images, and these networks excel at such tasks. In general, CNNs can be considered a group of machine learning architectures based on convolutions to extract features from input images. The architectures are usually structured as multiple layers that perform convolution, where each layer feeds into the next. We refer to the layers as L_n, where the subscript $n \in \{1, 2, ..., N\}$ denotes a specific layer. This is illustrated in Figure 2.4.

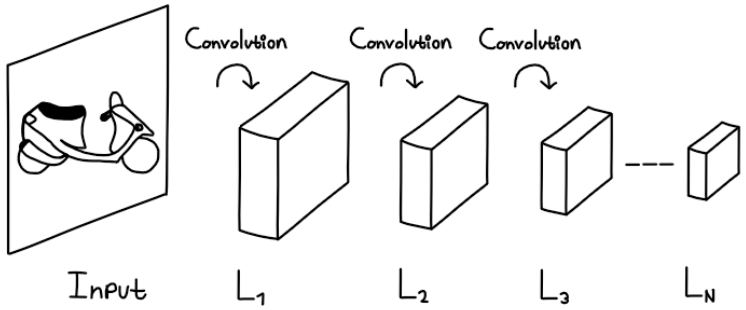


Figure 2.4: The figure shows that an input is sequentially processed by multiple convolutional layers.

2.4.1 Convolution, Kernels, And Filters

A convolution[4] is a linear operation where a *kernel* $k \in \mathbb{R}^{S \times S}$ is applied multiplicatively in an elementwise fashion over an input array $I \in \mathbb{R}^{\beta \times \gamma}$ producing an output e, as shown in Equation 2.1. We represent the fact that the convolutional operation is centered on an element $I_{x,y}$ using the terms $x' = x - \frac{S-1}{2}$ and $y' = y - \frac{S-1}{2}$. The different concept that are part of a convolutional operation is illustrated in Figure 2.5.

$$e = k * I_{x,y} \stackrel{\text{def}}{=} \sum_{i=0}^{S} \sum_{j=0}^{S} k_{i,j} \cdot I_{x'+i,y'+j} \tag{2.1}$$

[4]Since the kernels are not rotated, the operation is technically a correlation

The convolutional operation allows the network to create a linear encoding of the inputs. By multiplying a small number S^2 of weights over the input, one can transform the input into a feature representation that can be combined with other feature representations. The combination of feature representations in CNNs is performed by *filters*, which is just a combination of M kernels. The difference between a kernel and a filter lies in their responsibility; The kernels are used to extract features from inputs, while a filter represents a linear combination of features. The equation for the output f of a filter is just the input weighted by its kernels and then summed, i.e., a weighted sum as seen in 2.2. The variable b represents a bias and is used as a constant to shift the output of a filter. Typically there exist multiple filters for each layer in a CNN.

$$f = b + k_1 * I_{x,y} + k_2 * I_{x,y} + ... + k_M * I_{x,y} \overset{\text{def}}{=} b + \sum_{i=0}^{M} k_i * I_{x,y} \tag{2.2}$$

2.4.2 Activation Functions

The outputs from filters are processed through an *activation function*, which we denote as g, to make it possible for CNNs to solve problems that require non-linear combinations. This can be seen in equation 2.3. Specifically, we define f as the pre-activation outputs of a filter, while $\hat{f}$ is defined as the activated outputs of a filter.

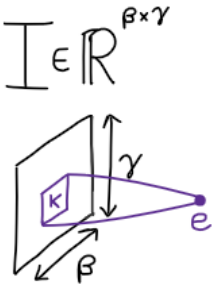


Figure 2.5: The figure shows the relationship between an input I, a kernel k, and the convolutional output e.

Many activation functions can be used for CNNs, with two prevalent ones being the sigmoid and ReLU functions. The sigmoid function, defined as $\hat{f} = \frac{1}{1+e^{-f}}$, behaves as a soft binarizer by squeezing the value range between $[0, 1]$. The ReLU function, defined as $\hat{f} = \max(f, 0)$, behaves as a thresholder by setting any negative value to zero. In our work, we carefully look at the pre-activation outputs of filters and replace the non-linearity by applying logistic regression to transform outputs into probabilities. More details about this will come in section 4.7.

$$\hat{f} = g(b + \sum_{i=0}^{M} k_i * I_{x,y}) \tag{2.3}$$

2.4.3 The Concept Of A Feature Map

When one calculates the output f of a filter for every input position, we call the corresponding map of features a *feature map*, which is illustrated in Figure 2.6. Furthermore, when one does the same for every filter in a layer, we get a stack of feature maps, as

illustrated in Figure 2.7. We use the notation $F \in \mathbb{R}^{C \times W \times H}$ for the stack of pre-activation feature maps and $\hat{F} \in \mathbb{R}^{C \times W \times H}$ for the stack of activated feature maps. The dimensions of the feature maps are the following: channel C corresponds to the filter dimension, i.e., one channel is the output from one filter. The width W and height H correspond to the 2D dimensions of feature maps that are output by the filters of a layer.

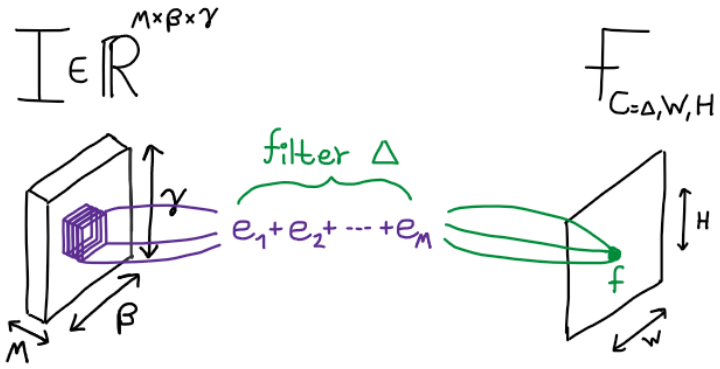


Figure 2.6: The figure shows a feature map $F_{W,H}$ that was generated by applying a filter Δ to some input I.

2.4.4 Summarization Of The Processing In CNNs

The summary of how a layer processes input is as follows; (1) A layer takes in an input of features, denoted as $I \in \mathbb{R}^{m \times \beta \times \gamma}$. (2) For each filter of the layer, a feature map is calculated by convolving the input with its kernels and summing them, which results in a stack of pre-activation feature maps, denoted as $F \in \mathbb{R}^{C \times W \times H}$. (3) The stack of feature maps is activated to introduce non-linearity, which becomes the final output of the layer, denoted as $\hat{F} \in \mathbb{R}^{C \times W \times H}$. This process repeats itself for each layer, where each layer feeds into the next.

The feature representations are processed through a specialized last layer depending on the visual task. For classification, this is commonly a fully connected layer. The fully connected layer is structured differently than convolutional layers because each neuron receives a scalar input that is the weighted sum of the outputs from the previous layer, i.e., there is no convolution but still a weighted sum. Therefore, the output from the last convolutional layer must be flattened before passing to the first fully connected layer. The output of the final fully connected layer is usually, in the multi-class case, passed to the softmax function. *Softmax* is a function that ensures that the sum of hypotheses H is one, which means it, in essence, outputs a pseudo-probability for the input being a specific category. The definition of softmax is shown in Equation 2.4, where the vector $z \in \mathbb{R}^K$ is the output of the final layer before softmax is applied.

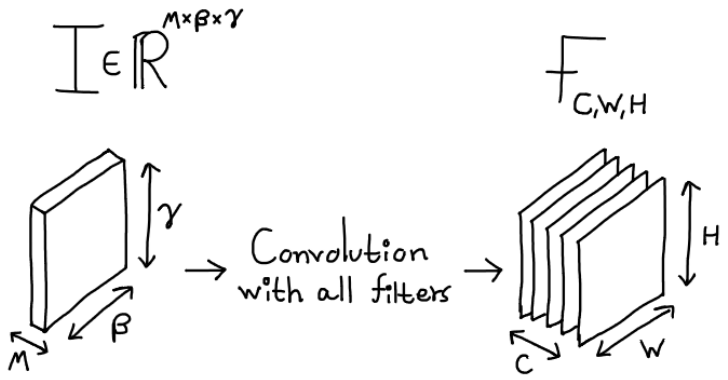


Figure 2.7: The figure shows that a stack of feature maps are formed by applying the set of all filters to an input.

$$H_i \stackrel{\text{def}}{=} \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}} \tag{2.4}$$

It is essential to highlight that the concepts presented here constitute the minimum architecture that can be considered a CNN. In other words, more advanced architectural extensions exist, such as skip connections, dropout nodes, and localization layers. This choice was made to keep the section related to only the most essential CNN concepts. We do not use any form of specialized layers in our work and are primarily interested in the *backbone* of a CNN. We use the term backbone for the convolutional subset of a network, i.e., the layers that perform feature extraction.

2.4.5 How Weights In CNNs Are Learned

In our thesis, we only rely on pre-trained networks, i.e., we do not perform any training ourselves. We still briefly describe how networks are trained to give the reader a completer picture. The learning of features in CNNs relates to the kernel weights in the filters of each layer. At first, when a network is initialized, its filters do not represent a specific concept since their kernel weights have not been learned. The weights are learned using a loss function. The learning process can intuitively be understood as modifying the weights to minimize the loss between the CNN outputs generated from processing examples of a dataset and the ground truth also provided in the dataset.

The loss is backpropagated, meaning the weights are updated sequentially from the end to the start of the network. Backpropagation is necessitated by the weights of layer n being defined in terms of weights in layer $n-1$. The optimization is typically performed using one of many variants of gradient descent, the simplest being stochastic gradient descent. In other words, the update rule of the weights is typically defined in terms of a gradient.

2.4.6 Additional CNN Concepts

Some additional concepts might be useful to know. The first concept is the *receptive field*. For each layer L_n in a DCNN, there exists a receptive field defined as R_n. The receptive field is the region of the input at some prior layer $L_{<n}$ that can be traced forward to a single position at layer L_n. We only use the receptive field because it is the region of the original image pixels that can be traced forward to a single position at layer L_n, i.e., we only care about the region of pixels relative to some layer. This form of attribution guarantees that pixels outside the receptive field have not affected the output. Therefore, activations can be traced back to a region of interest in the original image. We do not calculate receptive fields ourselves but use existing work from Araujo et al. [ANS19] that has derived mathematical formulas and pre-computed such receptive fields for popular networks. We show an example of a receptive field in Figure 2.8.

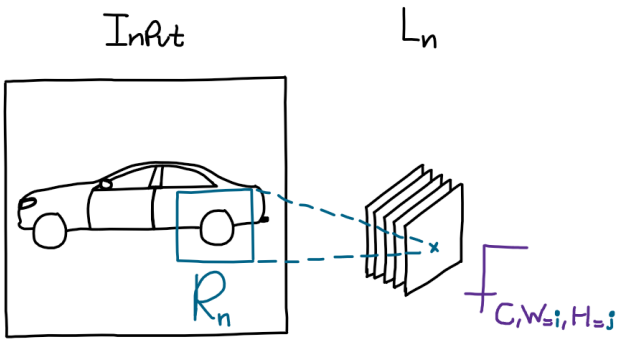


Figure 2.8: The figure shows that each element in the feature map corresponds to a receptive field in the input.

There is also the concept of a *feature vector*, defined as $w \in \mathbb{R}^C$. Some of the related work seen later base their approaches on using the C dimensional vectors in a stack of feature maps. To avoid ambiguity: there are $W \times H$ positions in a stack of feature maps, the vectors we are talking about traverse the C dimension, meaning there are a total of $W \times H$ numbers of such vectors in each layer. These vectors represent all learned features of a layer since they contain all of the filter outputs at one specific location, which is also why the term for these vectors is 'feature vectors'.

2.5 Adapting Deep Learning And Compositionality

[5]Some of the most successful solvers of object detection today are DCNNs[6]. At the same time, these deep networks cannot explain why they make decisions due to the hidden internal representations in the final layers of networks. Compositionality offers an inherent way to improve the explainability of a deep network. By forcing object detection decisions to be made through a specific set of rules, i.e., reasoning about the spatial con-

[5]The text below related to issues in DCNNs has evolved from the pre-project.
[6]This statement is based on related work in chapter 3.

figuration of parts, one effectively provides a framework that can be used for tracing and troubleshooting decisions. For example, a failure can be traced back to a part not being detected or parts appearing in unexpected spatial configurations.

2.5.1 Issues With DCNNs That Compositionality Can Address

Interestingly, compositionality also appears to offer advantages in covering some of the unsolved issues plaguing deep networks in image recognition. The review paper by Yuille et al. [YL21] introduces some of the problems faced by DCNNs:

- A large amount of work is required to create the datasets used by supervised models.
- Defending against adversarial examples is challenging. An adversarial example is an image that has been intentionally perturbed in a manner that causes the model to misclassify the image.
- The sensitivity of models to the context of the image region surrounding some object of interest. This context sensitivity is exemplified in the Yuille et al. paper by showing that the classification of a monkey is changed to human by adding a guitar in front of it.
- Partial occlusion of objects.
- Object viewpoint changes.

One possible solution to alleviate some of the issues DCNNs face is creating a dataset containing training samples covering the problems. For instance, it is possible to train for certain occlusion cases by including examples of occluded objects in the dataset. The argument against training away issues is the combinatorial explosion of examples that would ultimately have to be provided to train for every possibility.

It appears that by adopting compositionality, one would have a natural way to combat many of the issues listed above. For example, architectures based on compositionality appear naturally robust against occlusion because their detections do not rely on seeing the whole object, thus not failing if some parts are missing. The property of robustness against occlusion is demonstrated in a paper by Kortylewski et al. [Kor+20b].

Another example of robustness in compositional architectures is their innate defenses against specific variants of adversarial image attacks. The paper by Cosgrove et al. [Cos+20] demonstrates that compared to standard DCNNs, a compositional architecture is robust against patch attacks without the need for adversarial training. Using a compositional architecture could also reduce the challenge of context-sensitivity since it would be able to ignore non-rule-related contexts. For example, in the context problem presented above of a monkey and guitar being interpreted as a human, a compositional model could have learned that the rule-set for a human consists of finding body parts and ignoring the guitar altogether.

2.5.2 Using Deep Learning In A Compositional Architecture

In our opinion, deep learning-based object detection outperforms older detection approaches simply because they can learn efficient and generalized high-dimensional features from large datasets, not because they have an inherent ability to make better decisions. Prior work provides some proof that this might be the case. For example, in the paper by Brendel et al. [BB19] it is shown that DCNN features can be used in a Bag of Words[7] approach with good results.

The fact that prior work shows that the internal features of a DCNN can be used by non-neural methods and still function well hints at how we should approach the problem of leveraging deep learning. To incorporate both the benefits of deep learning and compositionality, our system should do the following:

1. Extract high-dimensional features from a DCNN by processing images through it.
2. Processing high dimensional features in a system that can generate part detections.
3. Processing part detections in a joint-structure detector that can generate object detections.

The upcoming chapters elaborate on the issue of how to generate part detections using DCNN features and how to combine these part detections into an object detection. In chapter 4 we explore how part detections can be generated from DCNNs, and in chapter 5 we explore how a joint-structure system can combine part detections into an overall decision.

[7] See section 2.2 for an explanation of Bag of Words

Chapter 3

Related Work

We performed literature searches on object detection and compositionality during the pre-project and the work on the thesis. We find that there are already a vast amount of relevant approaches that do something similar to what we want to achieve. The review paper by Zou et al. [Zou+19] provides excellent insight into the object detection field from 1998 to 2018. The paper divides the field into two historical paradigms using 2014 as a cutoff year: The pre-deep learning era detectors before 2014 and the deep learning era detectors after 2014. Not all of the approaches cited in the review paper by Zou et al. are based upon using part compositionality, but they are still interesting as they are directly related to the task of object detection. In the upcoming sections, we summarize the Zou et al. paper and bring up other papers we found during our literature searches.

3.1 Pre-Deep Learning Era

[1]The pre-deep learning detector era did not have the comfort of extensive datasets and powerful computing, reflected in how approaches were crafted. The usual methodology was focused on using hand-crafted features to represent an image and then applying machine learning to the extracted features to learn a detector. The Viola-Jones detector [VJ01; VJ04] and HOG detector [DT05], although not directly related to our search of compositionality, are good examples of the pre-deep learning detector paradigm. The Viola-Jones detector was used for detecting human faces using hand-crafted rectangular feature descriptors. In contrast, the HOG detector was used for detecting whole humans by using histograms of oriented gradients as feature descriptors.

3.1.1 Parts-Based Models

One interesting compositional detection approach of the pre-deep learning detector era is the Parts-based model (PDM) by Felzenszwalb et al. [FMR08]. PDM was revolutionary for the object-detection field in 2008 due to its impressive improvements over the current state-of-the-art result. The model uses HOG feature descriptors to represent an

[1]The pre-deep learning text related to Bayesian graph models and Hierarchical contour models has evolved from the pre-project.

image and manages to learn deformable decompositions of the HOG features into parts. Inference of objects is made by finding a matching composition of these learned parts. Specifically, the model works by using two filters that attempt to recognize features: (1) one large root filter that looks for rough object features and (2) smaller filters that detect finer part features. The smaller filters are directly related to the larger root filter by a vector that defines the expected spatial position of the part. A deformation region is modeled around the expected spatial position, allowing for some position deviance. Other papers [FGM10; Fel+10; GFM11] further improve upon the PDM model by, for example, expanding the model to handle more variability such as occlusion or speeding up the inference. The different components of a PDM is shown in Figure 3.1.

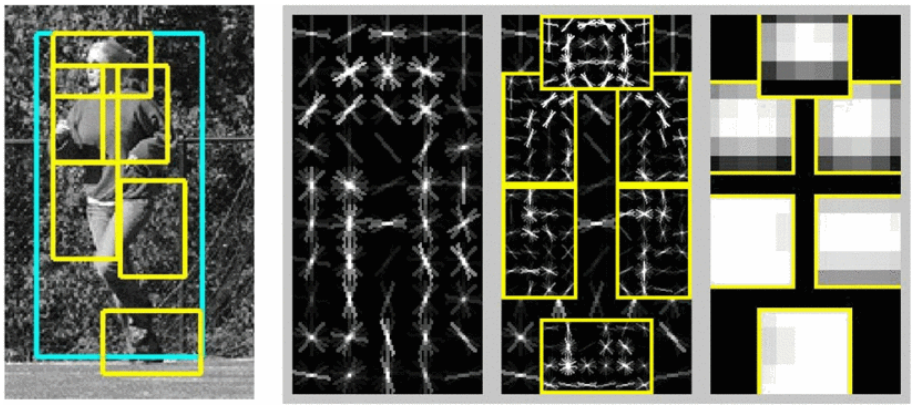


Figure 3.1: *Original image from [FMR08] - Copyright © 2008, IEEE*
See subsection C.1.1 in Appendix for reuse permission. The figure shows different components of a part based model. From the left: (1) example of object detection, (2) the course root filter, (3) the specific part filters, (4) the spatial deformation model of the parts.

3.1.2 Bayesian Graph Models

Although not detection related, we find the paper by Ommer et al. [OB07] as another interesting pre-deep learning approach that uses compositionality. They use a Bayesian framework to validate compositional candidates and measure an object hypothesis from found compositions. The paper's foundation focuses on extracting "atomic parts" of images. These are just pixel regions that have extracted multiple histograms of edge orientations, edge strength, and color as local feature descriptors. Atomic parts are further combined into compositional candidates with a Bayesian relevance model. All the atomic parts that make up a valid composition are then processed in a final Bayesian graph model that considers evidence such as the shape and position of compositions to decide on object category. The approach from Ommer et al. is mainly useful for categorization, but they also discuss how their approach can extend to, e.g., finding missing parts (occlusion) or segmentation. The processing pipeline from Ommet et al. is shown in Figure 3.2.

3.1.3 Hierarchical Contour Models

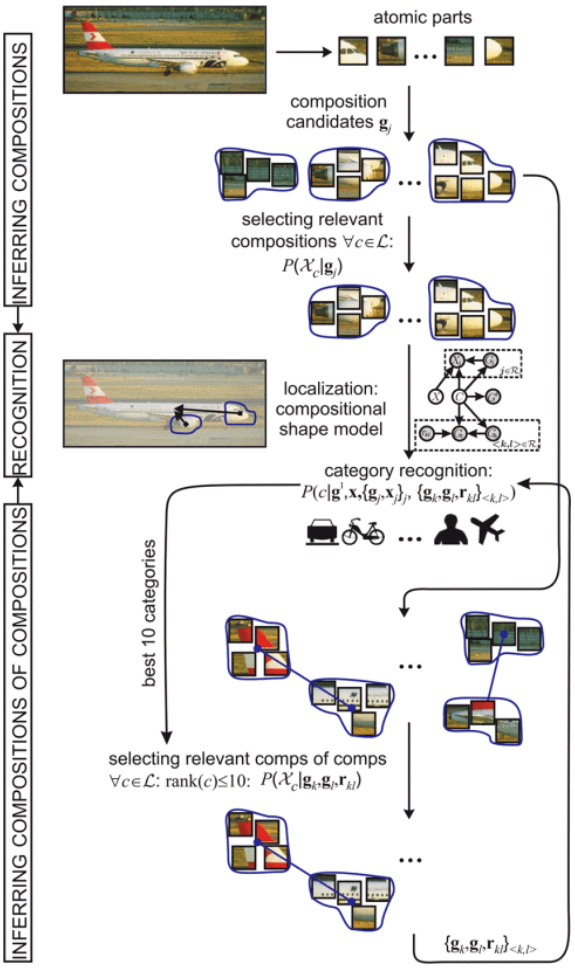


Figure 3.2: *Original image from [OB07] - Copyright © 2007, IEEE*
See subsection C.1.2 in Appendix for reuse permission. The figure shows the processing pipeline from Ommer et al. for scene categorization.

Another example of pre-deep learning compositional approaches is those that create hierarchical dictionaries of pixel contours, where each contour can transform in relation to its predecessor in a tree-like structure. The papers from Fidler et al. [FBL14], Dai et al. [Dai+14] and Kortylewski et al. [Kor+19] are examples of this. The basis of these approaches is centered around extracting Gabor features with a Gabor filter bank. This type of filter extracts edge directions in the images, for example, as seen in figure 3.3. The Gabor features are then allowed to shift in relation to each other, and a larger dictionary is built. More complex shapes will emerge in the higher levels of such a dictionary, and contours that describe specific objects can be found at the top level. An example of a hierarchical tree structure that Dai et al. [Dai+14] builds up can be

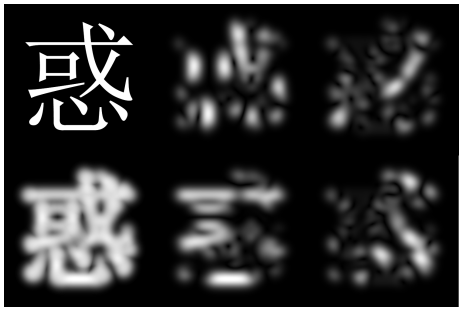


Figure 3.3: *Original image from [Com20b] - Public domain, via Wikimedia Commons*
The figure shows a Gabor filter being applied on a Chinese character in different directions. Top left: Original image, Bottom left: A superposition of all directional features, Top middle: 0 degree filter, Top right: 35 degree filter, Bottom middle: 90 degree filter, Bottom right: 135 degree filter.

seen in 3.4. The papers cited above primarily focus on classification, but Fidler et al. [FBL14] seem to have extended their approach to performing a type of detection based upon contours.

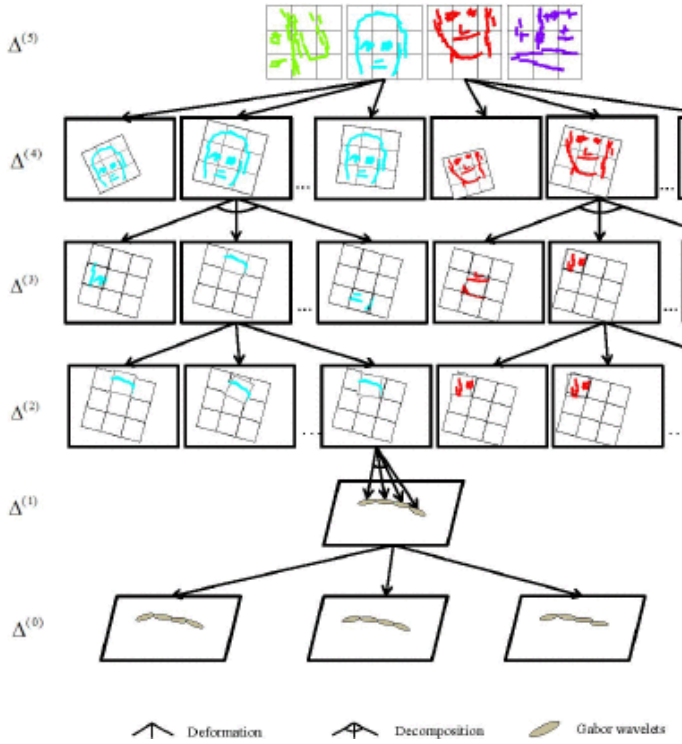


Figure 3.4: *Original image from [Dai+14] - Copyright © 2014, IEEE*
See subsection C.1.3 in Appendix for reuse permission. The figure shows an example of how the hierarchical contours from Dai et al. [Dai+14] are formed in a compositional structure.

3.2 Deep Learning Era

[2]As the deep learning era detectors took over, the paradigm of detection changed from using hand-crafted feature descriptors with machine learning to using automatically learned features in deep convolutional neural networks. This section summarizes the approaches Zhou et al. point out as milestones for deep learning detectors. If the reader is interested in the details behind the approaches, we advise them to read the original paper or look at the review paper. The deep learning era detectors are categorized as either two-stage or one-stage detectors. Two-stage detectors are based upon a module that first proposes relevant regions of an image, then detects using the proposals. The following are milestone two-stage detectors suggested by the review paper: RCNN [Gir+14; Gir+15], SPPNet [He+15], Fast RCNN [Gir15], Faster RCNN [Ren+15]. FPN [Lin+17a]. One-stage detectors perform detections by utilizing pre-defined bounding box templates, which are fitted at inference time. The following are milestone one stage detectors suggested by the review paper: YOLO [Red+16], SSD [Liu+16], RetinaNet [Lin+17b].

[2]The deep learning text related to clustering feature vectors and generative models has evolved from the pre-project.

3.2.1 The Current State-Of-The-Art

From what we were able to find, it seems that the current state-of-the-art deep learning object detectors in 2022 are Vision Transformer architectures, specifically DINO: DETR [Zha+22], which achieves 63.3 test AP on COCO. The best DCNN architecture we find is YOLOR-D6 [WYL21], which achieves 55.4 test AP on COCO. We also find that Vision Transformers have the best performance for the image classification task. The best architecture we find is COCA [Yu+22], with a top-1 accuracy of 91% on ImageNet.

3.2.2 Clustering Feature Vectors

The most prominent compositional deep learning approaches we find seem to focus on clustering the internal representations of DCNNs. This concept can be seen in papers by Wang et al. [Wan+15; Wan+17a] and Liao et al. [Lia+16]. The basic is idea is that image regions are encoded in feature vectors[3]. By processing many images of an object through a DCNN, one can extract examples of feature vectors and clusters to find which ones represent specific repeating image regions. The cluster centers can then be used to detect object parts by comparing how close feature vectors lie in vector space at inference time.

3.2.3 Generative Models From DCNN Feature Clusters

Two essential papers by Kortylewski et al. [Kor+20a; Kor+21a] further expand on how feature vector cluster centers can be used for introducing compositionality into DCNNs by creating generative models. In the paper, [Kor+20a] they focus on using a Bernoulli distribution as their generative model and then perform classification with it. They propose to use the generative model as a backup for a DCNN when its prediction accuracy is below a certain threshold. According to the paper, learning a generative model is complicated because feature vectors are high-dimensional and real-valued. As a result, they propose to encode the feature vectors as sparse binary vectors instead. Binarization is done by comparing feature vectors with the previously mentioned cluster centers using cosine distance. The comparison output is then evaluated against a threshold to decide if the distance is small enough to consider the feature vector an instance of a cluster center.

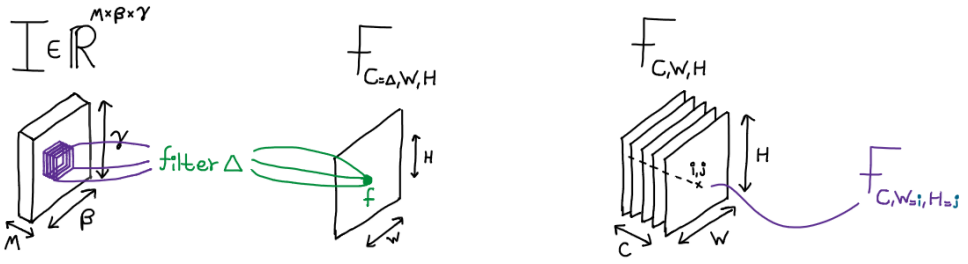
Additionally, the papers also introduce some extensions to their generative model to give it the possibility of handling different object viewpoints, occlusion, and background. In the paper [Kor+21a], they extend their generative model to perform detection and handle high dimensional and real-valued feature vectors. They do this by modeling with von Mises-Fisher distributions instead of a Bernoulli distribution. They also abandon using the generative model as a backup for a DCNN and instead integrate it into the network as a differentiable and trainable component. The new architecture they create is called a Compositional Convolutional Neural Network (CCNN).

[3] See subsection 2.4.6 for the definition of feature vectors

There also exist papers that further expand on the CCNN concept from Kortylewski et al. by applying the architecture or similar architectures to specific visual tasks and problem domains. The paper by Yuan et al. [Yua+21] performs instance segmentation with multi-object occlusion. The paper by He et al. [HKY21] applies a knowledge base to find shared part representations with unseen objects. The paper by Sun et al. [SKY22] performs Amodal Segmentation, i.e., segmenting in which parts of an object are occluded. The papers by Kortylewski et al. [Kor+21b], and Cosgrove et al. [Cos+20] demonstrate the robustness of the compositional architectures against occlusion and adversarial attacks. The paper by Wang et al. [Wan+20] introduces improvements to disentangle context from objects.

3.3 Our Work

Our interests lie in detecting objects through a composition of their constituent parts. We want our method to be based upon the strengths of deep learning while also incorporating explainable reasoning for system decisions. With these constraints in mind we developed a method in section 4.7 that uses filters of pre-trained DCNNs for generating probabilistic part detections. In section 5.2 we further attempted to combine our part detections into an object classification. Finally, in chapter 7, we also attempt to implement localization using our classifiers. Our work is different from the pre-deep learning era approaches since our part detectors rely on automatically learned internal representations of a DCNN, i.e., we avoid using arbitrary feature descriptors that must be hand crafted. We also differ from the deep learning era approaches by how we generate our object decisions. The standard paradigm for deep networks has been to generate decisions that are inherently incapable of being explained, e.g., fully connected layers for classification and region proposals for detection. We want to generate these decisions from an explainable system that can reason about parts instead.



(a) Example of what a filter is in the context of convolutions: It produces one feature map. Filters are what we use in our work.

(b) Example what a feature vector is in the context of convolutions: It is one position in a stack of feature maps. Feature vector is what the CCNN approaches have used.

Figure 3.5: The difference between feature maps and feature vectors.

3.3.1 Difference Between Our Work And CCNNs

The most similar method to ours is the CCNN approach, which attempts to cluster feature vectors and then process them in generative models. Our methods find part detections from internal representations of DCNNs and then use them in explainable systems. The significant difference, however, lies in the fact that we use filters while they use feature vectors. A feature vector contains the value of every filter in a layer at one position, which means they quickly become high dimensional at later layers of DCNNs. Our method avoids the problem of high dimensions by attempting only to use the filters that are measured as important for an object, i.e., pruning unimportant filters in a layer. We illustrate the difference between a filter and a feature vector in Figure 3.5.

The CCNN approaches require clustering to localize parts, which we would argue makes their approach less explainable due to the following. One must define a pre-set number of centers to find, which then assumes prior knowledge of how many centers exist, which is very hard due to the high dimensional nature of feature vectors. Alternatively, one can also trust that the clustering method has converged to a correct number of centers, which means relying on high-dimensional machine learning techniques. Instead, our method takes advantage of filters' inherent localization ability, which is very simple and avoids any form of clustering.

Chapter 4

Building A Part Detector

This chapter is dedicated to detailing our efforts in developing part detectors. In section 4.1 we define what a part is and further describe the compositional relationship that exists between parts and objects. In section 4.2 we describe related work that utilize the internals of DCNNs to create part detectors. In section 4.3, the dataset we use in our thesis is described, and the following sections 4.4, 4.5, 4.6, and 4.7 describe the different approaches of creating part detectors that we tried. The most important approach is the one from section 4.7, which is to our knowledge, a novel part detector approach that we have created with some inspiration from the papers [Bau+17; ZWZ18].

4.1 The Concept Of Parts And Part Detectors

We first begin by defining what parts are and what properties we expect from them. Since we operate in a subset of visual tasks that consists of recognizing objects, we only look at parts in the sense that they are an element of an object. Specifically, we use two terms in this thesis: 'part' to describe an object-constituent and 'object' to describe a whole. It is important to emphasize that parts are subjective and not strictly defined. One could, for example, ask what parts a car consists of, the answer to which would be different depending on who is asked. Some might divide a car into parts by its construction, for example, its wheels, chassis, and doors. However, other divisions also exist. Some might, for example, divide a car into three arbitrary sections dependent on direction, such as front, middle, and back.

4.1.1 The Difference Between Semantic Parts And Visual Concepts

In the context of using DCNNs, the subjective nature of parts is important because we can not exactly know what such networks learn. Since DCNNs are often optimized without part labeling, their learned concepts do not necessarily match what humans expect. Related work by Wang et al. [Wan+15; Wan+17b] has adopted the categorization of *semantic parts* and *visual concepts* to distinguish further what is meant behind this. Semantic parts are human-understandable, which means they map directly to natural language. Visual

concepts are parts that do not necessarily match the human understanding of a part but rather a visual pattern that can easily be recognized because they contain a specific feature. Although the ideal behavior for an interpretable system is to detect semantic parts, we expect part representations in DCNNs to be visual concepts instead of semantic parts. This expectation is born from the view that learning semantic parts over visual concepts has no inherent benefit for an optimization process. We show an example of a semantic part in figure 4.1 and a visual concept in figure 4.2.as

Figure 4.1: Example of a semantic part in the form of a car wheel. The bounding box is manually drawn for illustration purposes.

Figure 4.2: Example of a visual concept in the form of the circular edge between the car wheel and car body. The bounding box is manually drawn for illustration purposes.

4.1.2 The Deeper Hierarchy Of Parts

Parts do not only exist in a singular relationship with objects; we also expect them to exist in a larger hierarchy where one part could be the child of another part. Therefore, what we describe as parts also depends on what objects we are looking to detect and how deep in a part hierarchy one wants to search. One example of an object could be a cat whose parts are the head, torso, legs, and tail. Another example of an object could be the cat head itself. The ears, eyes, nose, and mouth could be parts in that instance. We illustrate the cat and cat head example in figure 4.3 and figure 4.4.

4.1.3 The Multiplicative Relationship Between Parts And Objects

We further define parts as having a many-to-many multiplicity with objects. This multiplicity means we expect parts to be a shared constituent of multiple objects and that objects can use a part multiple times. This multiplicity is important because it simplifies a recognition system through abstraction. We would, for example, not want a system to detect specific parts like 'front bus wheel' and 'left car wheel', but rather the general concept of a wheel that can be used multiple times. This form of abstraction is inherently subjective. We could, for example, expect a wheel to be a shared part that belongs to vehicles like buses, cars, and mopeds. Some might then argue that a bicycle wheel does

not belong in the same category since it has some differences from automotive wheels. We do not make any fine-grained distinctions or rules regarding this but rather highlight that the many-to-many relationship exists and should be considered. Figure 4.5 illustrates an example of how a wheel could have a many-to-many relationship with multiple vehicles.

4.1.4 The Spatial Relationship Between Parts

Parts also have an important spatial relationship with other parts that we expect a recognition system to consider. As humans, we not only use the presence of parts to indicate the presence of an object, but we also use the spatial offset between the parts to evaluate what an object is. For example, if one takes the cat head from Figure 4.4 and starts moving the parts around, at some point, we as humans would have a threshold where we no longer detect the cat head. We show this example in Figure 4.6. It is possible to ignore the spatial relationships since part presence alone is often a good indicator of an object's presence. However, that would be less interpretable and robust than properly considering spatial relationships.

4.1.5 Reasoning About Part Detectors

Using the definitions given in the previous subsections, we can describe part detectors and how we expect them to behave. Given that parts exist in hierarchies, we would ideally want a part detector to only fire at a specific level of a part hierarchy. In other words, it should be particular enough only to detect the level it has been assigned and ignore lower-level parts. This particularity ensures that the part detector is consistent in the parts it detects and lets us avoid tracking the hierarchical detection level. Given the property of parts having many-to-many relationships with objects, we would also want a part detector to be general enough to detect the same part in different contexts. This generality ensures that a wheel will still be detected without needing to be part of an object such as a car or bus. A part detector that is too sensitive to context could, for example, avoid firing on the same part if it becomes biased to specific visual cues. Given the last property of parts having spatial relationships to other parts, we would also want a part detector to localize its detected parts. If part detectors only have the option of doing classification, we would lose crucial spatial information that can be used, consequently limiting a recognition system of objects to only perform classification.

4.2 Part Detectors In The Context Of A DCNN

As stated in section 2.5, we aim to create part detectors by leveraging the internal part representations of DCNNs. We believe that by doing so, we can create part detectors without needing part annotations. However, it is not immediately obvious how one best can extract the part detectors. Therefore, we devote the following subsections to exploring several proposed approaches to creating part detectors using DCNNs. In subsection 4.2.1, we explore how one can encourage the learning of part representations in

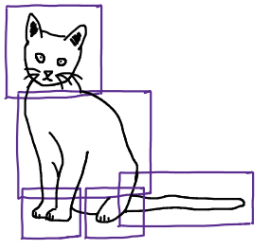


Figure 4.3: Example of parts localized on a cat.

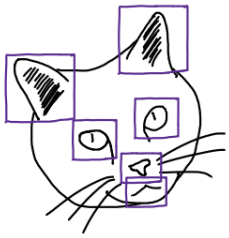


Figure 4.4: Example of parts localized on a cat head.

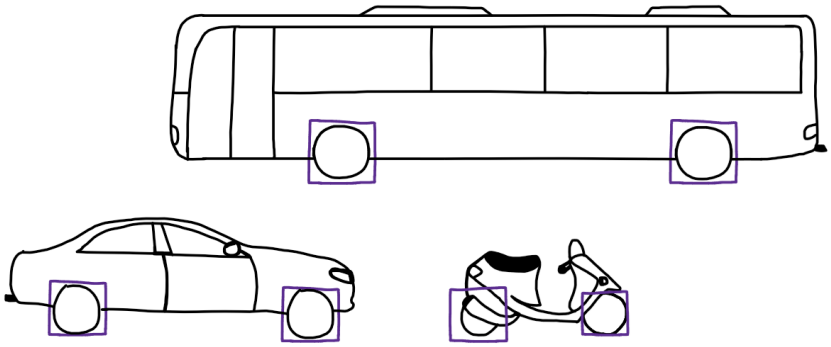


Figure 4.5: Example of a many-to-many relationship. Here a wheel is an example of a part that is shared between multiple objects like car, moped & bus. The objects also use the part multiple times.

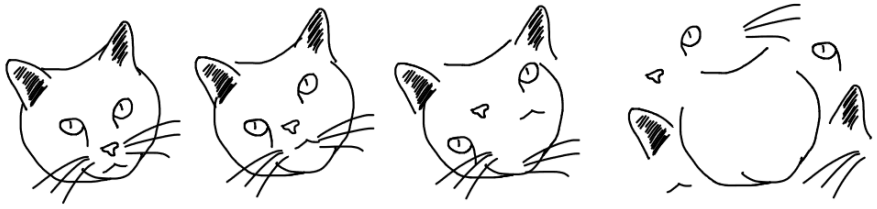


Figure 4.6: Example of how a cat head can be distorted by moving its parts around

DCNNs through several different means. In subsection 4.2.2 we explore utilizing the existing representations learned by DCNNs without the need for directing the networks. In subsection 4.2.3, we explore a group of approaches that, in essence, consider the part representation of multiple filters in aggregate by forming a feature vector spanning all the feature maps.

4.2.1 Learning Parts In DCNNs

Training To Detect Parts

One way to use DCNNs for generating part detections is by following the existing paradigm of solving visual tasks by training on many examples. The most direct way to do this is through supervised training on deep neural network architecture, which also requires access to a dataset with labeled parts. Some examples of such datasets are the following:

- PASCAL-Part dataset [Che+14]
- PartImageNet dataset [He+21]
- Animal Parts dataset [NLV16]
- CUB-200-2011 dataset [Wah+11]
- CelebAMask-HQ dataset [Lee+20]

The paper by Gonzalez et al. [GGMF18] is one example that follows the standard paradigm and trains a custom neural network architecture directly on a part dataset. Their approach is built upon an architecture that considers part appearance, location, and the overall object when generating part detections. In their approach, a wheel is not detected directly from its visual cues but rather a hypothesis where the context of the object and the part's visual cue lead to detection. Another example is the paper by Tritrong et al. [TRS21] which investigates if Generative Adversarial Networks (GAN) learn meaningful structural representations. This paper deviates somewhat from the standard paradigm as it proposes using internal representations extracted from a pre-trained GAN to train a segmentation network with a small number of labeled examples. The paper by Huang et al. [Hua+16] is yet another example. Their approach builds upon training a DCNN network to generate part locations and then feeding the locations to another DCNN network that performs classification using the locations.

Training To Recognize Parts

Examples not directly related to generating part detections but rather the task of part learning can be found in fine-grained image classification, which is classifying sub-categories of an object super-category. Because object sub-categories share many visual similarities, detecting their parts is one of the few ways that subtle differences can be recognized and then used to perform classification. These approaches typically construct custom architectures and training regimes to ensure neural networks learn internal representations

that consider parts when recognizing objects.

The paper by Zhang et al. [Zha+21] is an example that explores part learning by introducing a custom architecture called PCA-Net. The paper's approach is to find unique sub-category parts by performing a forward pass with two input images of the same subcategory and then analyzing the similarities of the generated neural representations. Attention modules in PCA-Net then focus attention on the complementary representations between different sub-categories, ensuring that the subtle part differences are learned. The paper by He et al. [HP19] is another example where a custom training regime called fine-grained visual-textual representation learning is used. This approach builds upon the fact that textual features can describe visual information and takes advantage of this by first discovering what visual and textual features are related via pattern mining. These pairs are then used in a learning scheme where visual information locates parts and textual information distinguishes the detected parts.

Part learning also appears in different methods based upon regularization through masking. The papers by Yan et al. [Yan+20] and Ji et al. [Ji+21] are examples of this. They both base their approaches on erasing patches from an input image. The patch erasure is done by removing the region which garners the strongest network attention. The level of network attention is measured using a saliency map, which is an attribution of pixel contribution to the network hypothesis. In other words, it serves to locate the pixels in the input image most important to the final hypothesis. Both approaches utilize separate networks for generating the saliency maps. The intuition behind the approaches is that by removing regions, one can force the network to consider visually different image patches and thereby enforce part learning. Another example of masking can be found in the papers by Zhang et al. [ZWZ18; Zha+20]. Their approach attempts to introduce interpretability into DCNNs by making filters uniquely represent parts. They use a custom layer that regularizes filters by masking their feature maps.

4.2.2 Leveraging Part Sensitivity In Filters

Another approach for generating part detections relies on using the internal filters of DCNNs as concept detectors. This approach is built upon the many works investigating the internal behavior of neural networks. The general discovery from these works is that neural networks can encode meaningful information and that filters of a network can represent specific concepts.

For example, the paper by Zhou et al. [Zho+14] shows that DCNNs trained on scene classification contain filters that behave as object detectors through receptive field estimations from discrepancy maps. The paper by Yosinski et al. [Yos+15] collects and expands visualization methodologies into a single tool that lets users view the output of filters live. The line of papers by Bau et al. [Bau+17; Bau+20] further shows that filters in DCNNs not only represent objects but can also represent many different concepts such as object parts, textures, colors, materials, and scenes. They also show that filters at later layers of networks represent higher-level concepts such as objects and object parts. In contrast,

filters at earlier layers represent low-level concepts such as textures and colors. Their method is called network dissection and differs from other works by directly mapping feature map outputs of filters to ground truth segmentation masks of concepts.

Other related papers do not directly use filters but instead take advantage of their concept-mapping behavior to find parts. One example is the paper by Zhang et al. [Zha+19] that proposes an unsupervised method of part detection through mining co-occurring patterns in filter feature maps. Another example is the paper by Zhang et al. [Zha+18] which proposes to use the relationships between activation peaks of feature maps in multiple layers to create an explanatory part graph. It should be noted that, in general, many of the other cited works from earlier sections also use this behavior directly or indirectly. The regularization paper mentioned earlier by Zhang et al. [ZWZ18] is based upon masking filter feature maps, which uses the fact that filters activate over parts as a presumption for why their masking works.

4.2.3 Feature Vectors As Parts

Another possible avenue to create part detectors from DCNNs is using feature vectors. Feature vectors are generated by extracting activations from feature maps along the channel dimension resulting in $W \times H$ feature vectors $w \in \mathbb{R}^C$. Each feature vector corresponds to a receptive field in the input image, with vectors from later layers corresponding to a larger region in the input image. We illustrate the concept of a feature vector in Figure 4.7. The use of feature vectors is explored in several papers, such as [Wan+15; Lia+16; Wan+17a; Kor+20a; Kor+21a]. We consider the primary difference between this group of approaches and what was described in subsection 4.2.2 to be that no attempt is made to decouple the representations learned by the filters. Instead, the filter activations are considered in an aggregate vector form.

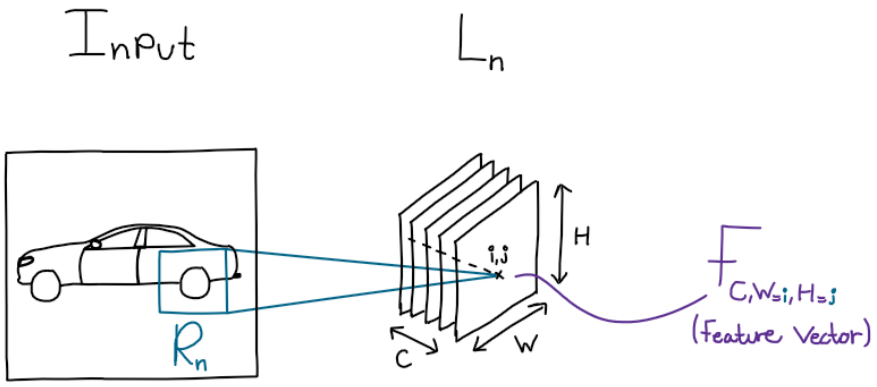


Figure 4.7: Example of a feature vector at position 'i,j' in a stack of feature maps at layer L_n, and its corresponding receptive field R_n.

Generally speaking, the idea is to generate a dictionary of K different visual words found in the feature space of the extracted feature vectors by clustering, which generates cluster centers or other representations that can form a word. Once the cluster centers have been

found, one can compare future feature vectors against a dictionary of centers and select argmax to assign the vector. The papers use different approaches to selecting the number of visual words.

The papers [Wan+15; Kor+20a] both apply K-Means++ clustering on feature vectors extracted from N input images. [Wan+15] uses euclidean distance for clustering while [Kor+20a] states that they use cosine distance. However, [Wan+15] does show that l_2-normalized vectors achieves better results. Note that the distance found between l_2-normalized vectors using the euclidean distance is proportional to the cosine distance. Therefore, the authors of [Wan+15] hypothesize that the improvement is due to direction being a better visual similarity measure than distance. The papers use different approaches to selecting the number of visual words. [Wan+15] initializes K to C and uses a greedy cluster merging algorithm to reduce this. [Kor+20a] sets K to be 50 per class to detect.

The papers [Kor+20a; Kor+21a] further extend the ideas mentioned above by creating mixture models to handle different viewpoints that can occur in the training set. One difference in [Kor+21a] compared to the others in terms of clustering is the use of von Mises-Fisher instead of K-Means++ to find clusters. Both [Kor+20a] and [Kor+21a] use an occlusion model to explicitly handle the presence of an object occluder in the input image.

Alternative Applications Of Feature Vectors

The papers [Wan+17a; Lia+16] take a somewhat different approach to apply feature vectors. In [Wan+17a] they use feature vectors as a visual cue to the presence of a semantic part. Therefore, their proposed method requires part labels to model the spatial offset between the feature vectors and the semantic parts. The presence of a semantic part is defined using a probabilistic formulation based on the presence and location of visual cues. The paper [Lia+16] proposes three clustering regularization terms, one of which is based on using feature vectors that they name spatial clustering. They expect that adding this regularization term will encourage the network to capture parts shared by multiple objects or scenes. The regularization term is calculated using the euclidean distance between the feature vectors. The other regularization terms were introduced in subsection 4.2.1.

4.3 Creating A Dataset

In all of the investigated approaches, we used a custom dataset created to facilitate finding car part detectors. We restricted ourselves to cars for two reasons: (1) cars have been one of the predominant classes that other part detector-related works in section 4.2 use. (2) when we attempted to use a dataset of boats, we found water detectors together with part detectors, which was unexpected. Therefore, we experimented with other classes on hold until we understood why that was. Later on in the discussion section 6.2.3 we elaborate on the problems we observed when using boats.

4.3.1 The Car Dataset

Our custom dataset consisted of ≈ 4600 images, where half of these were car images that originated from the Stanford cars [Kra+13] dataset and the other half were non-car images that originated from the COCO [Lin+14] dataset. The dataset mixture was created by selecting an equal amount of random images from both. We set the label of the Stanford cars images as 1, and the label of random COCO images as 0. When extracting the random images, we ensured that no car images were used from COCO, but we did use images of other vehicles like buses and bikes. Both datasets had bounding boxes, but we did not use these in our approaches.

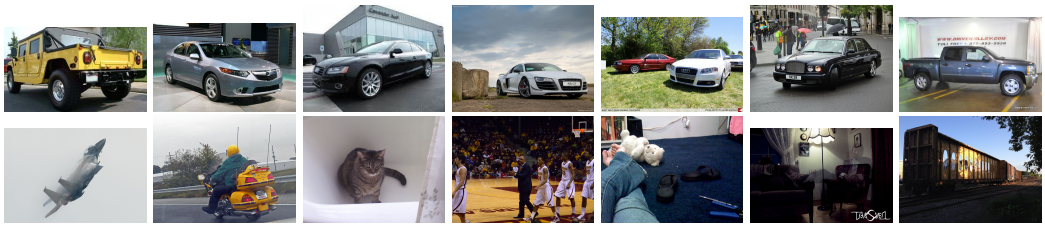


Figure 4.8: Example images from our mixture dataset. Top row shows the car images with label $= 1$. Bottom row shows non-car images with label $= 0$

4.3.2 Pre-Processing Steps

We performed preprocessing steps on our data, such as scaling images down to predefined input size. In all of our approaches, we ended up using VGG16, which means images had to have the dimensions 224x224. To standardize the image size, we used the existing resize transformation supplied by the Torchvision library [Tor] to force images to fit 224x224 as they were loaded.

4.4 Approach A - Feature Vectors As Parts

4.4.1 Introduction

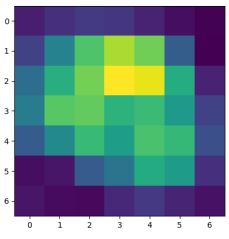
In the preliminary experiments conducted during the pre-project, we explored extracting a set of feature vectors from feature maps and clustering them, which was proposed in several papers such as [Wan+15; Lia+16; Wan+17a; Kor+20a; Kor+21a]. These approaches were described in high-level detail in subsection 4.2.3. One of the primary issues we encountered during the pre-project was that finding meaningful part representations from the cluster centers was difficult. Therefore, we decided to take a second look at the feature vector approach in our thesis with the hope that we could address the issues we experienced. We tested a minor filtering extension that tries to filter out redundant information. In the following sections, we will describe the results of this effort and the steps taken to produce them.

4.4.2 Creating Feature Vectors

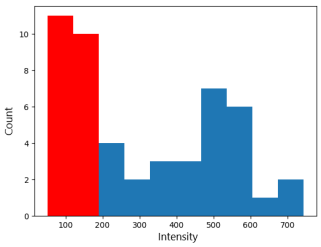
The first step of this approach requires us to create a set of feature vectors from the feature maps generated by one layer in a CNN. We define the set of feature vectors Ψ as vectors extracted along the channel dimension of the set of feature maps $\hat{F} \in \mathbb{R}^{C \times W \times H}$. Therefore, we define feature vectors in the same way as [Kor+20a]. However, as mentioned earlier, we have introduced additional pre-processing to filter out redundant information. The filtering step involves selecting the feature vectors where the sum of component scalars is greater than a threshold ϕ. More formally, we create a proper subset of feature vectors $\Psi \subset \Psi_{unfiltered}$, where $\Psi_{unfiltered}$ consists of $N = W \cdot H$ feature vectors $w \in \mathbb{R}^C$ extracted from F. The subset Ψ is therefore defined as:

$$\Psi = \{w | w \in \Psi_{unfiltered} \wedge \sum_{i}^{C} w_i > \phi\} \tag{4.1}$$

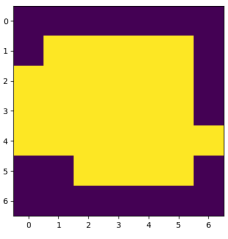
(a) This image shows a bilinearly interpolated mask from Figure 4.9d overlaid on the input image.



(b) An example of a summed feature map $\hat{F}_{Sum}$ generated by layer Pool5 in VGG-16.



(c) The intensity threshold found for $\hat{F}_{Sum}$ by minimizing Equation 4.2.



(d) A mask generated from $\hat{F}_{Sum}$ using the calculated intensity threshold. The light colored grid elements indicates that the corresponding feature vectors are part of W.

Figure 4.9: Example images that shows the steps to filter the feature vectors of an input image using Equation 4.1.

The threshold ϕ is an intensity threshold found using Otsu's method on the sum of all feature maps $\hat{F}_{sum} = \sum_i^C \hat{F}_i$. Otsu's method works by finding a threshold ϕ from $\hat{F}_{sum}$ such that the inter-class variance σ_B^2 between the intensity distributions of two classes is maximized [Ots79]. In other words, we assume that there is always a clear distinction

between the intensity distributions of the background class and object classes' intensity distributions. We create a histogram and assign each of the intensity values in $\hat{F}_{sum}$ to a bin with index i. In the greyscale case, it is natural to set the number of bins L to 256, but we find that ten bins are sufficient for the summed feature map. Each bin represents the probability p_i that a randomly drawn element has an intensity belonging to a bin of index i, i.e., $p_i = n_i/N$ where n_i and N are the bin count of bin i and the total element count respectively. ϕ is selected as the maximum value of the intensity range that defines the elements of a bin. The optimization term is defined in Equation 4.2 while the weights and means are defined in Equation 4.3 and Equation 4.4 respectively. An example of steps of the whole filtering process is shown in Figure 4.9.

$$\sigma_B^2 \stackrel{\text{def}}{=} \omega_0\omega_1(\mu_0 - \mu_1)^2 \tag{4.2}$$

$$\omega_0 \stackrel{\text{def}}{=} \sum_{i=1}^{k} p_i \qquad \omega_1 \stackrel{\text{def}}{=} \sum_{i=k+1}^{L} p_i \tag{4.3}$$

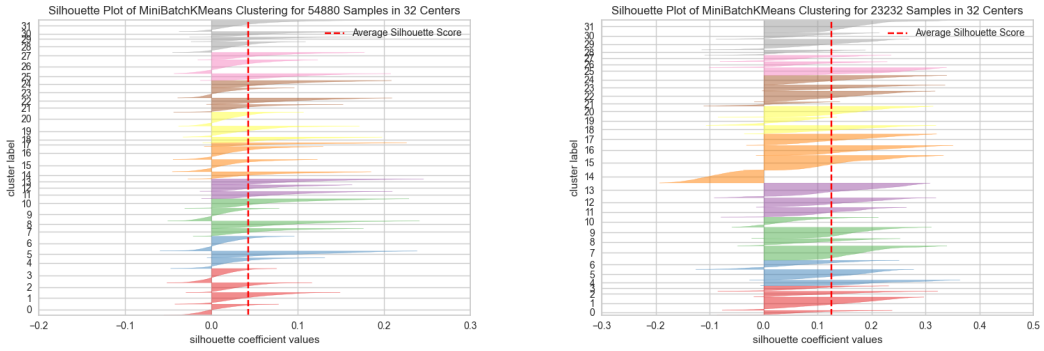
$$\mu_0 \stackrel{\text{def}}{=} \sum_{i=1}^{k} \frac{i \cdot p_i}{\omega_0} \qquad \mu_1 \stackrel{\text{def}}{=} \sum_{i=k+1}^{L} \frac{i \cdot p_i}{\omega_1} \tag{4.4}$$

We achieved higher cluster density and faster run time by using this filtering algorithm. The increase in cluster density can intuitively be understood as the cluster no longer having to represent the vast set of different features that can occur in a background region, so only features relevant to the categories of the classifier are represented in the cluster space. See figure 4.10 for a representation of the density before and after applying the filtering algorithm. The cluster density is measured using the silhouette coefficient. The *silhouette coefficient* gives a metric based on the mean distance of some point j to all other points m in the same cluster A, denoted as $a(j)$ and the mean distance to the closest cluster B, denoted as $b(j)$ [Rou87]. The silhouette coefficient is defined in Equation 4.5, while the two distance terms are defined in Equation 4.6 and Equation 4.7.

$$s(j) \stackrel{\text{def}}{=} \frac{b(j) - a(j)}{max(a(j), b(j))} \tag{4.5}$$

$$a(j) \stackrel{\text{def}}{=} \frac{1}{1 - |C_A|} \sum_{j \neq m, m \in C_A} d(j, m) \tag{4.6}$$

$$b(j) \stackrel{\text{def}}{=} min_{B \neq A} \frac{1}{|C_B|} \sum_{m \in C_B} d(j, m) \tag{4.7}$$



(a) Unfiltered

(b) Filtered

Figure 4.10: Two plots showing the silhouette coefficient as a metric for cluster density on unfiltered and filtered feature vectors. The difference in samples is simply the result of $\Psi \subset \Psi_{unfiltered}$.

4.4.3 Finding Clusters And Matching Feature Vectors

The second step of this approach entails finding K clusters among the feature vectors W. We have chosen to use Mini Batch K-Means++ to find the clusters as this yields a center that can be used for similarity thresholding. Once we have found K cluster centers $d \in \mathbb{R}^C$, we treat these as a dictionary D of visual words against which we can compare all vectors in W. We use cosine similarity $S(w, d)$ as the similarity metric. We define cosine similarity in Equation 4.8.

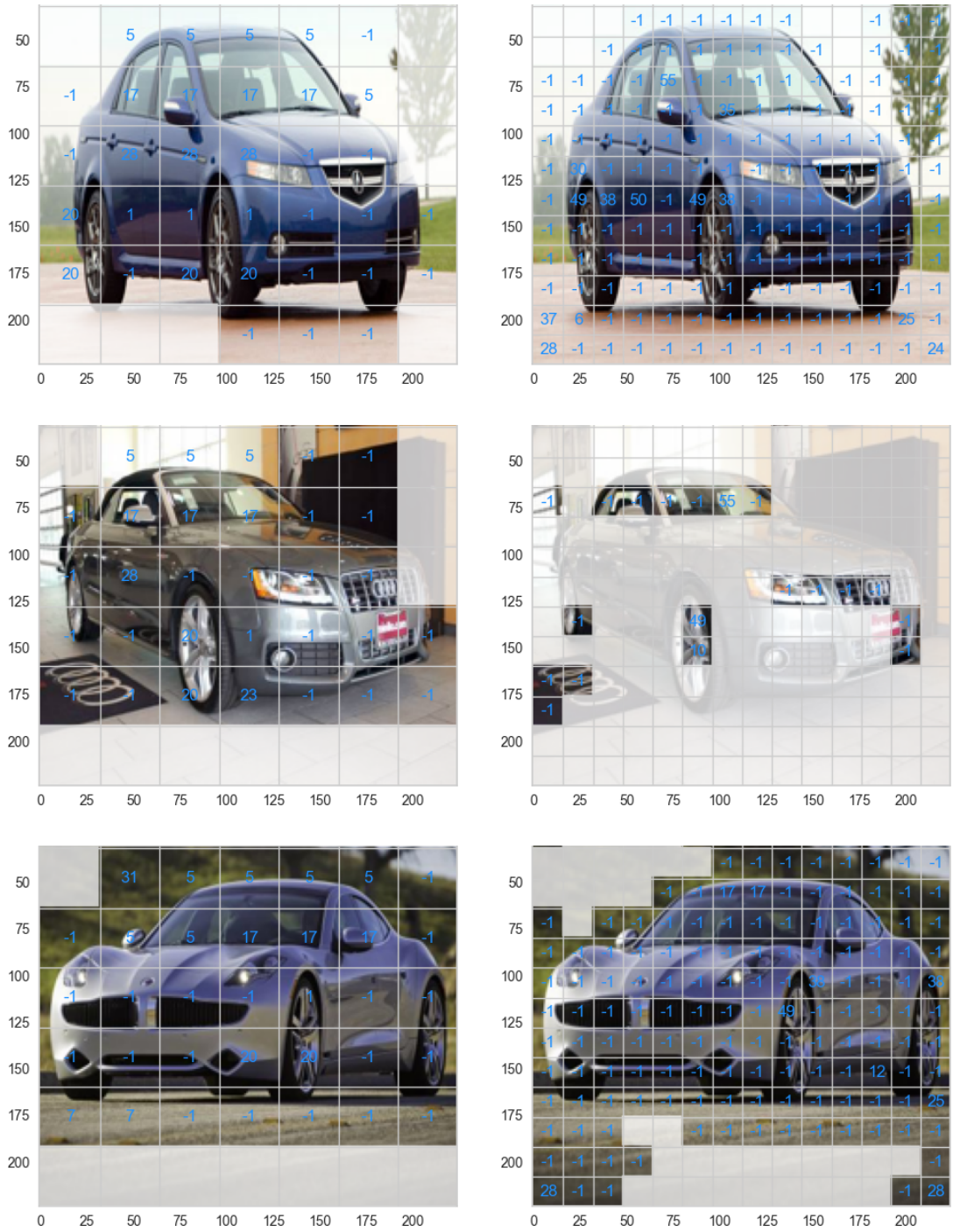
$$S(w, d) \stackrel{\text{def}}{=} \frac{w \cdot d}{\|w\|\|d\|} \tag{4.8}$$

This approach is, in practice, the same as done in [Kor+20a], the only difference being that they use the complementary cosine distance instead. We can use a larger similarity threshold ρ than was used in their approach, but no conclusions can be drawn since we do not use the same dataset. Categorizing feature vectors as one of K visual words from the dictionary of cluster centers is done using Equation 4.9.

$$word(w) = \begin{cases} argmax_D(S(w, d)) & \text{if } \exists d \in D, S(w, d) > \rho \\ -1 \end{cases} \tag{4.9}$$

4.4.4 Applied As A Part Detector

The method was tried on layers Pool4 and Pool5 of VGG-16. Some results from applying the approach to input images are shown in figure 4.11. The filtering approach seems to yield good results on the outputs of Pool5 in VGG-16. In contrast, the results for the earlier Pool4 layer are quite weak. This disparity can likely be partly explained by the earlier layers capturing more shared information, i.e., some features are likely to occur both in the background and in the object. In other words, the features are less class-specific in earlier layers. Shared features will make it more difficult to separate based on the activation intensities alone.



(a) VGG16 Pool5 ($K = 32$)

(b) VGG16 Pool4 ($K = 64$)

Figure 4.11: Input images overlaid with an interpolated image with dimensions originally equal to the output feature maps of the source layer. The numbers inside each grid element indicates which cluster the corresponding feature vector was assigned using function 4.9. The unassigned regions correspond to feature vectors $w \notin \Psi$, which were described in subsection 4.4.2. It is important to note that the grid elements do not correspond to the full theoretical receptive field, but rather a smaller center region of the receptive field.

One concern we have is the selection of the clustering algorithm since it likely requires reasoning about the topology of the feature vector space. In other words, K-Means fits K centroids to the feature vector space, but it is not apparent that a centroidal definition

of the vector space is appropriate. While this does not preclude the methods from being performant, it does allow for the possibility that a more appropriate clustering algorithm exists that could produce improved accuracy.

Ultimately, the second look at this approach did not yield any significant improvements concerning finding meaningful part representations. We find the representations captured by the feature vectors to be quite abstract and therefore challenging to interpret and recognize in other images. Still, there is likely more that could have been done. For instance, looking more in-depth at the feature representations captured by the different layers is possible. However, we found that using a filtering method to remove background information holds promise. The experiments indicate that it is advantageous to remove the background information as it seems to produce a set of vectors that are easier to delineate via clustering.

4.5 Approach B - Part Detection Via Pattern Mining

4.5.1 Introduction

The paper [Zha+19] proposes an approach utilizing pattern mining to find recurring activations across feature maps in a CNN. The proposed method does not require object or part annotations but does require a pre-trained CNN. In the following sections, we will describe our attempt at recreating the proposed part detection approach and evaluate its suitability. Note that we did the software implementation since we could not find a publicly available repository containing an implementation of the approach.

4.5.2 Generating The Support Map

The authors of [Zha+19] decided to use VGG-16 as their base model, which we also did. The approach is based on detecting recurring activations across all feature maps $\hat{F}_R$ and F_P from the layers relu5 and pool5, respectively. They state in the paper that the motivation for extracting feature maps from two layers is to alleviate the loss of information incurred from only considering one layer. From our understanding, including relu5 allows the algorithm to consider non-maximum activations, which may be discriminative while maintaining the importance of maximum activations.

The feature maps F_P have lower dimensionality than $\hat{F}_R$, which requires us to upsample all feature maps in F_P. Upsampling is done using bilinear interpolation, resulting in N feature maps $F \in \mathbb{R}^{W \times H}$. Each feature map F is flattened, resulting in N vectors $t \in \mathbb{R}^{W \cdot H}$. The set of all vectors T is defined as $T = \{t_1, t_2, ..., t_{N-1}, t_N\}$, with each element being called a transaction. The reasoning for flattening the feature maps is to simplify finding patterns. More specifically, the algorithm for finding patterns does not operate on 2D grid structures but instead expects a set of vectors. Each position in a feature map is considered a candidate for being part of a transaction t. The candidate position's coordinates i are added to a transaction t if the scalar of the position is greater than α, with α being the average of all activations in a feature map greater than zero. In

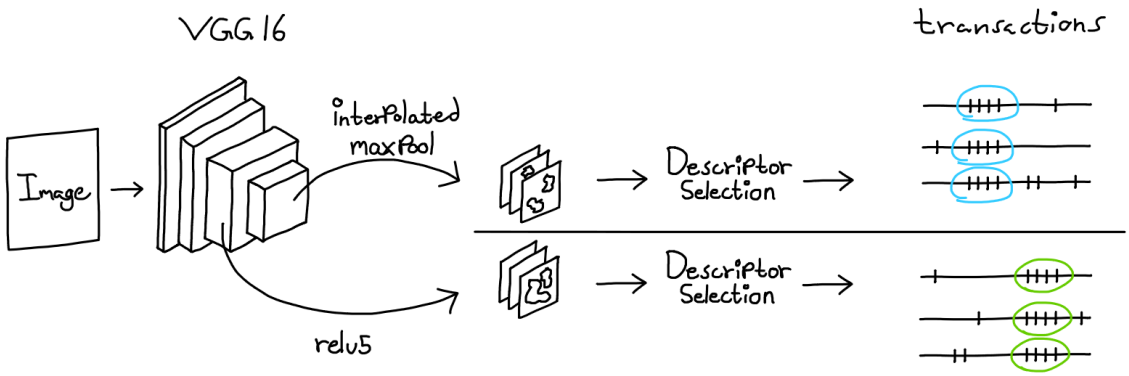


Figure 4.12: Visualization of the transaction generation pipeline.

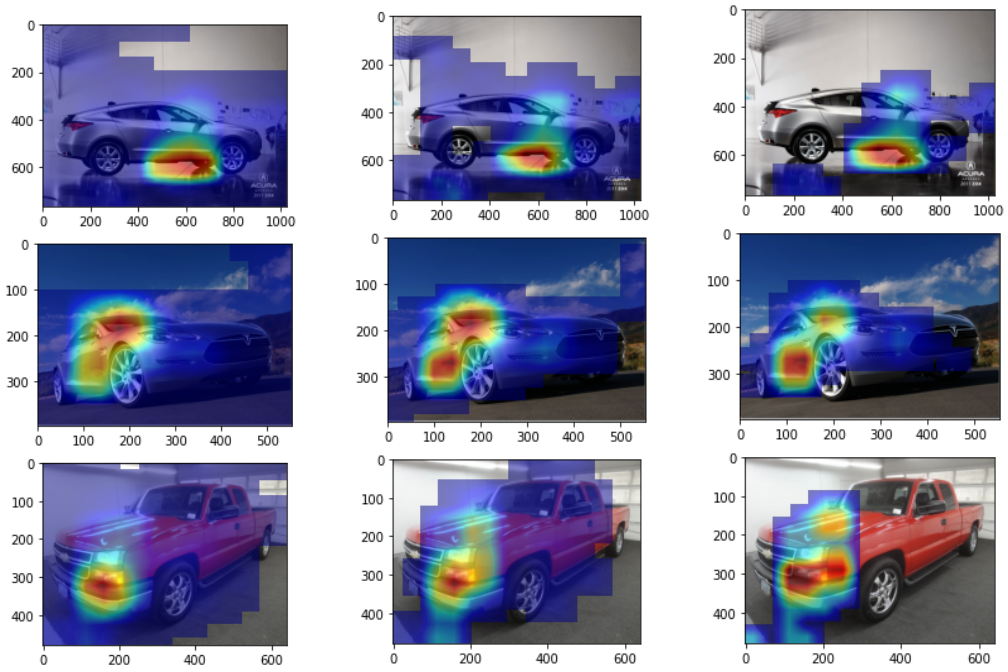
the paper, they seem to send all transactions T to an algorithm called Apriori. However, we encountered issues with this since some transactions were empty, meaning there were no activations greater than zero in the corresponding feature map. Therefore, we sent the set of transactions $\{t|t \in T \wedge |t| > 0\}$ to the Apriori algorithm instead. See Figure 4.12 for a visualization of the transaction generation pipeline.

The Apriori algorithm uses a minimum support threshold β to generate sets of items I which frequently occur from the transactions. In other words, if $supp(I) > \beta$ then it is considered a pattern in the data. Apriori iteratively grows all itemsets until no further changes can be made. In the paper, they set $\beta = 0.06$ for the Stanford car dataset, but we found that it produced noisy results and instead opted for $\beta = 0.12$.

The next step is to create the support map. To create the support map, we merge all N_I patterns found via the Apriori algorithm into the support map $S \in \mathbb{R}^{W \times H}$. Since each item i is a vector position, we must map them to coordinates on a 2D grid. We initialize the support map as an array of zeroes. Then for each position (x, y) in S, set the scalar to the frequency $f(x, y)$ of an item occurring in position (x, y). The paper defines the support map using Equation 4.10.

$$S(x,y) = \begin{cases} f(x,y) & \text{if } \exists I_j, i_{x,y} \in I_j, j \in [1, N_I] \\ 0 \end{cases} \tag{4.10}$$

Using bilinear interpolation, we upsample the support map to the same size as the input image. Finally, we remove all regions not part of the largest contiguous region. The authors do not define what constitutes a valid connection between elements. However, based on the provided figures, we assume diagonal connections to be valid and therefore use the centrosymmetric matrix J_3 to find contiguous regions. Another point of ambiguity in the paper is the order of upsampling and region filtering. They state early in the paper that they first do filtering and then upsampling but later state they upsample and then filter. We decided to implement the last order since it was described in the pseudo-code of their proposed approach. A few examples of a support map can be seen in Figure 4.13.



(a) $\beta = 0.06$ (b) $\beta = 0.09$ (c) $\beta = 0.12$

Figure 4.13: The support map for three different cars using three different values of β.

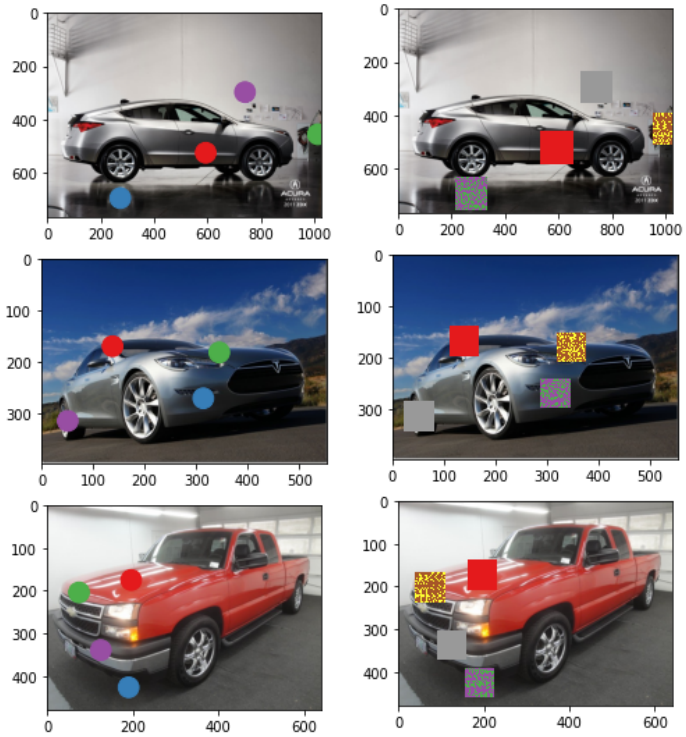
4.5.3 Detecting Parts Using The Support Map

The next step of the approach is to cluster the support maps to find part centers. In the paper, they use K-Means, but we decided to use K-Means++ for faster convergence. From the support map S we create vectors defined as $s = [x, y, S(x, y)]$ where $S(x, y) > 0$. We set the number of clusters to $K = 4$, the same as in the paper. We get the cluster centers $C = \{c_1, c_2, ...c_{K-1}, c_K\}$ from K-Means++, which we set as the spatial center of parts in the input image. See Figure 4.14 for some examples of cluster centers.

The next step is to generate part segmentation masks from the cluster centers. The part segmentation masks are generated using a spatial constraint, defined as $l = \lambda \cdot min(w, h)$, with w and h respectively being the width and height of the support map bounding box. The support map bounding box is created by calculating the extreme points $[x_{min}, x_{max}, y_{min}, y_{max}]$ where the scalars are greater than zero. We generate K masks, meaning we have the set of masks $M = \{m_1, m_2, m_{K-1}, m_K\}$ where $m \in \mathbb{R}^{W \times H}$. By multiplying a mask elementwise with the input image, we get the pixels corresponding to the part. See Figure 4.14 for some examples of part segmentation masks.

4.5.4 Applied As A Part Detector

We could not recreate the results shown in [Zha+19], with our results being significantly noisier than those found in the paper. We believe that this approach is not stable enough to be usable for our purposes. Furthermore, the results heavily depend on selecting a good value of β. Our approach will ideally not require manual tuning of parameters such as β. The parts detected heavily depend on what the base model pays attention to. We can, for instance, see that the model most heavily focuses on the space between



(a) Some examples of part centers. The size of the centers are exaggerated to increase visibility.

(b) Some examples of generated part segmentation masks

Figure 4.14: Examples of detected parts. All of three examples were generated using $\beta = 0.12$ and $\lambda = 0.25$.

the chassis and ground in the uppermost car in Figure 4.13. We fear that such erroneous foci on the part of the base model will likely be further exacerbated if we introduce an occluding element. However, the pattern mining approach will likely improve if the base model is significantly improved.

4.6 Approach C - From Activation Masking To Part Detectors

4.6.1 Introduction

The papers [ZWZ18; Zha+20] were investigated as a potential approach for generating localized part proposals. They propose integrating interpretability into DCNNs by modifying pre-trained layers into interpretable layers and then post-training the DCNN. Each interpretable layer has a regularization scheme that masks out feature maps and a layer-wise loss that explicitly guides the layers' filters to represent a part uniquely. We attempted to re-implement the approach as defined in the paper, and the following subsections further detail the paper concepts and our re-implementation.

4.6.2 The Interpretable Layer

The Regularization Masks

The interpretable layer has the task of training its filters to represent a part explicitly. This is achieved by masking the feature maps during the forward pass and generating a local loss for each filter in the backward pass. The loss is meant to guide each filter to learn a unique object part, while the masks help each filter unlearn non-part related visual cues by strengthening part regions and suppressing non-part regions in the feature maps.

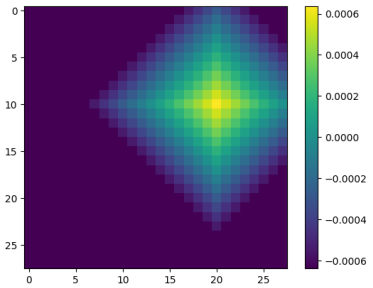


Figure 4.15: Example of a mask constructed by measuring the distance from point (10, 20) with L1 norm.

The masks in the approach are pre-defined and do not change shape. The paper constructs their masks by applying a spatial distribution of values around a position. In the paper, they use the L1 norm to construct their masks [1], but other functions are also possible. An example of a mask generated with the method from the paper is shown in Figure 4.15. One mask exists for each position i, j of a feature map, meaning there are $N = W \cdot H$ possible masks in total. The set of all masks M is defined as $M = \{m_1, m_2, ..., m_{N-1}, m_N\}$.

[1] There are also some additional calculations in the equation of the mask, which reverse the values and compress them to the range [-0.0006, 0.0006].

The Forward Pass

The overall forward pass process of the interpretable layer is at first just a standard convolutional operation. The layer takes in input $I \in \mathbb{R}^{m \times \beta \times \gamma}$ and convolves it with C filters to produce the feature maps $F_{C,W,H}$. This is illustrated in Figure 4.16. Afterward, the feature maps $F_{C,W,H}$ is processed through the ReLU function, setting any negative values in the feature map to zero.

The masks are then selected for each filter Δ by finding the position i, j of the filter's feature map with the largest activation. This maximum position is referred to as μ. This means that during the forward pass, each mask strengthens the feature map activations in the local region around the maximum position μ and suppresses the other parts of the feature maps. This is demonstrated in Figure 4.17.

The final step is to apply the selected masks by taking the Hadamard product of the feature maps $F_{C,W,H}$ and the corresponding masks $M_{C,W,H}$. Because the masks contain negative values, the result is also processed through the ReLU function such that the masked feature maps $F^{M}_{C,W,H} = max(M_{C,W,H} \circ F_{C,W,H}, 0)$. This suppresses anything, not around the maximum activation of each feature map and is illustrated in Figure 4.18. The whole forward pass process described above is also illustrated in Figure 4.19.

The Backward Pass

In the backward pass process of the interpretable layer, a local guidance loss is calculated for each filter and then combined and propagated with the normal gradient flow. For a certain number of iterations, the loss is defined as the minus mutual information between each filter Δ ReLU activated feature maps X and the assigned masks T over a certain number of iterations. The loss definition can be seen in Equation 4.11. This is used jointly with cross-entropy loss, leading to the final loss definition shown in Equation 4.12.

The primary interpretation of the loss, as given in the paper, is that it should guide each filter to represent a single part by forcing both a low categorical and spatial entropy. In other words, each filter should ideally only activate for a single category and a single location during post-training. Further information about how the loss is derived and combined with the normal gradient flow can be seen in section 3 of the original paper.

$$\mathcal{L}_{\Delta} \overset{\text{def}}{=} -MI(X;T) \overset{\text{def}}{=} -\sum_{T} p(T) \sum_{x} p(x|T) \log(\frac{p(x|T)}{p(x)}) \tag{4.11}$$

$$\mathcal{L} \overset{\text{def}}{=} \mathcal{L}_{CE} + \mathcal{L}_{\Delta} \tag{4.12}$$

4.6.3 Applied As A Part Detector

The part detectors found from this approach are the filters in the interpretable layers of a post-trained DCNN. The paper's post-training regime consists of binary or multi-

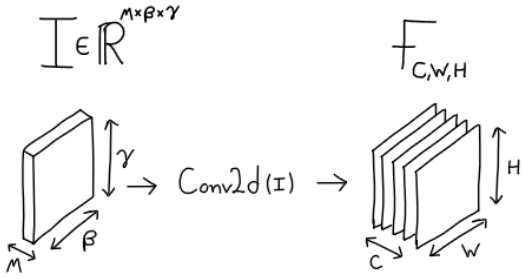


Figure 4.16: The first part of an interpretable layers operation is a standard 2D convolution.

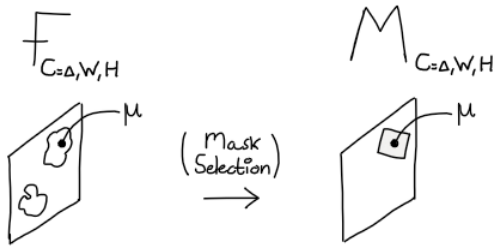


Figure 4.17: The second part of an interpretable layers operation is the selection of masks that correspond to each feature maps maximum value position.

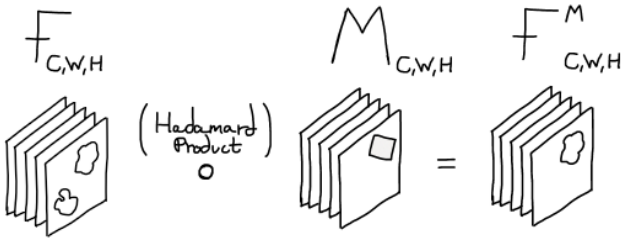


Figure 4.18: The third part of an interpretable layers operation is to perform elementwise multiplication of the masks with the feature maps, suppressing unwanted activations and strengthening the maximum activations. The result is the masked feature maps, which is also the output of the interpretable layer.

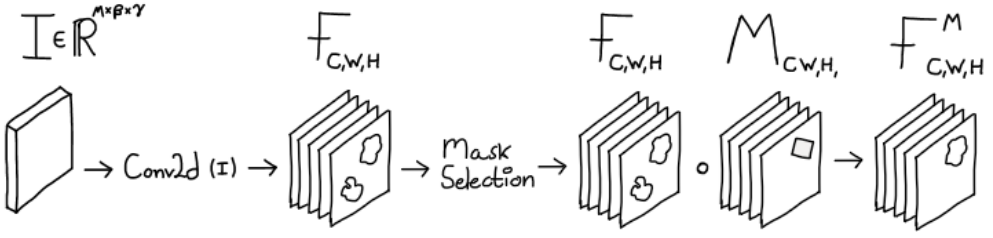


Figure 4.19: The whole interpretable layer flow from input $I \in \mathbb{R}^{m \times \beta \times \gamma}$ to output $F^M_{C,W,H}$.

category classification on a few generic datasets. The models they used were standard DCNN's[2] modified to use interpretable layers. The results in the paper are based on evaluation metrics called part interpretability [Bau+17], and location instability [Zha+18]. These metrics make it very clear that the approach shows superior performance over standard DCNNs. Regarding the classification accuracy of the datasets, the approach does not seem to show a decisive improvement. In binary classification, the standard models seem to perform better, while for multi-category classification, the interpretable models seem to perform better.

We attempted to re-create these results by implementing the interpretable layers as described in the papers and then post-training with the dataset described in section 4.3. At first, we did not manage to make the approach work, but after closer inspection, we found that the authors had released MATLAB code. This code deviates from some of the concepts described in the papers, and when we accounted for this, we managed to make our interpretable layers work.

Our general observation is that we do not find this approach directly useful for finding part detectors since it relates more to regularizing filters to match unique concepts better. The more interesting insights we got from re-implementing this approach is that filters in DCNNs already contain part knowledge. We also believe that the argmax position of pre-trained DCNN feature maps has localization capabilities since the paper successfully applied masks around this position to regularize the filters. This led us to our next approach in section 4.7, where we looked at using filters of pre-trained networks directly as part detectors. This was based on a paper cited by the Interpretable CNN authors on network dissection [Bau+17] and the Interpretable CNN's use of the feature map argmaxes.

4.7 Approach D - DCNN Filters As Part Detectors

4.7.1 Introduction

With the insights we got from implementing the interpretable CNN approach in section 4.6.1 we focused our following approach on using filters in DCNNs directly. We did this by basing our approach on the existing work on network dissection [Bau+17; Bau+20]. This form of dissection has already found evidence that neural network filters learn to represent specific concepts like objects, object parts, textures, colors, materials, and scenes. This means that in particular layers of a DCNN, we expect that it should be possible to extract part detectors.

In section 4.1 it was mentioned that for the recognition system to do more than classification, the part detectors it uses should also have the ability to localize detected parts. Because the network dissection paper successfully uses segmentation masks as ground truths when matching filter outputs to concepts, we assume that this also means that filters can localize. Interestingly, the paper also finds that batch normalization reduces interpretability by leading to fewer unique detectors. This suggests we want to avoid

[2] Specifically AlexNet, VGG-M, VGGS, and VGG-16

this form of regularization in the DCNN from which we extract part detectors.

From the paper's results, we also expect filters to behave general enough to detect parts in many-to-many relationships. However, it is unknown to what extent context affects the filter's localization capability. We assume that in instances where the receptive field of a filter matches a part closely, the filter can only be affected by the part pixels and should therefore be able to detect the part independent of context. If the receptive field of the filter is larger than the part, however, we cannot draw any conclusion as to how different contexts will affect the filter. We illustrate this concept in figure 4.20 and figure 4.21.

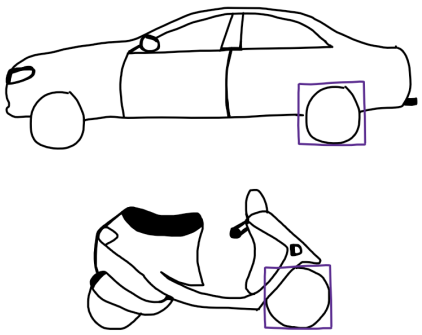


Figure 4.20: Example of a filter with a receptive field tightly around the wheels of a moped and a car. In this instance, one should be able to assume the filter will have generalized to the given concept and that it will activate similarly on both wheels.

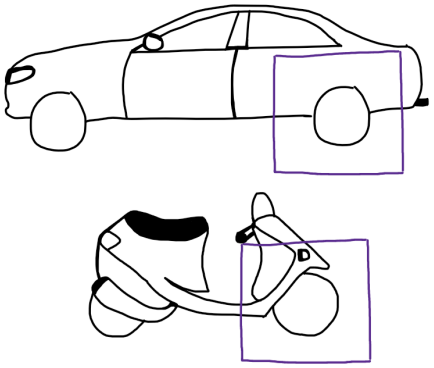


Figure 4.21: Example of a filter that has a receptive field larger than the wheels of a moped and a car. It is unknown how the context around the wheel has affected the filter output.

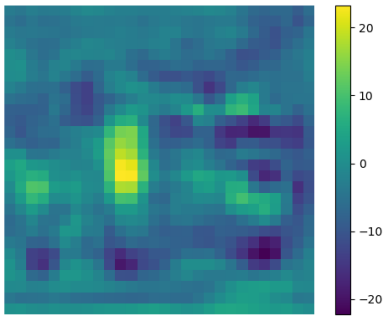
Although the dissection paper allows us to make some assumptions regarding the fact that filters learn to represent specific parts, there is no clear indication if this means they exclusively activate for only one part or if they also activate for other parts during a forward pass. This uncertainty results from feature maps not being restricted to single activation peaks, which means that a different part could also be detected at a different location. Therefore, we cannot know to what extent a filter represents multiple parts and how this affects their use as part detectors. We show an example of this in figure 4.22.

4.7.2 Finding Part Detector Filters

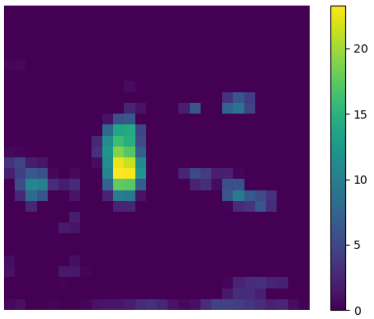
As mentioned, we replicate the dissection methodology of measuring filters to find part detector candidates. The dissection method requires segmentation masks, but our dataset did not contain such a mask. For this reason, we created a similar method which only requires class labels [3]. Our method uses two labels, where label A is defined as the object

[3] With the assumption that we have good images containing only the given object class. It is arguable how well this method would work with more challenging images.

(a) Input image used in forward pass. Image originates from the Stanford cars dataset [Kra+13].



(b) Feature map output before a non-linear activation function.



(c) ReLU activated feature map.

Figure 4.22: Example of output from filter 429 in the 'Conv 4-3' module in VGG16. The forward pass is done with the car in figure 4.22a, which produces the feature map output in figure 4.22b and the ReLU activated feature map in figure 4.22c. There is a clear activation peak at one location in the feature map of figure 4.22c, but there is also other smaller activations at other locations.

class of interest and label B as a set of other classes which are not A.

To find what filters are good part detector candidates, we want to measure the response of a filter when the target class is present versus when it is not. This means we want to measure the strength of the feature maps $F_{\Delta,W,H}$ for a filter Δ. Since we have no ground truth masks, we do not know what positions i, j of the feature map cover the object, and as a result, we cannot directly know which values of the feature map to measure.

When we implemented the interpretable DCNN approach from section 4.6, we learned that using the maximum value is a good substitute because, with the assumption that class A is the most prominent class present in an image with label A, the maximum response of a filter is likely located on the object part if the filter is a part detector for class A. This has the consequence of ignoring the weaker activations that a filter produces. We still assume that measuring the maximum response works well enough since the main part a filter finds is likely the most prominent visual cue due to the network training converging to it. We therefore end up measuring histograms of the values y defined in equation 4.13 for each filter Δ.

$$y_\Delta = \underset{i,j}{\text{maximum}}\, F_{\Delta,i,j} \tag{4.13}$$

Because we are interested in finding filters that strongly respond to images with label A, we also need a baseline for each filter when label A is not present. We use the general label B since it mostly represents objects defined as not A. We, therefore, create two different histograms, one for images of label A and the other for images of label B. In our experiments, we find that these histograms are normally distributed. On this basis, we assume that the probability density functions for these measurements are always close to being normal distributions. This lets us estimate the conditional probabilities $p(y_\Delta|A)$ and $p(y_\Delta|B)$ with maximum likelihood estimation, such that for each filter Δ we have the two normal distributions $\mathcal{N}_\Delta(\mu_A, \sigma_A^2) \sim p(y_\Delta|A)$ and $\mathcal{N}_\Delta(\mu_B, \sigma_B^2) \sim p(y_\Delta|B)$. Figure

4.23, figure 4.24 and figure 4.25 visualizes what is described above. Finding part detector filters now becomes an issue of measuring the separation between the two conditional probabilities instead. Since the probability functions represent the filter activation strength given a label, a filter that favors one label over the other will also have a large separation between the conditional probabilities. We show an example of this in figure 4.26 and figure 4.27.

We find the Bhattacharyya distance [Bha46] to be a suitable measure of separation between normal distributions as it considers both the mean and variance. The general definition of the Bhattacharyya distance for continuous probability distributions is given in equation 4.14, where the functions p and q are the probability distributions, and $x \in X$ is the shared domain of the distributions. We know that other methods exist for finding a measure of distribution similarity, for instance, the Kullback-Leibler divergence. However, we found that the Bhattacharyya distance worked well for our purposes and settled for it.

$$BD \stackrel{\text{def}}{=} -\ln(\int \sqrt{p(x)q(x)}\, dx) \tag{4.14}$$

Equation 4.14 can be derived specifically for normal distributions, which results in equation 4.15. Instead of showing this derivation we cite the existing proof A.14 provided by [Sch67, Appendix A].

$$BD_{\mathcal{N}} \stackrel{\text{def}}{=} \frac{1}{4}\ln\left(\frac{1}{4}\left(\frac{\sigma_p^2}{\sigma_q^2}+\frac{\sigma_q^2}{\sigma_p^2}+2\right)\right)+\frac{1}{4}\left(\frac{(\mu_p-\mu_q)^2}{\sigma_p^2+\sigma_q^2}\right) \tag{4.15}$$

4.7.3 Analysis Of Found Filters

To test the feasibility of our approach, we perform an analysis where we attempt to find filters that act as car part detectors by using the dataset defined in section 4.3. This means label A is for cars, and label B is for everything besides cars. We choose to extract pre-activation feature maps from Conv4-3 of a VGG16 that has pre-trained on ImageNet. We choose Conv4-3 in VGG16 because the network dissection paper [Bau+20] shows that this layer has a larger number of part detectors compared to earlier layers. Another reason for selecting this layer is that it has a receptive field small enough not to capture the entire car in the Stanford images. This ensures that we will not find any whole car object detector filters since there is no way for the filters to have seen the entire car. We want to clarify that we do not perform any training; we only use the dataset to extract feature maps by performing forward passes with VGG16.

For each of the 512 filters in the Conv-4-3 layer we extract the histogram over the argmax values y_Δ for label A and B. Then we calculate the Bhattacharyya distances $BD_{\mathcal{N}}$ by using the maximum likelihood estimated parameters μ_A, σ_A, μ_B and σ_B for each filter Δ in equation 4.15. We find that the distance values follow an exponential distribution[4]. This

[4]To clarify, this means that when looking at a histogram of the $BD_{\mathcal{N}}$ values we see that the histogram

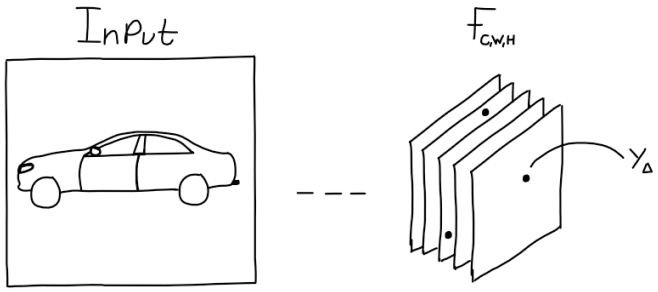


Figure 4.23: For a given image, each feature map will have a maximum scalar value at a chosen layer. We denote this as y_Δ where Δ is a filter. In the case of multiple maximums generated by a filter, we select the first maximum as the sorting algorithm sorts it. In other words, the selection is arbitrary.

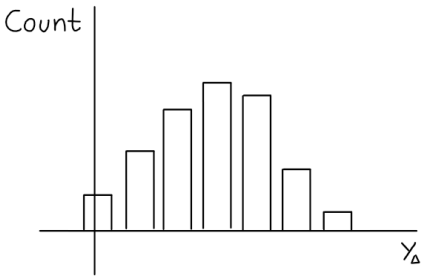


Figure 4.24: For each filter Δ, we collect a histogram of values y_Δ for a given label. Our observation is that this data is normally distributed.

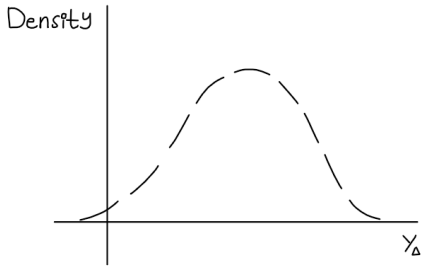


Figure 4.25: Given a histogram as in figure 4.24 we can use maximum likelihood estimation to estimate the probability density function as a normal distribution over the observed values.

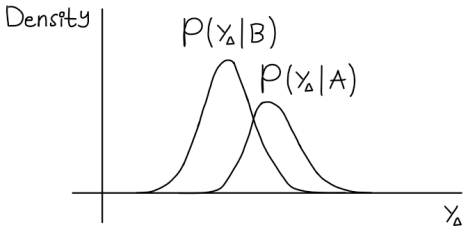


Figure 4.26: A filter Δ with overlapping conditional probabilities.

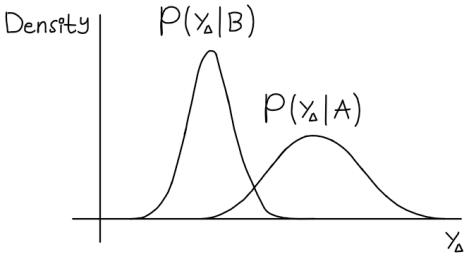


Figure 4.27: A filter Δ with more separable conditional probabilities in comparison to figure 4.26.

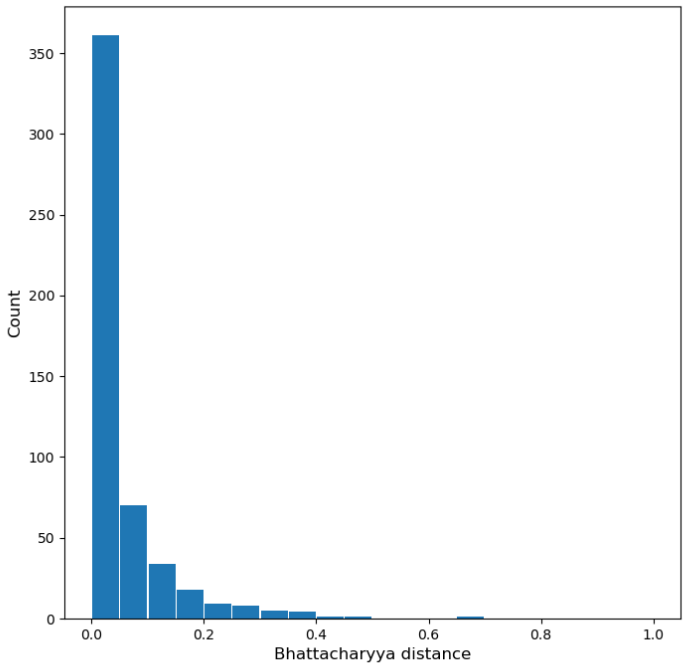


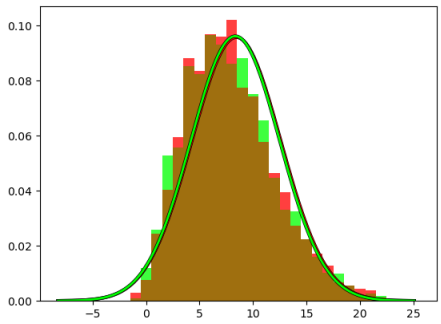
Figure 4.28: The histogram of the 512 Bhattacharyya distances extracted from the 512 filters of Conv4-3 in VGG16. The distances are calculated using the parameters for each filter (μ_A, σ_A, μ_B, σ_B) in equation 4.15.

distribution shows that a few top K filters has a large distance, while a majority of the other filters has a low distance. Figure 4.28 shows a histogram of the $BD_{\mathcal{N}}$ values. Figure 4.29 shows an example of the y_Δ density histogram and estimated normal distribution for filter 341 with a low $BD_{\mathcal{N}}$ value and filter 429 with a high $BD_{\mathcal{N}}$ value.

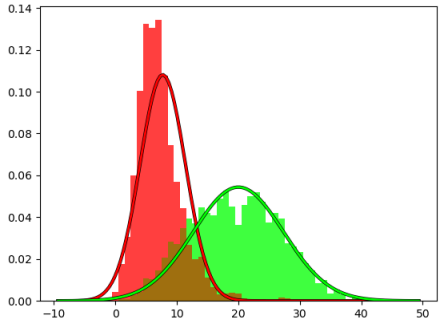
We further investigate what filters 341 and 429 localize by using interpolation like the network dissection paper. We specifically use linear interpolation on the ReLU activated feature maps and then set the normalized interpolated feature maps as the alpha channel of the input images. This can be seen in figure 4.30 for filter 341 and figure 4.31 for filter 429. We observe that the filters with a high $BD_{\mathcal{N}}$ value localize parts on cars with strong activations. Filter 429, for example, seems to activate on the curved arch between the car wheels and body. This also seems to generalize for the non-car image arches, as seen on the clock image row in figure 4.31. Since we do not have segmentation masks, we cannot empirically confirm to what extent a high $BD_{\mathcal{N}}$ value correlates with filtering localization. We also find examples of low $BD_{\mathcal{N}}$ value filters that sometimes localize on cars, but this appears more random and does not have a strong enough feature map strength to explain when the filter sees a car part in compared to something else.

It should also be noted that the ordering of the distributions matter. We find that there are cases where filters are more inclined to activate for the baseline, which means that

follows a exponential distribution. This means there are many filters with low distances and a few filters with large distances.



(a) Example of the A and B histograms and estimated normal distributions for filter 341 with $BD \approx 0$. The red normal distribution is not visible since the distributions overlap almost completely.



(b) Example of the A and B histograms and estimated normal distributions for filter 429 with $BD \approx 0.67$.

Figure 4.29: The difference between a filter with a low BD value and a filter with a high BD value from figure 4.28. The green represents class A (car) while red represents class B (baseline). The brown represents the common part of the two histograms.

distribution B is to the right of distribution A. These cases can also have high $BD_{\mathcal{N}}$ values, but we did not attempt to use these filters. We avoid using these filters by simply adopting a direction parameter where $\mu_A > \mu_B$ is a positive direction and $\mu_A < \mu_B$ is a negative direction, and then only using the filters with positive direction.

4.7.4 Applied As Part Detector

With the insight that our approach seems to find part detector filters that strongly activate for car parts, the next question that arises is how a decision procedure can translate filter outputs into a detection. When measuring the pre-activation feature maps, each filter Δ has a unique range interval between $(-\infty, \infty)$. A decision procedure must therefore be able to relate each filter's unique range to a detection. This can be done by setting a soft or hard decision boundary over the already estimated normal distributions $\mathcal{N}_\Delta(\mu_A, \sigma_A^2)$ and $\mathcal{N}_\Delta(\mu_B, \sigma_B^2)$.

Setting a hard decision boundary comes down to the question of what error rate is acceptable. The most obvious threshold lies at the intersection of the two conditional probabilities, but the boundary can also be moved for a stricter or more lenient decision. The output of this process is a binary map because it transforms the output of each filter into a pixel-wise decision of part or non-part after thresholding. To further understand how a hard threshold can be set from conditional probabilities, we looked into the book [The89]. This leads us to the Log-likelihood ratio, which lets us find a threshold for all filters from a common ratio value. The ratio is defined in equation 4.16, where P and Q are likelihoods. The simplest threshold is to find the normal distribution intersections, which means solving for $LR = 0$ when the prior likelihoods are equal[5]. There are generally two such intersections, but because we only want large activation strengths, the rightmost

[5]This is the case for our normal distributions because we have a balanced dataset of 50% cars and 50% non-cars.

Figure 4.30: Localization examples for filter 341 which has a low BD value. Each row corresponds to one example. The first column is the input image, the second column is the ReLU activated feature map corresponding to the input image, and the third column is the input image where the interpolated and normalized feature map is set as the alpha channel.

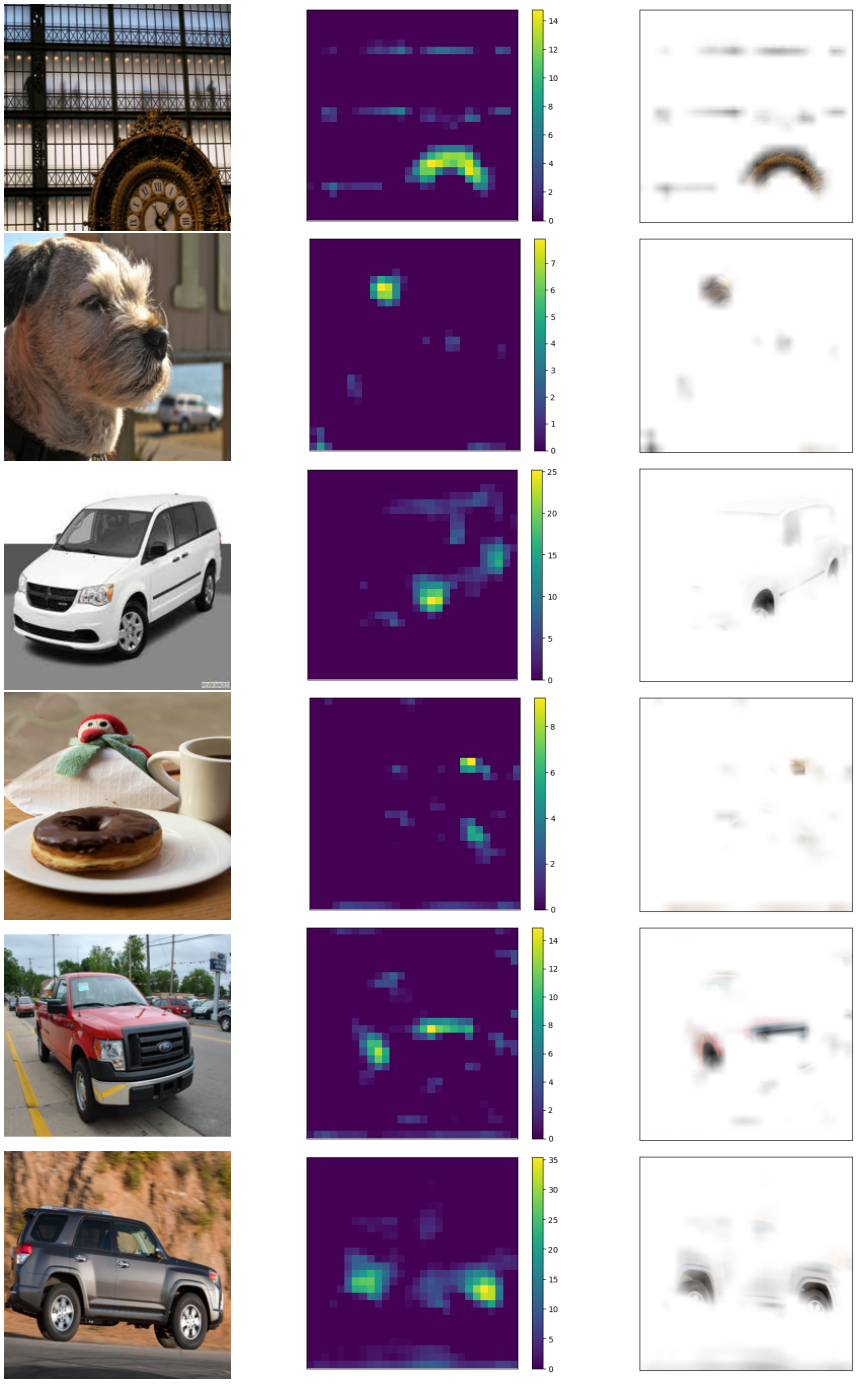


Figure 4.31: Localization examples for filter 341 which has a low BD value. Each row corresponds to one example. The first column is the input image, the second column is the ReLU activated feature map corresponding to the input image, and the third column is the input image where the interpolated and normalized feature map is set as the alpha channel.

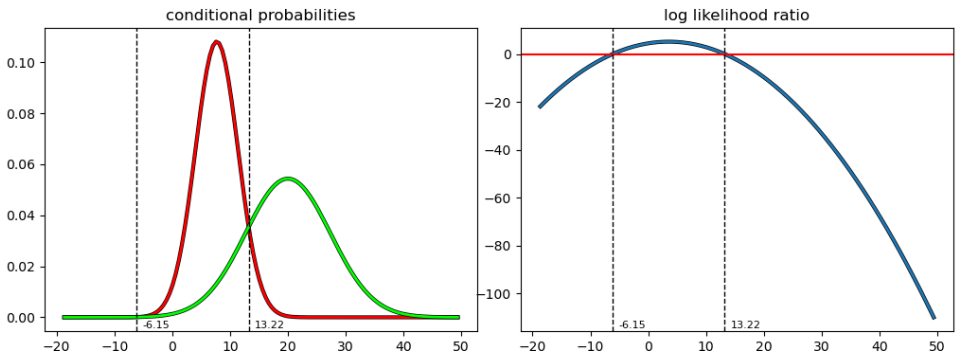
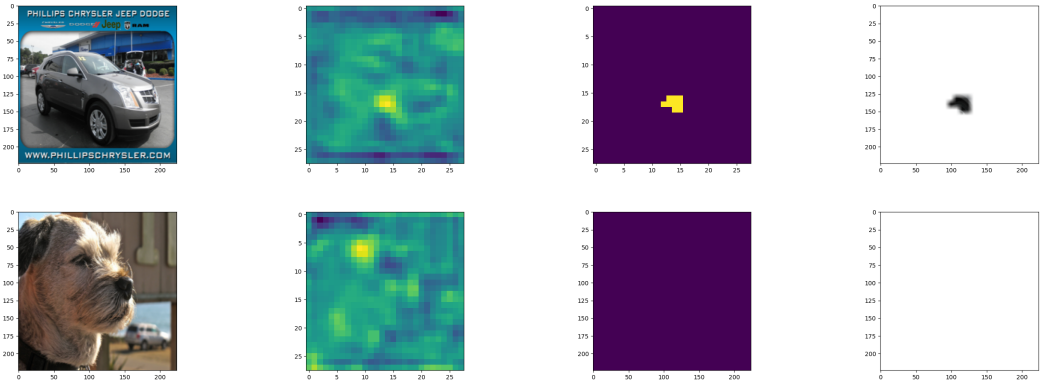


Figure 4.32: The estimated conditional probabilities and corresponding log likelihood ratio (blue curve) for filter 429. When solving the log likelihood ratio for 0 we find the solutions -6.15 and 13.22, which corresponds to the two intersection points of the conditional probabilities. The rightmost solution can be applied on the filters feature maps to generate binary decision maps.



(a) The original images.

(b) The pre-activation feature map produced from forward pass with the images from Figure 4.33a.

(c) The binary maps produced from thresholding the images from Figure 4.33b with $t = 13.22$.

(d) The images from Figure 4.33a with the binary maps in Figure 4.33c set as the alpha channel after linear interpolation.

Figure 4.33: An example of using the threshold found from solving $LR = 0$ for filter 429 on both a car and non-car image.

intersection is the only one we choose. Solving for a smaller ratio value will lead to a stricter decision, while solving for a larger one will do the opposite. Since there are generally always two solutions when solving for any ratio, we always choose the rightmost one.

$$LR \overset{\text{def}}{=} -2\ln(\frac{P}{Q}) = -2(\ln(P) - \ln(Q)) \tag{4.16}$$

As an example, we also calculate the log-likelihood ratio for filter 429. We do this by taking the conditional probabilities from figure 4.29b and calculate the corresponding ratio which can be seen in figure 4.32. The threshold found from this ratio is ≈ 13.22, and we show what binary decision map this threshold produces for a car and non-car image in Figure 4.33. Our observation from this form of thresholding over different filters with a high BD value is that it can accurately localize the filter's most prominent car part visual

cues and, for many non-car images, completely threshold away non-part cues. Since the argmax was used in the measurement, there is the issue of thresholding away minor filter activations that might also correspond to cars, but we still find that the most prominent filter part is kept.

Another decision boundary can be used is a soft function that transforms filter outputs to a probabilistic output fixed between $[0, 1]$. There already exists commonly used activation functions[6] that do something similar. However, because we want a statistically grounded decision when transforming the outputs, we find that using a learned logistic function is the most ideal. Learning such a function is easily done through logistic regression over our already existing dataset with the maximum y_Δ values and their respective label A or B. As an example we do this for filter 429, and show the resulting logistic function that was fit over the data in Figure 4.34. By using the fit logistic functions as activation functions over the pre-activation feature maps of each filter, we get a probabilistic output instead. We also show examples of this for filter 429 in Figure 4.35 for car and non-car. Since there is no hard thresholding with this method, the issue of removing potential useful minor filter activations is avoided. This can be seen by comparing how the hard and soft thresholding activations differ. For filter 429, we generally see that the curve over the wheels is very prominent, but it also has some other activations on cars that are lost when using a hard threshold.

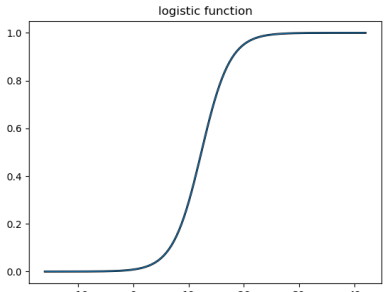
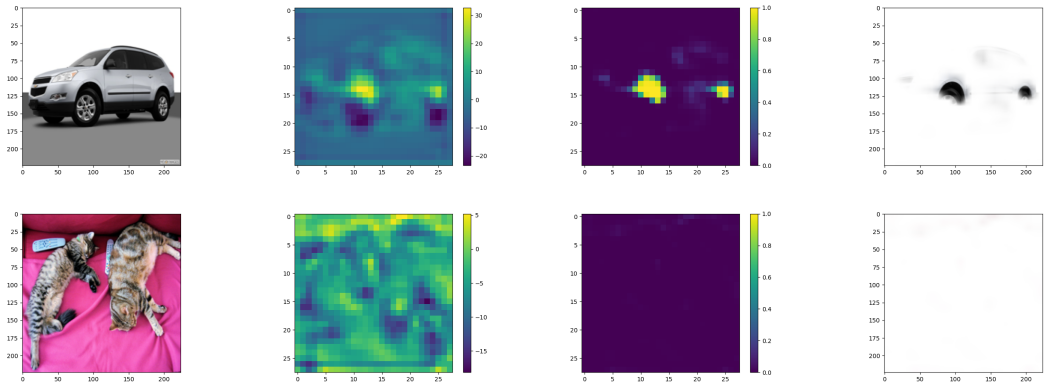


Figure 4.34: The logistic function fit to y_{429} for filter 429.



(a) The original images.

(b) The pre-activation feature maps produced from forward pass with the images in Figure 4.35a.

(c) The probability maps produced from activating the feature maps in Figure 4.35b with the fitted logistic functions shown in Figure 4.34.

(d) The images from Figure 4.35a with the probability maps in Figure 4.35c set as the alpha channel after linear interpolation

Figure 4.35: An example of using the fit logistic function for filter 429 on both a car and non-car image.

[6] relu, sigmoid, leaky relu, selu, tanh, and so on.

Chapter 5

Building A Joint Structure System

This chapter is dedicated to detailing our efforts in developing a joint structure system. In section 5.1 we outline some methods for modeling part-to-object and part-to-part relationships. In section 5.2 we propose methods for further processing the output from the part detectors described in section 4.7. The final sections describe different approaches of creating joint structure classifiers.

5.1 The Concept Of Part Compositionality

In chapter 4 we described how one might construct part detectors for a class. The next step is aggregating the part detections found via the described part detectors and producing object-class hypotheses. This process builds upon the concept of object-to-parts compositionality.

We investigate how one might model and leverage the relationship between the different parts constituting an object and between the parts and the class of interest. It is important to note that we do not operate directly on the semantic parts, but part detector hits. From now on we will refer to the part detections in the map generated by the approach described in section 4.7 as *part hits*. Furthermore, any use of the word part should be understood as referring to part hits unless otherwise stated. However, many of the concepts generalize to fully-fledged parts as well.

The primary avenue we pursue concerning modeling part-to-part relationships is spatiality. In other words, we investigate how to model the relative position of one part to another. The part-to-object relationship is already represented by the probability generated via the logistic regressor we described in subsection 4.7.4. Still, we also explore other approaches which can be applied to model this relationship. The following subsections should be understood as a summary of the reviewed methods. In other words, we did have sufficient time to analyze all the presented avenues of approach empirically. However, we hope that by offering these methods, we can give the reader an idea of the breadth of ideas which can yield an adequate solution to the problem at hand.

The reader should note that we are primarily interested in creating a model of how part

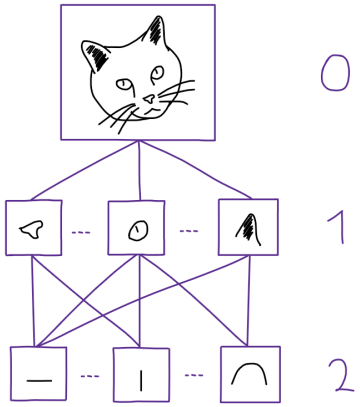


Figure 5.1: A simplified tree representation of the part hierarchy that forms an object.

hits directly relate to an object and each other. In other words, we do not pursue an approach based on explicitly constructing a hierarchical structure that defines a relationship from low-level features, such as edges, up to and including high-level object definitions. However, a hierarchical representation of sorts is present in DCNNs, demonstrated in [Yos+15]. Therefore, a hierarchical model from low-level features to high-level ones is implicitly represented in the investigated DCNN-based approaches. Thus, the level of abstraction we are pursuing can be understood as level 0 and level 1 in a tree of objects and parts. This concept is illustrated in figure 5.1. However, we have introduced several papers that propose bottom-up-representation approaches in chapter 3.

5.1.1 Modeling The Part-To-Object Relationships

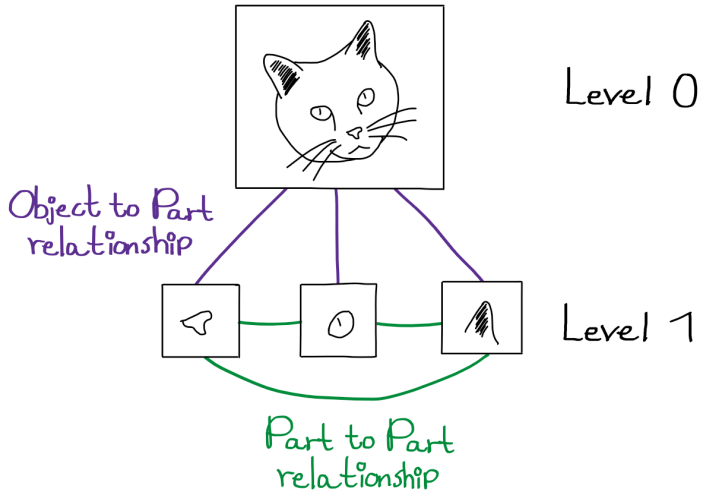


Figure 5.2: A visualization of the part-to-object relationship and part-to-part relationship as edges between the nodes in a tree-like structure. The part-to-object relationship can be thought of as moving between the abstraction levels, while the part-to-part relationship is between nodes on the same level of abstraction.

The principal aspect to consider when classifying an object in a compositional manner is the part-to-object relationships. The concept of part-to-object and part-to-part relationships are visualized in Figure 5.2. We have already described how one may use the fea-

ture map activation distributions to model this relationship in section 4.7 and will discuss more in-depth how this might be processed further in section 5.2. We first consider how one might obtain other models of part-to-object relationships. The reader should note that we focus solely on the enumerative part-to-object relationship. In other words, we look at models where the count of a part class A is used as a foundation for inferring the object class. However, we are aware that it is also possible to model spatial part-to-object relationships. One hypothetical way of modeling the spatial part-to-object relationship is to learn the relative offset of all parts to the bounding box edges. The relative offset could be used in several ways, for instance, to estimate the bounding box from a set of detected parts. Of course, this presupposes access to object-level labeling in the training set.

Estimating The Conditional Probabilities

The first approach we consider is finding the frequency of occurrence for a given part A_p when we are presented with an object of class A_c. Assuming we have access to class labels, we can count the number of times A_p is present in a sequence of parts P detected from an object of class A_c. Dividing this count R with the number of object samples presented to the part detectors results in an approximation of $P(A_c|A_p)$, which we can consider a model of the part-to-object relationship between A_p and A_c. By extending this algorithm to all other relevant parts, we have a complete part-to-object relationship model, which can be applied in probabilistic models such as Naive Bayes, an approach we describe and test in section 5.5. One flaw of this approach is that it requires hard part assignments due to the part counting. Naturally, this causes information loss since the filter-based part detectors produce soft part assignments. Furthermore, a probability threshold must be chosen to determine what constitutes a part being present in the object.

Modeling The Relationship Via Part Counts

Another approach is to model the part-to-object relationships using a vector definition. Suppose we have a set of part detections P detected in some object A_c. We can then define the object as a vector $d \in \mathbb{R}^{|P|}$, where $|P|$ is the cardinality of P. The components of the vectors can be defined in several ways. A simple algorithm is to use the part counts and set the corresponding vector component's scalar to that count value. By generating a set of such vectors D and using our class labels, we can fit, for instance, a Support Vector Machine (SVM) to the set of vectors and labels. The SVM will learn the part-to-object relationships in fitting to D. Of course, due to the black-box nature of SVM, this would counter our stated goal of maintaining a level of explainability concerning the object-class hypothesis. However, one should note that research is being conducted into making white-box variants of models like SVM [Deo+21; SG20]. Therefore, it may represent a viable approach in the future, so presenting this is done for the sake of completeness.

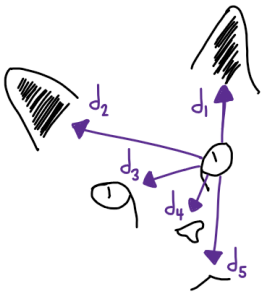


Figure 5.3: An illustration of how the relative position of a part can be defined in terms of a displacement vector d_n.

5.1.2 Modeling The Part-To-Part Relationships

We will, in this subsection, consider how we might model and leverage the data found in part-to-part relationships. More specifically, we will consider how to utilize the spatial part-to-part relationships. We define spatial part-to-part relationships as representing the positional co-occurrence between one or more parts. For instance, if one considers a part class B which frequently occurs in a location relative to part class A defined by some displacement vector d, one has captured an aspect of the spatial relationship between part classes A and B. This concept is illustrated in Figure 5.3. Intuitively, this is particularly useful for classifying object classes where relative movement between parts is limited, such as cars or boats. However, spatial modeling should be helpful even for articulated object classes, such as jointed objects like cats or other animals. One should note that the spatial part-to-part relationships are likely to be more complex if relative movement is possible.

All of the methods described in subsection 5.1.1 represent BoVW approaches to the problem of classifying an object on a compositional basis. Therefore, it is self-evident that we discard information that can be valuable since we only consider data related to the part-object relationships. It is conceivable that false positives will occur if several disjoint parts are detected in an input image. For instance, a BoVW approach will classify all images in the sequence in Figure 4.6 as a cat. One may reasonably argue that such examples are somewhat redundant because it is unlikely that the detector will ever be required to handle such inputs. However, suppose one instead considers a detector that is to perform classifications on objects where the inter-class differences are more subtle. In that case, it is not given that only considering the part-to-object relationships are sufficient to obtain satisfactory levels of classification accuracy. For example, let us consider a detector that detects trucks and sports cars. It is reasonable to assume that there will be a significant intersection between the part dictionaries of the two classes. Hence, it is natural to consider how one might utilize the part-to-part relationships, such as relative part-to-part positions.

Spatial Extension Of BoVW

Research has been conducted into creating an extension of BoVW that considers the input feature's spatial information. For instance, the paper [YN10] proposes using a spatial co-occurrence kernel to consider spatial information in combination with a histogram BoVW approach. The co-occurrence kernel is derived by finding the intersection of several co-occurrence matrices. The matrices contain the frequency of occurrence between a pair of features defined by some spatial predicate. They note that modeling all combinations in one matrix requires $D \times D$ elements, which would be prohibitive for an extensive dictionary. They solve this by considering a dictionary subset and creating several smaller matrices, which are then used to generate the co-occurrence kernel.

The previous paper takes inspiration from an earlier approach called Spatial Pyramid Matching (SPM), which was first applied in the same context in the paper [LSP06]. The core concept is the use of grids to create more and more fine-grained subdivisions of an input. One histogram is generated for each of the grid elements. The resulting histograms are combined using an SPM kernel which weighs the histograms according to the level to which the histogram belongs; more fine-grained histograms are assigned greater weights.

The paper [ZM10] proposes three different spatial extensions to BoVW. The three variants are based on an overlapping sliding window and a histogram to encode the information captured by the sliding windows. The first of these approaches captures information about features of a given part class B neighboring the central feature of class A of the sliding window. A count is produced for each such part-pair combination and is finally represented by a vector $r \in \mathbb{R}^{|C|}$, with $|C|$ being the cardinally of the set of parts C.

The second approach in [ZM10] is based on encoding information about the spatial distribution of neighboring parts of part class B to the center part A. This is encoded by creating a bit string of length L, where each position of the bit string indicates the presence of a part in the corresponding window position. The mapping between the window and string is done by flattening the window from left to right and top to bottom. The bit string can be considered a part shape configuration encoding. The number of each configuration encodings in the input are counted and represented by a vector $r \in \mathbb{R}^{2^L}$, with the scalar values of the vector being set to the counts. This bit string representation preserves the spatial configuration of parts B when A is the center. It is trivial to redefine this representation as a count of a part B at a position i relative to part A by simply summing the bits at the same bit position in the different bit strings.

The third method in [ZM10] is based on a more fine-grained encoding representing the distance and the direction using 32 different segments. The distance is defined using four co-centric circles, while the directions are defined using eight diagonals. The part-part relationship is encoded by placing it in a histogram consisting of 32 bins. The authors achieved the best results using approaches one and three.

Graph CNN Approaches

Another way to view a set of parts P detected via part detectors is as a graph. The paper [Xu+19] proposes an architecture based on processing graph inputs. These inputs can improve the original feature proposals made by the network that the proposed architecture is attached. To incorporate spatial information, they utilize an architecture module they name a Spatial-aware Graph Reasoning Module, essentially a graph convolutional network that enhances the head networks proposals. The paper [KPR20] proposes a somewhat similar approach because they create a module based on a graph convolutional network, which improves the output from some external network. Therefore, these two approaches can be considered a way to increase and enhance the amount of information represented by the feature map encoding.

However, these approaches do not inherently offer an improvement in terms of interpretability. Still, it would likely be possible to create an extension of these methods, which increases the level of interpretability. This assertion is based on our experience working on the outputs from regular CNNs; we were able to extract part information from CNNs. Therefore it is conceivable that similar approaches will work for the enhanced feature maps produced by the abovementioned methods. However, we can not speak to the possibility of extracting or in some way explicitly representing the spatial information the networks discussed above encode in the feature maps.

Relative Position Descriptors

Another approach to modeling the spatial part-to-part relationships is to adapt the work on Relative Position Descriptors (RPD) to the problem at hand. The review paper [NM15] defines RPDs as ”[...] a quantitative representation of the relative position of two spatial objects”. A plethora of methods has been developed to capture a representation of the relative positioning between objects, several of which are presented in the previous paper. Therefore, it is beyond the scope of this thesis to give a thorough introduction to the research topic. Still, we will attempt to outline some of the general ideas and concepts in this section with a foundation in the review presented in [NM15].

The paper outlines four different groups of spatial elements one can use to model spatial relationships between objects: pixels, points, segments, and cores. A point is a generalized definition suitable for both raster and vector inputs, but it is equivalent to pixels in the raster case. A segment is a subset of a set of pixels that combined represents an object A. A core is a region of A intersected by some chosen line. Furthermore, some methods are based on selecting elements from the boundary region of an object, while others consider the entire object region. Since we operate on point representations of parts found in an input image, the methods based on pixel or point elements are the most directly applicable. All the approaches described hold in common the usage of histograms to represent captured spatial information. For instance, an angle histogram captures the angles between pixels belonging to two different objects and, therefore, represents the spatio-directional relationship between the objects. Some of the approaches use several

histograms to capture information. An example is the Allen histogram, a tuple consisting of 13 F-histograms.

While the methods generally appear to hold some promise concerning modeling the part-to-part spatiality, it is apparent from the definition of RPDs that the methods primarily capture object-to-object relationships. Therefore, it is not immediately obvious how laborious it would be to adapt these approaches from object-to-object to part-to-object. Consequently, we decided against pursuing these approaches beyond a surface-level literature search due to time constraints.

Point Pattern Analysis And Matching

We also looked at research done on Point Pattern Analysis (PPA) and Point Pattern Matching (PPM). PPA is in [BP22] defined as being "[...] concerned with describing patterns of points over space, and making inference about the process that could have generated an observed pattern". Using this definition, PPA can generally be viewed as a toolkit of approaches applied in fields where an understanding of spatial processes is required, such as geographical epidemiology and ecology [Gat+96; WM13]. However, we were unable to find papers tackling this topic in the context of computer vision. PPM is defined in [LMH03] as an "[...] approach for establishing a correspondence within two related patterns". Furthermore, the paper states that PPM is a group of approaches applicable to solving problems related to computer vision.

In our view, PPA should primarily be considered as a way to perform quantitative analysis of the spatial processes that produce the part patterns we get from the filter-based part detectors. PPA, for instance, offers tools to measure the spatial randomness of parts via the Poisson point process. Thereby, one can analyze how a pattern deviates from being a random process and select the patterns with the most significant deviation. The intuition is that a part pattern deviating strongly from a random spatial point process is a better class indicator than one closer to random. Several other tools are, of course, available through PPA, but we will refrain from presenting others not to exceed the scope of this section. We view PPA as a group of ancillary approaches which can enhance the empirical evaluation of the spatial point patterns inherent in the part detections.

PPM encompasses a group of methods that allows us to match part patterns with a level of robustness against lacking pattern segments and other spatial perturbations. For instance, the paper [Cha+97] proposes a PPM approach invariant to translations, rotations, and scale changes to different point patterns. In other words, PPM encompasses methods that can be useful in robustly recognizing recurring patterns. Overall, the methods that PPM represents were deemed unsuitable for the task at hand due to the need to capture and store point patterns on which to match against. However, if one were to create such a dictionary of patterns, then PPM could be a viable approach and warrant further evaluation.

Part Cliques

We define part cliques as a relationship between two part classes defined by some relative spatial displacement, a simple example shown in Figure 5.3. We limit ourselves to part hit cliques in the form of diads[1], but it is possible to construct more complex cliques. Therefore, any future references to cliques should be understood as part hit diads, but many of the concepts can potentially generalize to more complex forms of cliques.

One can define an arbitrary number of different spatial displacements. Part of the motivation for using part cliques is that one explicitly captures the spatial relationship between parts given some spatial offset. Therefore, one gains a more fine-grained definition of the spatial part-to-part relationship than methods based on grid separation of parts in the input image, for instance, spatial pyramid matching, which was mentioned earlier in section 5.1.2.

One issue related to part cliques is the selection of spatial displacements. As mentioned in the previous paragraph, a significant number of displacements can be selected. Furthermore, one also has to decide between which parts to model relationships. Ideally, one selects the pairs of parts and displacements representing the relevant class's consistent visual aspects. While it is conceivable that one can perform a brute-force search for the optimal solution for a small part dictionary, this will quickly become computationally intractable for an extensive part dictionary. Therefore, one must consider efficient methods for deciding on good part-pairs and displacement selections. However, if the only viable approach proves to be computationally expensive, one can attempt to perform dimensionality reduction to reduce the size of the part dictionary. This idea is tangentially explored in section section 6.2.2 using PCA and other approaches.

Spatial Co-Location Pattern Mining

Due to the abovementioned issues concerning part cliques, we decided to look into spatial co-location pattern mining (SCLPM). The paper [Yao+18] states that SCLPM is "[...] employed to identify a group of spatial types whose instances are frequently located in spatial proximity". In other words, applying SCLPM would represent an avenue for discovering which parts frequently co-occur given some displacement threshold.

One issue with SCLPM is the selection of a displacement threshold, which effectively precludes spatial displacements based on a static vector. Therefore, the displacement will instead have to be defined as a boundary region, for instance, a circle that would produce a disk region of valid displacements. Assuming we use this form of spatial displacements, we should be able to discover good part pair combinations to use in part cliques. Due to time constraints, we could not pursue this line of research to completion.

[1] A diad is defined as being a pair of part hits

5.2 Processing The Part Detections

In this section we elaborate on how to further combine the outputs from the part detectors described in section 4.7. We want our part detector outputs to be probabilistic, so we base our processing on using the fit logistic functions as activation functions when generating part detections. In subsection 5.2.1 we describe a simple heuristic for creating a combined probability map that reduces the part detections into a single probability for each position. The problem with our heuristic is that it does not have a probabilistic meaning, so we also consider other more advanced avenues of how part detector outputs can be combined into a single hypothesis without ruining the probabilistic nature in subsection 5.2.2.

5.2.1 A Heuristic Approach To Forming A Probability Map

To describe our heuristic combination, we start by using the already established top K^2 filters based upon the Bhattacharyya distances from subsection 4.7.3. Using these filters, we can investigate how a merging of hypotheses behaves. If the filters we have found are performant part detectors, we expect to see that a merging of filter outputs will produce a form of segmentation, for example.

Our suggestion of a heuristic for creating a merged representation is the following. We first activate the top K filters and apply their designated logistic regressor. This transforms the feature maps $F_{K,W,H}$ into probability maps $P_{K,W,H}$, where $K \subset C$. After this, we combine the probability maps $P_{K,W,H}$ into a single segmentation map $S_{W,H}$ by taking the channel-wise maximum as defined in equation 5.1.

$$S_{W,H} = \underset{K}{\text{maximum}}\, P_{K,W,H} \tag{5.1}$$

The equation represents the selection of the largest probability value from the channel depth for each pixel. It is important to note that this form of heuristic does not have any inherent probabilistic meaning behind it, as it does not aggregate the probabilities produced by each filter. We illustrate the combination process in figure 5.4. We perform the maximum combination for some random examples, as shown in figure 5.5.

By qualitatively evaluating a few different examples, some general behaviors become apparent. Most of the time, the strongest activations are tied to wheel and window edge-related parts. The filters struggle more with viewpoints that do not contain such visual cues. They also seem to struggle with specific car colors like black, which could be the result of black wheels being harder to distinguish. We also see that the door region between the car wheels is not detected. However, this is expected because, as the network dissection paper has found, the type of filters that would detect this region is located in earlier layers. A better combination of filters is very likely to exist for this reason.

[2]Specifically, we use the top 27 filters. These represent the 95th percentile values of the exponential distribution from figure 4.29. Keep in mind that we only used the positive direction filters, which means 27 filters are left after pruning negative direction filters in the top 95%.

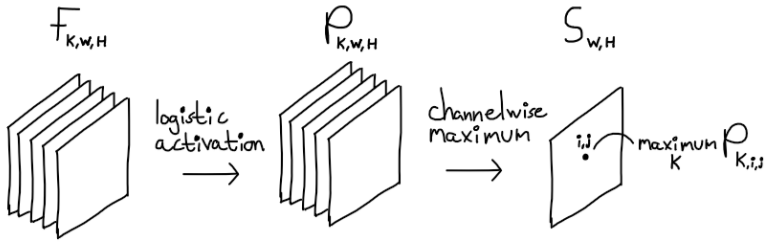


Figure 5.4: The combination process illustrated.

For example, a combination that uses filters from many different layers so that multiple concepts like textures, colors, and parts are all detected together. It is also likely that the dataset of A labeled images plays a significant role in determining what filters are suitable detectors. Instead of creating a general dataset of cars, it could be better to have extra labels based upon viewpoints and specific car sub-categories and then find good detector filters for each label. We further discuss all of the aforementioned in section 6.2.

5.2.2 Aggregating Probabilities To Form A Probability Map

It is important to note that although we only used a simple heuristic to combine the filter probabilities, other more advanced methodologies exist that doexist that do not ruin the probabilistic nature. The paper [ACR12] gives an insight into possible methods that can be used, and we do not attempt to implement these but briefly refer to this paper for future work. The paper mainly separates between aggregation methods that are either additive or multiplicative.

The first group of methods is based upon Linear pooling. This is just an additive method that takes the weighted sum of probabilities, meaning each probability has a corresponding weight that must be set. The constraints are that each weight must be in the range $[0, 1]$ and that all the weights sum up to 1. There are also extended versions, one of which is a method called Beta-transformed Linear pooling which uses a beta distribution to transform the weighted sum. Another version is Log-linear pooling, which differs from standard Linear pooling because it takes the logarithm of probabilities before using them in the weighted sum. This makes it a multiplicative method instead of an additive one since it uses the logarithmic property $\log(x) + \log(y) = \log(x \cdot y)$. There are also other multiplicative methods presented. One is an optimization scheme based upon distance measures such as Entropy or Kullback-Leibler. Another is a model approach where either the Tau or Nu models are used examples.

The interesting takeaway from the paper is the statement that aggregation is domain-specific. This means that selecting an aggregation method requires a much more in-depth investigation beyond what we had time for. There is also the question of finding the weights used in aggregation. The different options are setting them manually from expert opinion or estimating them from optimization. In our judgment, optimization is the most reasonable since we have access to training data.

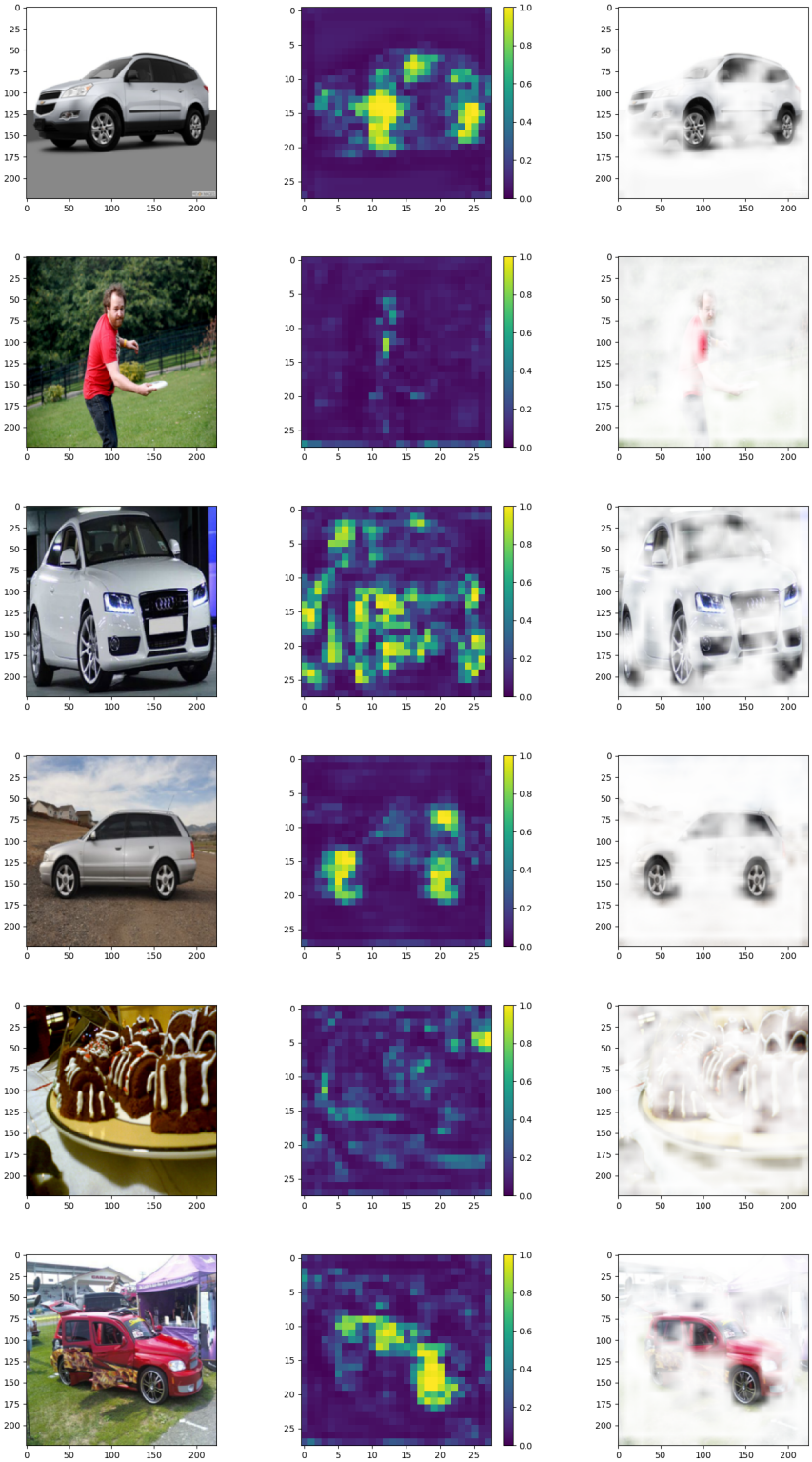


Figure 5.5: Examples of the combined probability maps. Each row is a unique example. The first column is the original image, the second column is the probability map combination, and the third image is the combination interpolated and used as the alpha channel of the original image.

5.3 Approach A - Bag Of Words Baseline

5.3.1 Introduction

It is useful to start with a simple and understandable baseline to evaluate different joint-structure classifiers. In so doing, it becomes simpler to evaluate the effects of additions. With this in mind, we choose to use a Bag of Words approach since this is a well-known and simple classification method.

We already introduced Bag of Words in chapter 2 but will give the reader a brief reminder here. Bag of Words is an approach that builds upon counting features in training examples and then learning to classify histograms of features. In the context of part detections, the approach can be viewed as leveraging the fact that certain parts should repeat for specific objects. This idea is illustrated in figure 5.6. Due to the nature of bag of words we will not be able to account for spatial relationships in the baseline approach. However, we see this as a positive since it will let us compare the difference between considering spatial relationships versus not doing so.

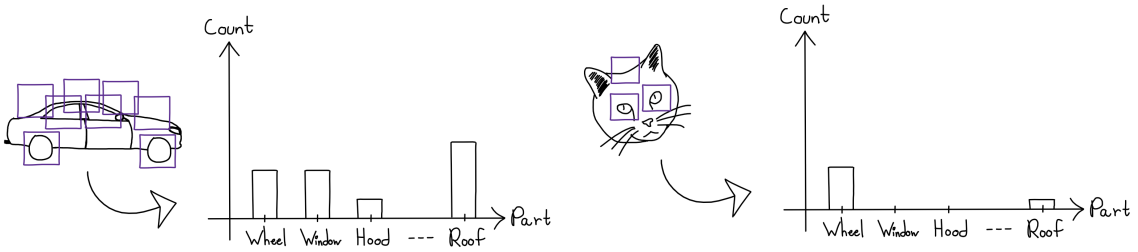


Figure 5.6: For many different objects there should be a distinct part count that can separate them. As an example one can imagine a car and a cat. For the car some part detectors might fire a certain amount of times on the wheels, hood, window and roof. For the cat these part detectors might also fire because certain visual cues appear similar to car parts, but the overall histogram should not be able to match that of a car.

5.3.2 Creating A BoW Model Based On Probability maps

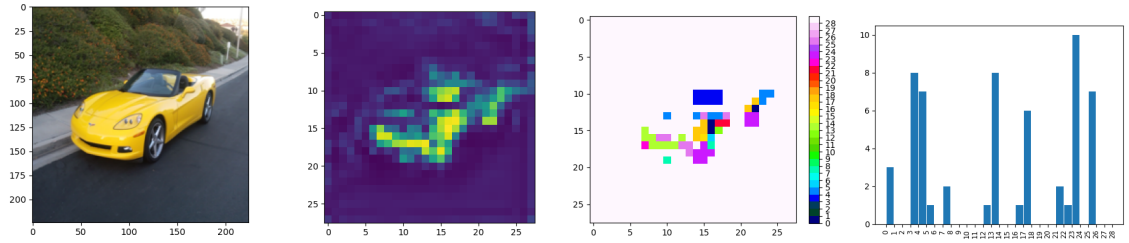
In order to count part hits, the soft probability maps $P_{K,W,H}$ need to be transformed into hard detections first. We do this with a heuristic that is similar to the one provided in section 5.2.1, where the difference is that we now take the channel-wise argmax shown in equation 5.2 instead of the previously shown channel-wise maximum from equation 5.1. This has the effect of reducing the probability maps $P_{K,W,H}$ into part detection maps $D_{W,H}$, meaning that instead of selecting the maximum probability value for each position, we now select the index of the filter that has the maximum value. This gives us a part detection output where we assume each filter represents a part, meaning each position can have one of K values.

By taking the channel-wise argmax we assume that the largest probability part is the only part present for that specific position. This is not necessarily correct since the receptive field at each position covers an image region, meaning multiple parts can have a presence at each position. We choose to ignore this because we still see that selecting the

filter with the largest probability still has inherent meaning, although some information might be lost. The added benefit of doing this is that the heuristic is very simple to understand and reduces the complexity of the data. Another inherent problem of taking the channel-wise argmax is that some of the related maximum probabilities of the part detection maps $D_{W,H}$ can be low and still be selected as a part. In order to avoid this we use a probability threshold t to ignore weak detections, meaning that part detections are only counted in the histograms if their corresponding maximum probability is larger than the set threshold t.

$$D_{W,H} = \underset{K}{\operatorname{argmax}}\, P_{K,W,H} \tag{5.2}$$

Our overall process is, therefore, to first reduce probability maps $P_{K,W,H}$ into part detection maps $D_{W,H}$. We then count how many times each of the K parts are present if their corresponding maximum probability was higher than a set threshold t. This turns each each part detection map $D_{W,H}$ into part histograms H_K. This process is shown in figure 5.7 for a car example.



(a) Original input image

(b) The corresponding segmentation map $S_{W,H}$

(c) The corresponding detection map $D_{W,H}$. The positions where probabilities of the segmentation map are under the threshold t is colored as white (the "not present" value 28)

(d) The histogram counted from the pixels of the detection map. The "not present" value 28 is not counted.

Figure 5.7: The process of retrieving a histogram

5.3.3 SVM-Based BoW As A Classifier

By extracting histograms from all the training and testing examples of the dataset defined in section 4.3 we then get a training and testing dataset of histograms corresponding to either cars or non-cars[3].

The last step of a Bag of Words approach is to learn a model to separate the histograms. Since the histograms are just vectors with 28 dimensions, any classification methodology that can handle vectors of at least 28 dimensions could be used. To keep our baseline simple, we choose to use SVM since this is a well-known model that is effortlessly set up from any modern machine learning library. We train the SVM model with the dataset

[3]We also applied a minor post-processing step where the extracted histograms were normalized between 0-1.

of training histograms and then classify the unseen testing histograms with the learned model. By comparing the classifications of the testing histograms with their ground truth labels, we calculate the evaluation metrics that can be seen in section 6.1.

5.4 Approach B - Bag Of Words Spatial Neighborhood Extension

5.4.1 Introduction

In section 5.3 we described how we implemented a BoVW baseline approach. Following the implementation of this baseline approach, it is natural to evaluate how simple spatial extensions of BoVW affects its performance. To that end, we decided to investigate the paper [ZM10]. The selection of an approach was primarily guided by a need for relative simplicity owing to time constraints. All spatial extensions in [ZM10] are intuitive and low complexity, making it a good choice for our purposes.

5.4.2 Spatial Relationship Modeling Via Part Neighborhoods

The paper [ZM10] proposes three different methods, all described in section 5.1.2. We will in this section, limit ourselves to only considering the first of the proposed extensions. Like in section 5.3, we will use a SVM to classify the part vector representation.

In this case, the part vector representation will not be a direct count of the parts but is instead the frequency of two part-classes A and B co-occurring in a neighborhood where an instance of A is the center part. It is possible to model this for two of the same part class, meaning $A = B$. The size of the sliding window defines the neighborhood size. For example, if the sliding window is 3×3 we have defined a neighborhood as being all pixels surrounding the center using a one-pixel distance. Likewise, we have a two-pixel distance if we use a 5×5 window. The process is shown in Figure 5.8. All pre-processing steps to decide which part is active in a location are the same as those outlined in subsection 5.3.2.

One concern associated with this approach is the selection of classes to consider valid neighbors. Selection is in [ZM10] handled by only considering $A = B$ as valid neighbors for all part classes. In other words, the generated histogram represents the frequency a part-class neighbors itself. This is not necessarily an unreasonable approach since we find that there are frequently multiple strong activations in areas where the part of interest resides. However, we can not guarantee that this is an optimal selection.

Another concern is the window size selection since it is not likely that the optimal offset is the same for all part class combinations. Ideally, a method for selecting the best combination of window size and part neighbors should be developed. Unfortunately, we were unable to implement such functionality in time. However, we did observe that for 3×3 windows, parts most frequently co-occur with itself. For both 5×5 and 7×7 windows, we found that most filters most frequently co-occur with part hits from filter 13.

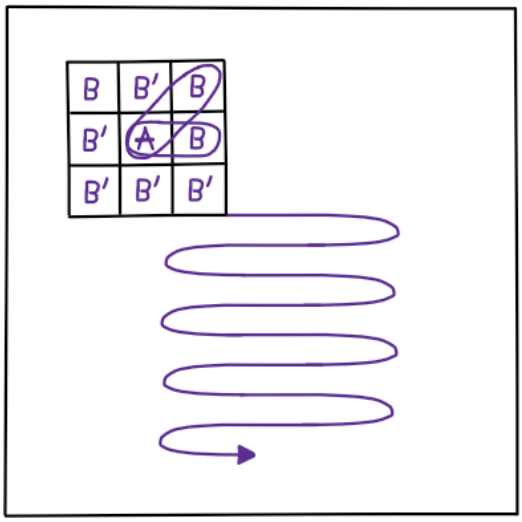


Figure 5.8: An illustration showing how the neighborhood is counted using the sliding window. B' represents all parts other than B, i.e., $B' = \neg B$.

The naive approach of using the neighborhood definition where $A = B$ yielded the best result when using a 3×3 window, and is also the one we evaluate in section 6.1.

5.5 Approach C - Bag Of Words Spatial Neighborhood Configuration Extension

5.5.1 Spatial Relationship Modeling Via Part Neighborhood Configuration

The second approach from [ZM10] that we implemented is based on using part neighborhood configurations. The part neighborhood configurations represent an encoding of how part hits of class B are spatially configured around a central part hit of class A in a sliding window. We used a 3×3 sliding window, which means that we have eight possible positions where a part hit of class B can present itself. In other words, we have 2^8 possible ways part hits of class B can be configured around a central part hit A.

The neighborhood configuration is first encoded using a bit string. The generation process from the sliding window to the bit string is shown in Figure 5.9. The bit string is then converted into a decimal representation. The decimal representation is used as a bin index in a histogram which counts the number of times a given neighborhood configuration occurs. While we can use an SVM to classify the histograms, we opted to try a probabilistic approach instead. The histogram is, therefore, used to represent the conditional probability of a part configuration occurring given the object class and the center part hit class. There are two histograms per selected filter, each consisting of 256 bins.

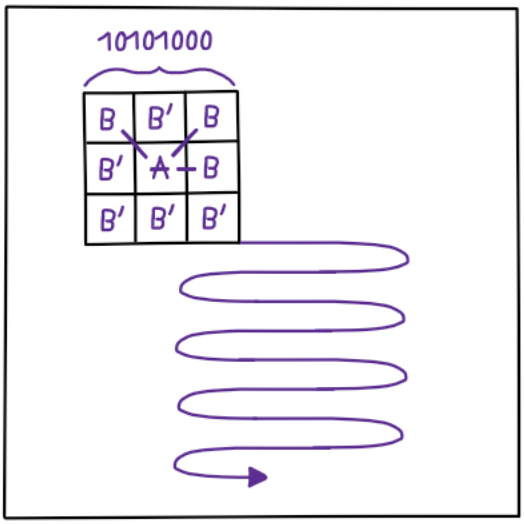


Figure 5.9: An illustration showing how the bit string representation is generated from the neighborhood configuration captured by the sliding window. B' represents all parts other than B, i.e., $B' = \neg B$.

5.5.2 A Spatially Extended BoVW As A Classifier

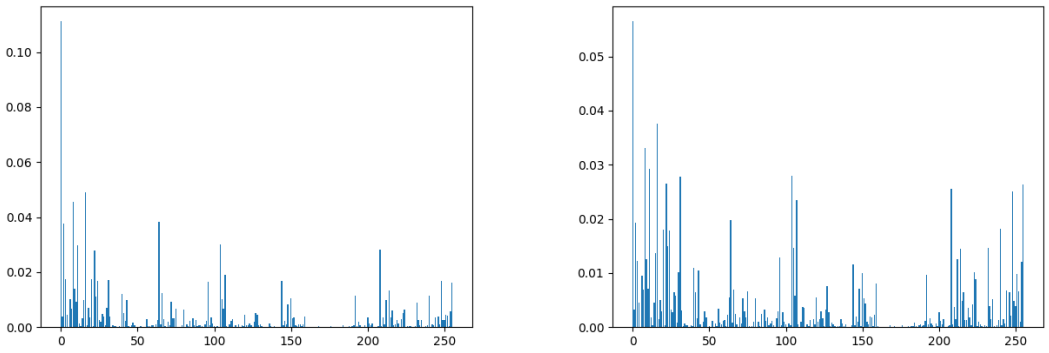
As mentioned, this approach takes a probabilistic view of the problem. Since the histograms are treated as estimates of conditional probabilities, we can use a variant of Naïve Bayes to create a generative model. We decided to use *Multinomial Naïve Bayes classifier*, defined in Equation 5.3. To avoid problems associated with underflow, we use a variant defined in terms of log, see Equation 5.4. The variable x_i represents the count of a neighborhood configuration i. The variable $p_{K,i}$ represents the likelihood of configuration i given the object class K.

$$\hat{y} \overset{\text{def}}{=} \underset{K}{\operatorname{argmax}}\, p(w|C_K) = \underset{K}{\operatorname{argmax}}\, \frac{(\sum_{i=1}^{N} x_i)!}{\prod_{i=1}^{N} x_i!} \prod_{i=1}^{N} p_{K,i}^{x_i} \tag{5.3}$$

$$\hat{y} \overset{\text{def}}{=} \underset{K}{\operatorname{argmax}} \sum_{i=1}^{N} x_i \log(p_{K,i}) \tag{5.4}$$

One issue that can arise from using the histogram is the existence of zero-valued bins. In the case of Equation 5.3 it zeroes out the probability, while in the case of Equation 5.4 it is undefined. To avoid this issue, we apply a smoothing function f_{smooth}. We decided to use the Laplace smoothing function, ensuring that all zero-valued bins get a small non-zero probability. The Laplace Smoothing function is defined in Equation 5.5. The variable α is the smoothing term, b is the number of bins in the histogram, and N is the total number of samples in all the bins in one histogram. We use $\alpha = 1$ to smooth the histograms. Examples of smoothed histograms are shown in Figure 5.10.

$$f_{smooth} \overset{\text{def}}{=} \frac{x_i + \alpha}{N + \alpha \cdot b} \tag{5.5}$$



(a) Filter 13, Non-Car.

(b) Filter 13, Car.

Figure 5.10: Examples of smoothed histograms containing conditional likelihoods for the 256 different neighborhood configurations.

Chapter 6

Analysis

6.1 Results

This section presents the results of the joint-structure classifiers introduced in chapter 5. All the methods use the same test dataset of probability maps $P_{K,W,H}$ which is processed into detection maps $D_{W,H}$ and segmentation maps $S_{W,H}$. In order to create detection maps, a threshold t needs to be set. We want to find out how the degree of strictness affects the results, and therefore chose to try out multiple thresholds, specifically all values of $t \in [0, 1)$ with a step size of 0.05. Since each approach uses the detection maps differently, some additional preprocessing is also done[1]. Approach A and B use SVM as their classifier, with the Radial Basis Function (RBF) set as the kernel. Approach C uses multinomial Naïve Bayes as its classifier.

Since our method is based upon using feature maps in VGG16, we also wanted to test how VGG16 would perform on our test dataset. Since the VGG16 architecture was pre-trained on ImageNet, there are 1000 different output nodes in the final layer of the network that represent a specific class. Many of these outputs represent different cars, and since we only wanted a single car class, we also performed some post-processing of the network outputs. If the Softmax output of the network predicted one of the following class ids $\{436, 468, 511, 609, 627, 654, 656, 675, 705, 717, 734, 751, 757, 817, 864\}$, we set the networks prediction as 'car'. If the Softmax output was any other class id, we set the prediction as 'non-car'. We tested a pre-trained version of VGG16 that we did not fine-tune towards our dataset and a version that we fine-tuned towards our dataset. The fine-tuning of the network was done over 5 epochs with a batch size of 8. We used SGD as our optimizer with a learning rate of 0.0005, momentum of 0.9, and weight decay of 0.0005.

All the metrics can be seen in Figure 6.1 as graphs over the different thresholds. The metrics we use are based on binary classification since we only have two labels in our dataset. We use the definition of a *Positive* to denote the car class and *Negative* to denote

[1]For example, the regular Bag of Words directly counts part hits while the Neighborhood extension counts part hits in a region, and so on...

the general non-car class in our data. From this, the inference can have one of four states. They can either be a *True Positive* (TP) which means the inference was positive and the class was positive, a *False Positive* (FP) which means the inference was positive and the class was negative, a *True Negative* (TN) which means the inference was negative and the class was negative and a *False Negative* (FN) which means the inference was negative and the class was positive.

We also calculate some additional metrics: the *Accuracy* as the fraction of correct inferences from the total samples, defined in Equation 6.1.

$$\text{Accuracy} \stackrel{\text{def}}{=} \frac{\text{Correct}}{\text{Total}} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{6.1}$$

The *Recall* as the fraction of how many positive samples are correctly inferred, which is defined in Equation 6.2.

$$\text{Recall} \stackrel{\text{def}}{=} \frac{\text{TP}}{\text{P}} \tag{6.2}$$

The *Precision* as the fraction of how many positive inferences are true, which is defined in Equation 6.3.

$$\text{Precision} \stackrel{\text{def}}{=} \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6.3}$$

Lastly, we also calculate the *F1-score*, which is the harmonic mean between precision and recall defined in Equation 6.4. Since this metric combines precision and recall it measures the ability to both retrieve and correctly classify positive samples.

$$F1 \stackrel{\text{def}}{=} \left(\frac{\text{P}^{-1} + \text{R}^{-1}}{2} \right)^{-1} = \frac{2}{\frac{1}{\text{P}} + \frac{1}{\text{R}}} = \frac{2 \cdot \text{P} \cdot \text{R}}{\text{P} + \text{R}} \tag{6.4}$$

6.2 Discussion Of Results And System Properties

The results achieved by our joint-structure classifiers are discussed in subsection 6.2.1. In subsection 6.2.2 we discuss different behaviors related to filters in DCNNs. In subsection 6.2.3 we discuss ways to improve the approach we used for generating part detection from section 4.7. Lastly we also discuss architectural and training related topics in subsection 6.2.4 and subsection 6.2.5. Because detection is very important to achieve we also dedicate chapter 7 to consider how our classifiers can be extending for performing detection.

6.2.1 The Joint Structure Classifier Results

We tested three different joint-structure classifiers, ”A - Bag of Words”, ”B - Spatial Neighborhood”, and ”C - Neighborhood Configuration”. We cannot distinguish between ap-

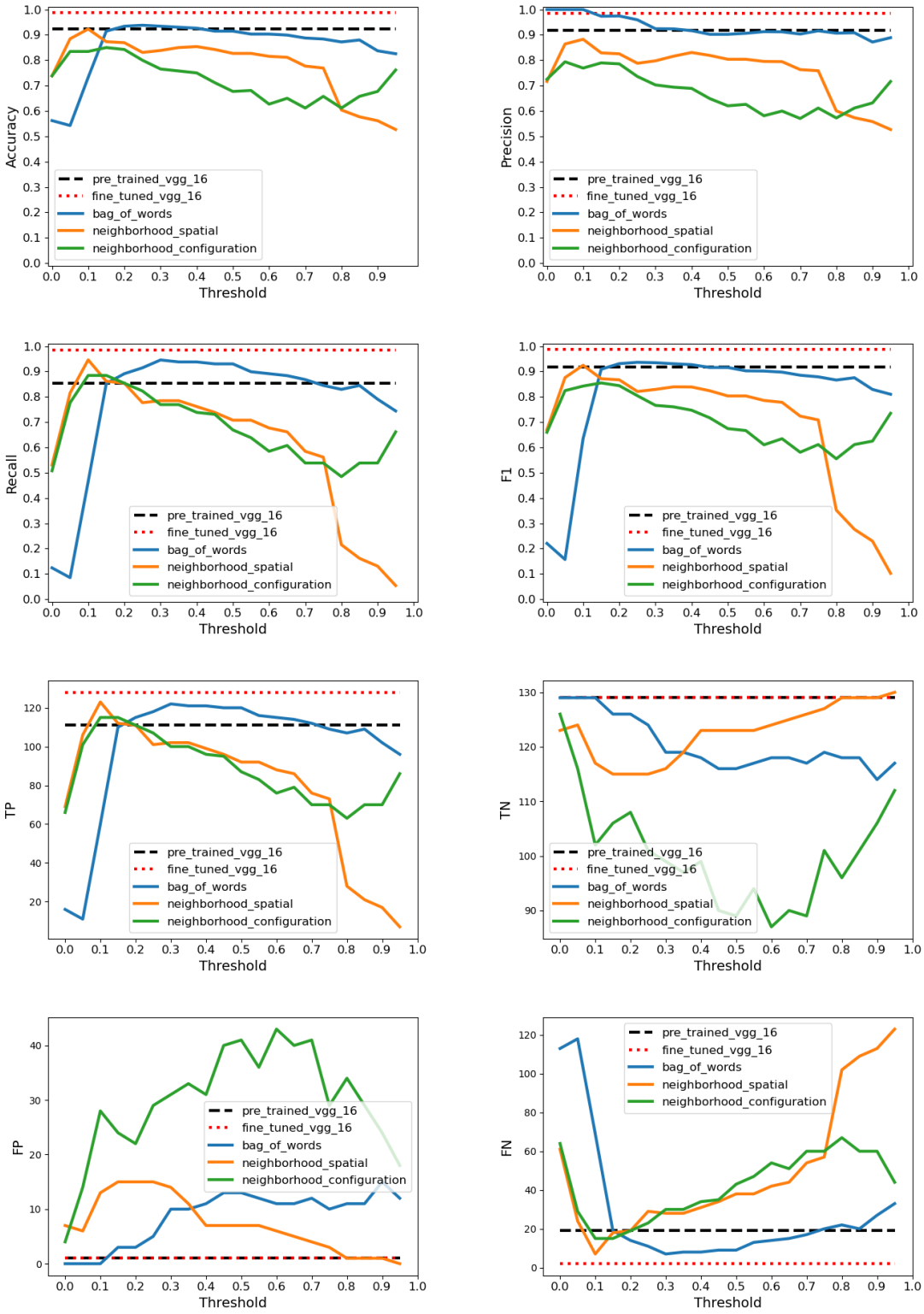


Figure 6.1: All the metrics we measured in graph form. The y-axis shows the metric value and the x-axis shows the threshold that was used.

proach A and approach B directly from the found accuracies because they seem to achieve almost the same results at different thresholds. Approach C seems to be outperformed by both A and B. We see that the VGG16 network outperforms every approach if it is fine-tuned on our dataset. However, if it has not been fine-tuned, it does not outperform approaches A and B at the optimal thresholds.

The interesting property of the spatially extended approaches is that they seem more capable of handling a noisy probability map. As we can see from Figure 6.1, both spatially extended methods outperform the BoW baseline on Accuracy, Recall, and F1 when the probability threshold is low. However, the spatially extended approaches are consistently less precise than the BoW baseline. We expected that incorporating spatial information would yield improved precision compared to the baseline. The reasoning is that the system would have a greater capacity to recognize unlikely spatial configurations of part hits, thereby reducing the number of false positives. The cause for this not being the case may be that the spatial extension we used is not expressive enough to represent such cases adequately. We believe that more precise modeling of the spatial part-to-part relationships should improve the precision relative to the BoW baseline.

We observe that approaches A and B outperforms the untuned VGG16 for a small range of t values. However, if we fine-tune the VGG16 classifier network, it outperforms all the classification approaches we created. Still, one should consider that the joint structure system discards layers after Conv4-3 of VGG16 and replaces the entire classification network with either SVMs or a Naïve Bayes classifier. With this in mind, it is reasonable to say that the joint structure system performs well for the following reasons. (1) It has effectively pruned 94% of the network parameters[2] without taking a hit to accuracy when compared to an untuned VGG16. (2) The approaches are built upon probabilities and not a hidden state.

Failure Situations

We also investigated each incorrect prediction from approach A to better understand what lies behind the failure situations. We do not investigate approaches B and C because the generated probabilities would be similar; the only difference would be how the approaches handle them. The failure situations for approach A can be seen in Appendix D.

In general, we observe that failures are rooted in two scenarios: (1) The part hits we find in our detection maps seem a bit noisy for some examples, which might create confusing histograms for the SVM classifier, ultimately leading to false negatives. (2) The non-spatial nature of Bag of Words seems to cause false positives when it sees many sharp corners, e.g., on laptops or picture frames.

[2]We calculated this in PyTorch. The VGG16 model before cutting it at Conv4-3 was ≈ 134.27 million parameters, and ≈ 7.64 million parameters afterwards

6.2.2 Analysis And Evaluation Of Filters

Unanalyzed Behaviors

Based on a qualitative assessment of the results we produced, we believe that the part filters are capable of relatively accurate parts localization. However, there are certain behaviors we did not sufficiently analyze to draw any conclusions. We, therefore, state these behaviors in the following paragraphs and suggest that they should be investigated further.

The first unanalyzed behavior is how the receptive field affects what parts a filter activates for. There is already the possibility to easily calculate the receptive field for each layer in a network using existing work [ANS19]. The paper [Luo+16] is also interesting to investigate as it hints at the receptive field being more complex than initially thought. One idea could be to use the methods from the papers to calculate the receptive fields of a layer and then measure activation strengths for filters in different settings. One example of this is the "wheel" specific filters we found. It would be interesting to investigate how different wheeled objects like mopeds and buses activate this filter. We assume that a larger receptive field could be more susceptible to the context around the wheel, but we currently have no way to prove this. Another interesting experiment could be to see how different wheel scales activate the filter.

The second unanalyzed behavior is the entanglement of filters. What we mean by this is that we do not know how many classes or parts a filter can activate for. For example, we observe that specific filters can activate on multiple parts, but we do not have any empirical measures of this. One example of this is filter 429, which likes to activate for the wheel region of cars. In specific situations, we also see that this filter can activate on the hood and roof of cars. An example of this can be seen in Figure 6.2.

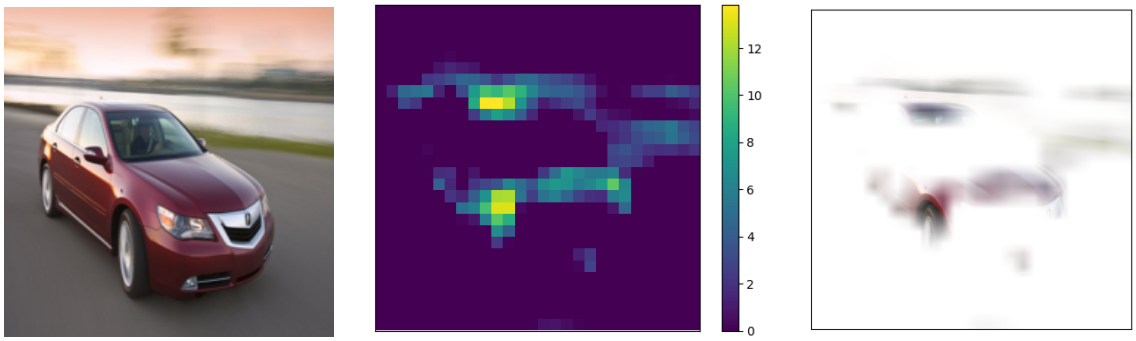


Figure 6.2: An example of how a filter can activate for multiple parts. The first column is the original image. The second column is the ReLU activated feature map at Conv4-3 in VGG16 for filter 429. The third column is the original image with the interpolated and normalized feature map set as the alpha channel. Note how the hood and roof are also activated over.

Large Negative Responses

Currently, we only measure filters as indicative of being a part detector by how positive their feature map maximum values are. We also noticed that some filters have large

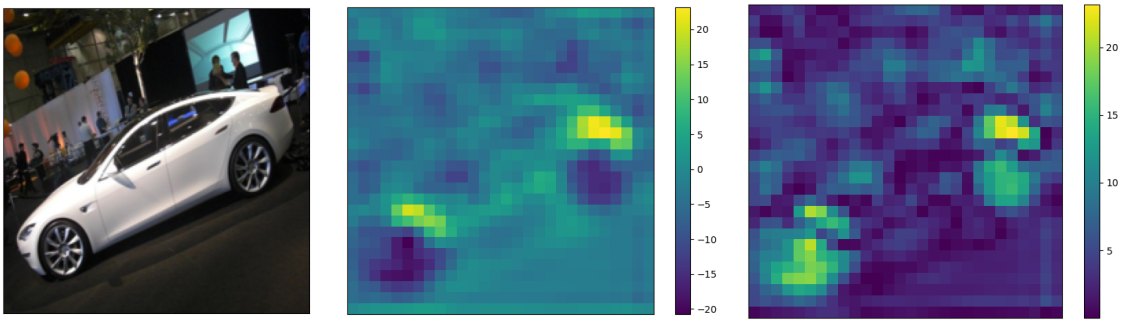


Figure 6.3: An example of a input image, the corresponding pre-activation feature map produced from filter 429 in the Conv4-3 layer of VGG16, and the absolute valued feature map. Note how the feature map has a large positive and negative response, which seem to map to wheels.

negative values to certain visual cues. An example of this can be seen in figure 6.3. We do not know what causes this effect is. It could, for example, just be a symmetric effect of a kernel that produces a large response. However, this is just an assumption, so we would recommend that further work also investigates what large negative responses indicate. It could be that minimum values of a feature map are also measurable for finding part detectors. If it turns out to be the case, a natural conclusion could be that the sign of feature map values does not matter for our approach but rather their magnitude. A possibility could be to measure the absolute pre-activation feature map values $|F_{C,W,H}|$ instead.

Direction Of Estimated Conditional Probabilities

Our approach was initially built on finding filters that could separate the conditional probabilities, but this did not account for which conditional probability the filter was indicative. In the last paragraph of subsection 4.7.3 we therefore defined the direction of the Bhattacharyya distances using the μ values. In our analysis, we ended up only using filters that had what we designated as positive directions, meaning $\mu_A > \mu_B$. This was done assuming that filters with large positive Bhattacharyya distances also behaved as part detectors for class A since it had learned something from the class that excited it more than the classes labeled with B.

However, the filters with a large negative Bhattacharyya distance could also have some indicative value for class A. Since the label B contains many different classes defined as not A, a filter that activates more for B has probably learned something common that appears for all those classes. This means that the absence of activations in these filters could indicate that class A is present. This depends on exactly what the filters are localizing, however, since the absence of something does not necessarily indicate the presence of something else. For example, if it turns out that these filters activate over a context like ”blue sky” or ”green trees”, it would be hard to justify that these could indicate the presence of a car directly.

Therefore, we suggest that further work should look closer into the filters with large negative Bhattacharyya distances and analyze what they localize and how they can be used.

We show an example of the difference between a filter with a large positive Bhattacharyya distance and a large negative Bhattacharyya distance in figure 6.4.

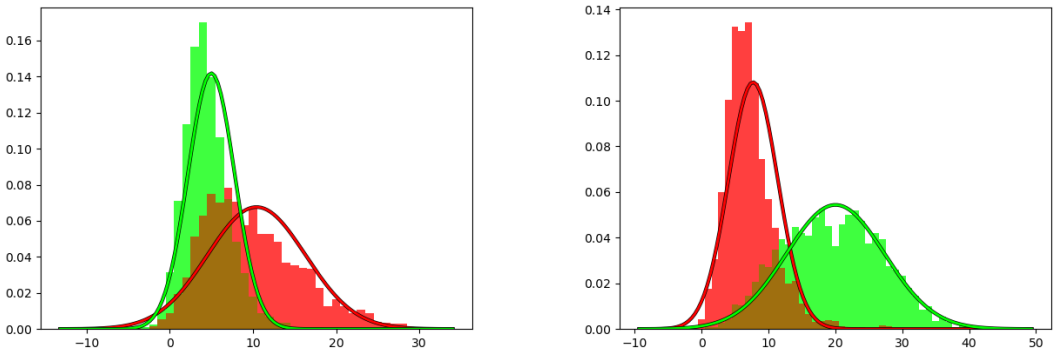


Figure 6.4: The difference between a negative distance on the left and a positive distance on the right.

Negative Signal

In all our experiments, we have consistently used ReLU as the layer activation function before the final layer, which uses logistic regression. As a quick reminder, ReLU is defined as $max(0, I)$, which means that negative input signals are discarded. We noticed that some filters produce a strong negative response for car parts, as shown in Figure 6.3. Due to the use of ReLU in our experiments, this information is lost. Therefore, using a loss function that conserves negative signals might be beneficial. One possible approach is to use a loss function like leaky ReLU. This function conserves the sign of the input signal but typically assigns relatively higher importance to the positive signal. Another possibility is to assign equal importance to both the negative and positive signals and switch the sign of the negative signal. [BCT16] proposes such an approach by duplicating a convolutional layer. The output from both the duplicate and original layers is normalized and zero-centered. Additionally, the output from the duplicate layer is multiplied by $x - 1$, which switches the signs compared to the output from the original layer. The outputs of both layers are concatenated and passed to a ReLU activation function, resulting in a doubling of feature maps per layer.

Further Reduction Of The Filter Selection

The filter selection process described in section 4.7 is designed so that the most class-discriminative filters are selected. However, the method does not consider the degree of mutual information between the selected filters. High degrees of mutual information manifests as co-occurring filter activations, i.e., multiple filters activate on the same visual concepts. Furthermore, the method does not directly consider the noise level produced by each filter. In other words, the method does not consider the number of false part hits a filter generates. As described in section 4.6, [ZWZ18; Zha+20] explicitly incorporates the mutual information in the loss function shown in Equation 4.11. While we believe the loss function is a reasonable way to reduce mutual information, it would mandate

additional training for the DCNN we use. The method described in section 4.7 does not necessarily require further training if it is pre-trained. This training flexibility is a feature that we believe is valuable to preserve.

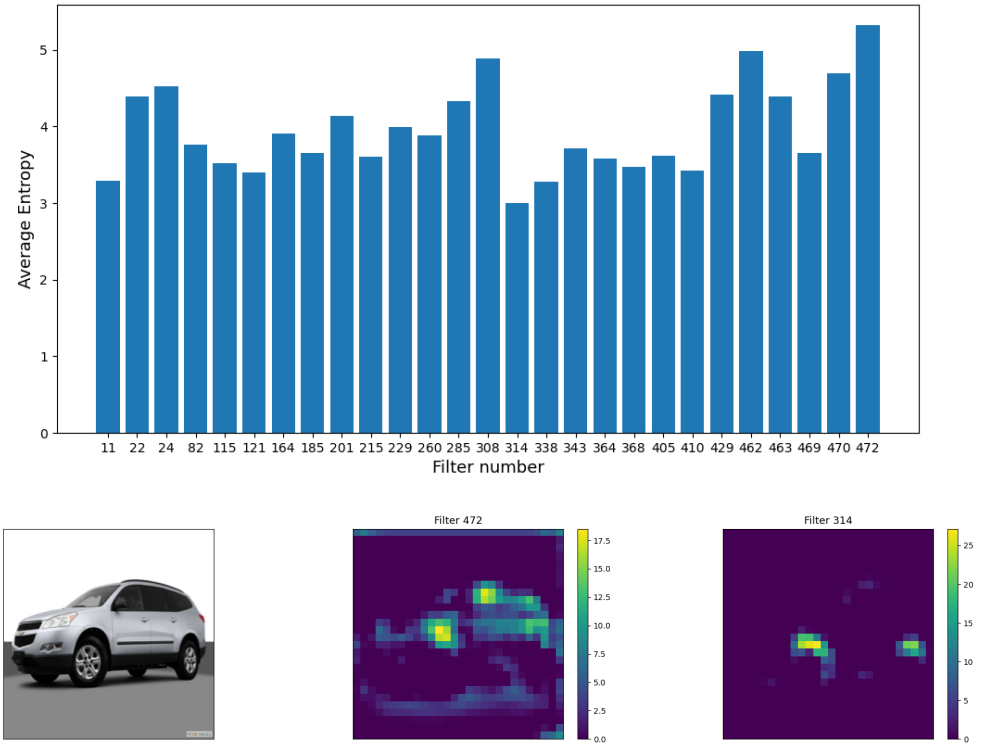


Figure 6.5: The measured average spatial entropy of the selected filters. As shown in the bar plot, filter 472 has the highest entropy while 314 has the lowest, this also seem to correspond to what we observe upon inspection. The second row show the original image in the first column, the output of filter 472 in the second column and the output of filter 314 in the third column. Note how filter 314 with lower average spatial entropy has more concentrated output.

One way to potentially reduce the number of false part hits is to select the filters which produce the most spatially concentrated outputs. High spatial concentration may be beneficial in reducing the level of noise introduced by the part detectors. One approach to measuring the spatial concentration of part detectors is to use spatial entropy. Low spatial entropy corresponds to high spatial concentration. In Figure 6.5 we show the measured average spatial entropy of filters. The measurements were made by processing all the training images of cars to generate feature maps with the selected filters[3] of Conv4-3 in VGG16. The feature maps were flattened, meaning they were transformed to vectors $w \in \mathbb{R}^{W \cdot H}$, and then used to measure the average spatial entropy.

The spatial entropy approach may yield lower noise levels but is insufficient to find a selection of filters with reduced mutual information. A possible way to measure the similarity of the outputs between filters is to use cosine similarity. We measure average cosine similarity by generating the same $w \in \mathbb{R}^{W \cdot H}$ vectors described above. An example of a cosine similarity matrix is shown in Figure 6.6. Cosine similarity gives a correlation

[3]This refers to the selection process based on the top Bhattacharyya distances from section 4.7

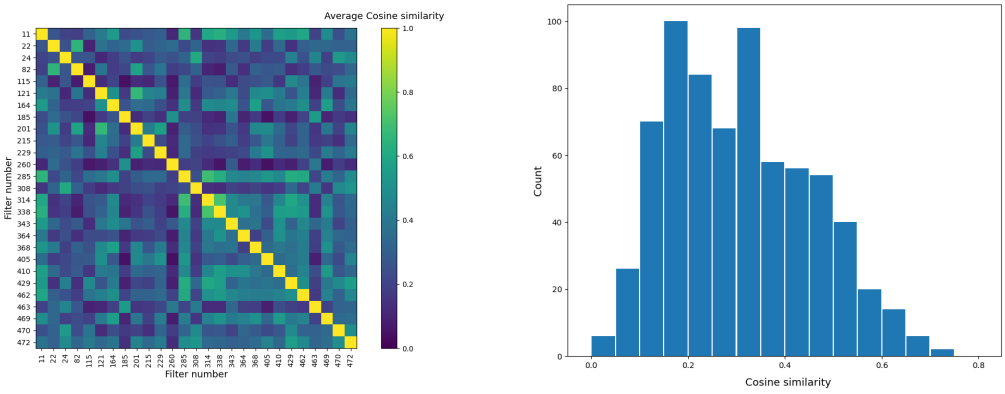


Figure 6.6: The similarity matrix shows the measured average cosine similarity between the selected filters. The histograms shows the count of the similarity values found in the similarity matrix. The diagonal is ignored since it shows the vector similarity against itself.

measure between the filter outputs, i.e., cosine similarity represents a different approach to measuring filter output similarity than mutual information. Since cosine similarity is a correlation measure, it can be used to reduce the filter selection on filter output similarity. However, it does not impose any constraints in terms of spatial concentration. Zhelezniak et al. [ZSH20] note that correlation-based approaches are typically used as word embedding similarity measures and propose a method for estimating mutual information, resulting in a good candidate similarity measure.

Another possibility is to avoid reducing the selection of filters altogether and instead perform dimensionality reduction on the outputs of the filters. Principal component analysis (PCA) is a method that can be used to reduce the number of components given some explainability target. This reduction should be understood as finding closely related variables and creating a joint representation in the form of principal components. In terms of the filter outputs, one can view this as identifying and merging closely related information found in multiple filter outputs. The explainability target is represented by the proportion of activation variance explained by the residual principal components. For example, one can set the target to be that the components are reduced to a set of principal components, representing a certain percentage of the original activation variance.

As evident from the approaches outlined above, several ways exist to reduce the filter selection or the dimensionality of their outputs. The reader should also note that the presented approaches in no way are exhaustive in terms of the possibilities that exist. We can not confidently state that pursuing such approaches will yield meaningful improvements in the parts detectors. However, we present them here since we think they could offer some benefits upon closer inspection.

Context Sensitive Filters

A phenomenon that has been observed in machine learning is the ability of systems to cheat, meaning they often use simple shortcuts when the opportunity to do so presents itself. Some examples of this can be seen in [Lap+19] which analyzes some invalid

problem-solving behaviors. For DCNNs, this problem is typically the misuse of context in images to generate a prediction. This can, for example, be non-valid correlations in training data such as watermarks or image artifacts which has no relation to the actual visual task. A real example is the horse classifier presented in [Lap+19]. This was a classifier that cheated on horse images by using the presence of a watermark to classify horses. The authors also demonstrated how this could be exploited by changing the classifications of a car to a horse when a watermark was added.

We also see that context problem appears in our approach to finding part detector filters. Since we do not control where filters activate, there is nothing that stops context-sensitive filters from being selected. Although we do not have empirical proof, it seems that road surface markings are one example of context-sensitivity that some of the car filters we found in section 4.7 likes to activate for. Figure 6.7 shows an example of a road surface marker sensitive filter.

One possible solution to avoid context-sensitive filters could be to use localization information to only select filters that localize where the wanted object class is. The assumption is that context is not an entangled property of filters and that simply separating filters can remove context issues. In the examples, we find where filters activate over context. This does not seem to be the case, however. To us, it seems more likely that context is ingrained in the filters, meaning a proper selection of filters will not fix the problem. For example, note how figure 6.7 shows a filter that activates over both surface markings and car parts. The explanation for why context entangles into filters is most likely related to what visual task a pre-trained network has trained for. In our example, we use a VGG16 pre-trained for classification on ImageNet, and this does not guide the network to avoid using context, which explains why some filters would also learn to activate for road markings. Therefore, a more reasonable solution is to use networks pre-trained for a visual task requiring localization instead, since this will have punished context use during training.

6.2.3 Improved Part Representations

Multi-Layered Approach

In the filter selection approach from section 4.7 we only attempted to use the Conv4-3 layer of VGG16 to find part detectors. We chose this specific layer due to the network dissection paper [Bau+20] presenting it as a layer rich with part representations. Therefore, a natural extension to our approach is to use the other layers when finding filters. Currently, we see that the part detectors we find are very focused on specific regions like window edges and wheels. We assume that these parts are found because they are the ideal size for the receptive field of Conv4-3. The potential benefit of using multiple layers is that we can find parts of other scales since the receptive field is either smaller in earlier layers or larger in later layers. Multiple layers will also allow the creation of a deeper tree of objects and parts than the two-layer deep tree outlined in section 5.1. [Zha+18] is an example of a paper that proposes something similar to what was described above.

Figure 6.7: Example of filter 429 from Conv4-3 of VGG16 that activates for car parts and road surface markings. The road line activations are usually weaker but still appear. Each row is one example, where the first column is the original image, the second column the ReLU activated feature map, and the third is the original image with an alpha mask that is one where the interpolated feature map is active and 0 everywhere else.

Inspiration could, therefore, be taken from their work.

We did a small experiment to test what parts might be found in earlier layers, so we also re-ran our method from section 4.7 for the Conv2-2 layer of VGG16. We primarily observe fewer high Bhattacharyya distance filters, i.e., we only ended up with K=5, whereas Conv4-3 had K=27. The filters we find for Conv2-2 also seem to have weaker expressive power, primarily focusing on edge information. We show some examples of the results from Conv2-2 in Figure 6.8. Our figure shows the original image, the corresponding segmentation map $S_{W,H}$[4], and the corresponding detection map $D_{W,H}$[5] made with $t = 0$ (i.e. we do not threshold the examples).

There seem to be expressive powers in filters, although their activations are weak; for example, note how the 'red' filter in the detection map is the argmax for many positions where probabilities are weak. It would also be interesting to see how much the car-specific filters of earlier layers affect the later car-specific filters. For example, one experiment could be to determine if the car-specific filters in Conv4-3 are affected by blocking the car-specific filters in Conv2-2.

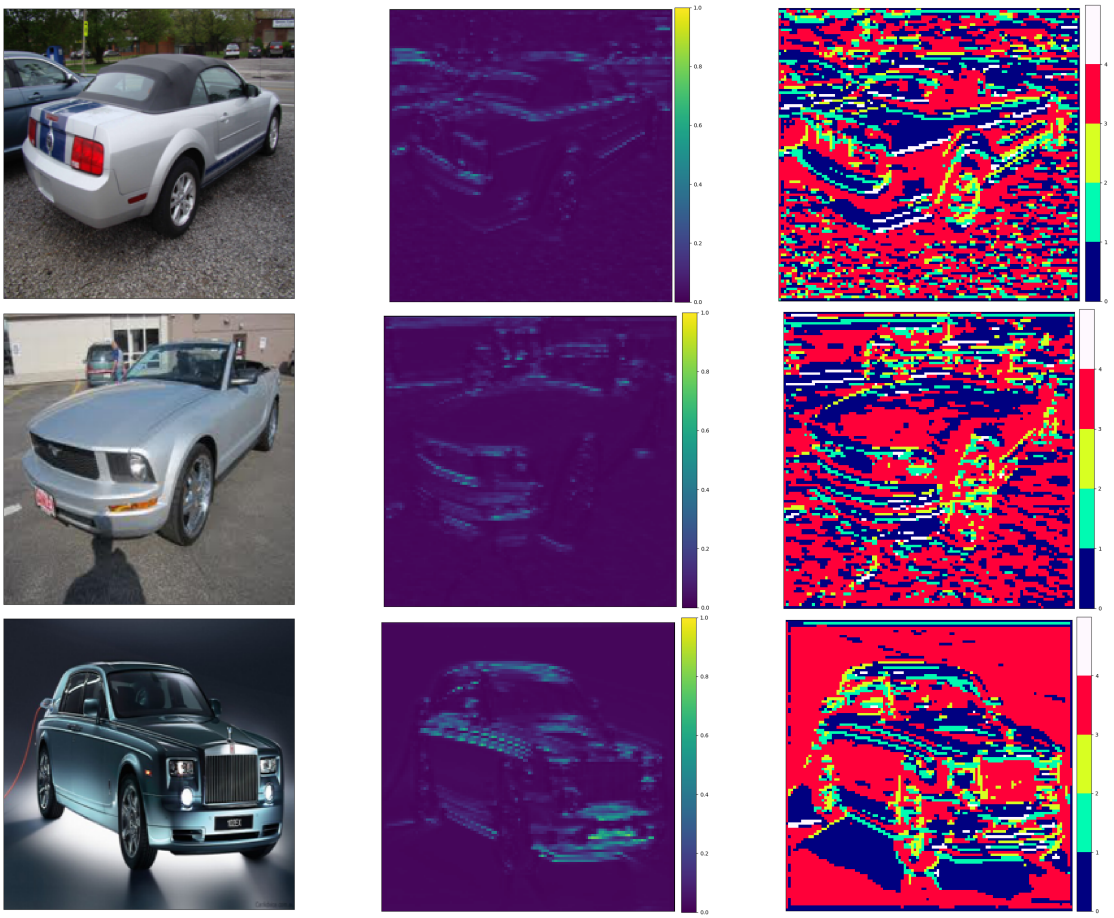


Figure 6.8: Each row shows one example. The first column is the original image, the second column is the segmentation map produced with the top 5 Bhattacharyya distance filters of Conv2-2, and the third column is the detection map that was also produced with the top 5 Bhattacharyya distance filters of Conv2-2.

[4]See section 5.2 for a reminder of how the segmentation map is made
[5]See section 5.3 for a reminder of how the detection map is made

Crossclass Part Filters

The filters we have chosen as part detectors in section 4.7 are heavily class specialized. This specialization occurs because we select a percentile of the filters with the highest Bhattacharyya distance. However, it is conceivable that shared parts, meaning filters with a low distribution distance, are beneficial in a joint structure system. One issue we encountered due to the exclusive use of class-specialized filters is that texturally homogeneous regions, such as car doors, are not detected. Exclusivity can, therefore, be detrimental to localization since it is likely that only a subregion of the whole object is detected. We show an example of the missing car door activations in Figure 6.9.

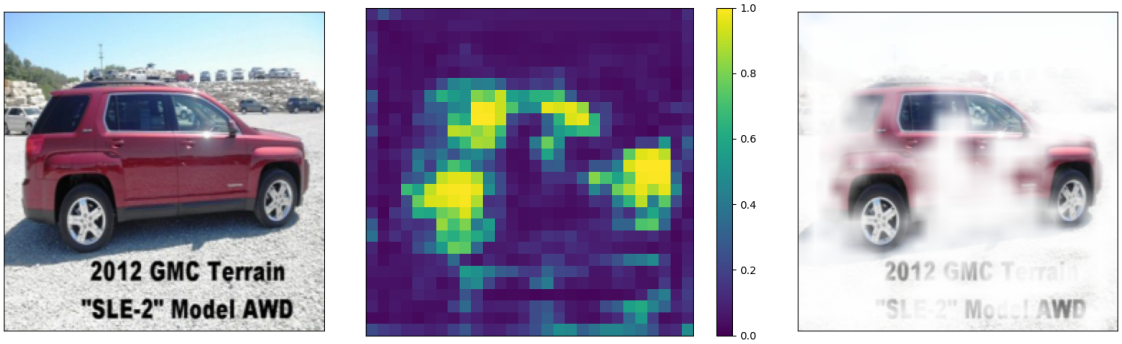


Figure 6.9: Example of the texturally homogeneous door region of a car not being found. The first column is the original image, the second column is the segmentation map $S_{W,H}$, and the last column is the interpolated and normalized segmentation map set as the alpha channel of the original image.

The selection of shared parts is non-trivial since it is not immediately evident which parts are shared if we only have image-level category labels. However, it is relatively more straightforward if more precise labeling, such as segmentation masks, is used. It is then possible to select those filters with a low distance that frequently activate on the pixels belonging to a relevant object instance. Some care needs to be taken so that filters that are too non-specific are not selected to avoid introducing too many background pixels. Another possibility is select filters that frequently cluster spatially close to the class-specialized filters. The idea is to generate a set of spatial bounds using the class-specialized filters. The spatial bounds can then be used to find and select filters that detect crossclass parts.

Dataset Sensitivity

Although it was not presented earlier, we also attempted to construct a boat dataset for use with the approach from section 4.7. This dataset was constructed the same way as the car dataset, where we ended up with ≈ 1930 close-up images of different boats as class A and ≈ 1930 random COCO images as class B.

Using the same approach for the boats as we did for the cars, we found some boat part detectors. The most prominent part detectors seemed to be for windows and bridges (control room) as seen in Figure 6.11. The reasoning for why we chose not to pursue using boat parts further was because we ran into an issue where we also found water detector

filters in our selection. Examples from a water filter can be seen in Figure 6.10. The problem with water detectors is that they are only contextually related to boats. Using them in a joint-structure system based upon parts would be wrong, so we would first need a way to prune the water filters.

The issues of finding water filters lead us to a problem that we did not initially consider: How the composition of the datasets affects our measurements. We assume that water filters were selected as part detectors because our non-boat images did not contain enough water, while the boat images contained both water and boats. This combination meant that boat parts were not the only unique visual cue but also water.

One possible solution to avoid dataset sensitivity problems is to start measuring with localization labels. This closely matches what the network dissection paper [Bau+17] does and would ensure we only find filters that activate on the object of interest. Additionally, more sub-category labeling in datasets could help find more part detector filters. For example, instead of collecting the concept of a car into a label A and then measuring over all the images, it would be better to measure each car sub-category itself. This could hopefully find a more diverse selection of part detectors since the measurements would not be as biased towards the dominant and common car parts like wheels. This idea could also be extended to other attributes like color and viewpoints.

6.2.4 Architectural Concerns

Using Other Models

In all the part detector approaches we tested, the only DCNN we attempted to use was a pre-trained VGG16. To have a more solid foundation for our approach in section 4.7 we think multiple different CNN architectures should be tested, for instance, ResNet. It is also possible that other types of architectures than CNN-based ones can also yield part detectors. We would therefore also recommend that this be explored. Examples of architectures that might be worthwhile to look into are AutoEncoders and GANs.

6.2.5 Network Training

Pre-Training And Regularization

When we analyzed the part detector approach from section 4.7 we chose to use a DCNN that was pre-trained for classification on ImageNet. This form of pre-training is one of the simplest that can be done and has some inherent weaknesses related to context abuse. Although we did not attempt other ways to train the network, we see that other forms of training and regularization will likely help create better filters. In section 4.2 for example, we present ways that part recognition can be improved in DCNNs. The simplest approach is to train on a part dataset, while the more complex approaches [Zha+21; HP19; Yan+20; Ji+21] perform specialized training or regularization. Other possible improvements can be found in making DCNNs less biased towards textures and low-frequency information [Gei+18; SG21; Li+22]. The assumption is that overuse of simple information

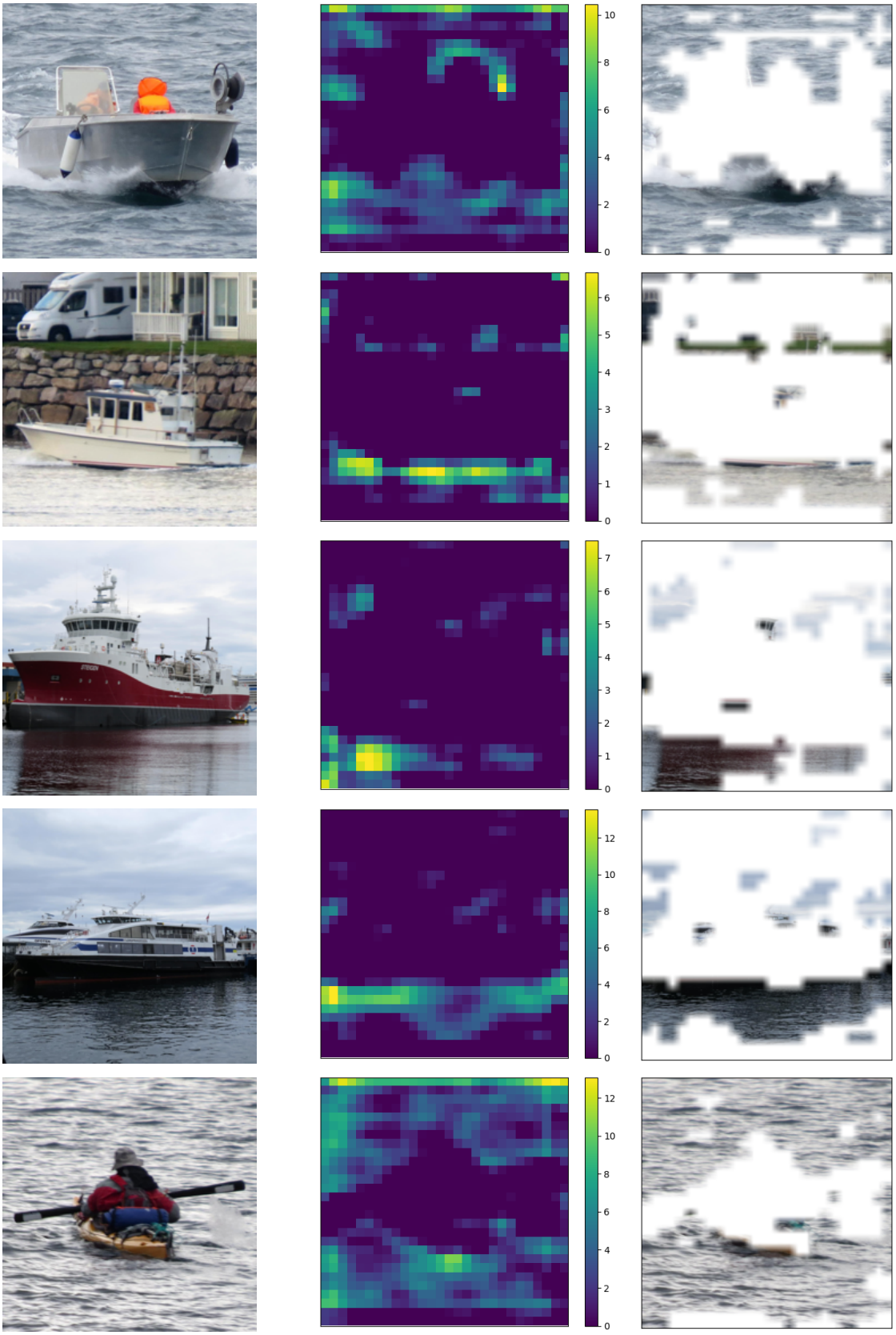


Figure 6.10: Example of filter 424 from Conv4-3 of VGG16 that predominantly activate on water. Each row is one example, where the first column is the original image, the second column the ReLU activated feature map, and the third is the original image with an alpha mask that is one where the interpolated feature map is active and 0 everywhere else.

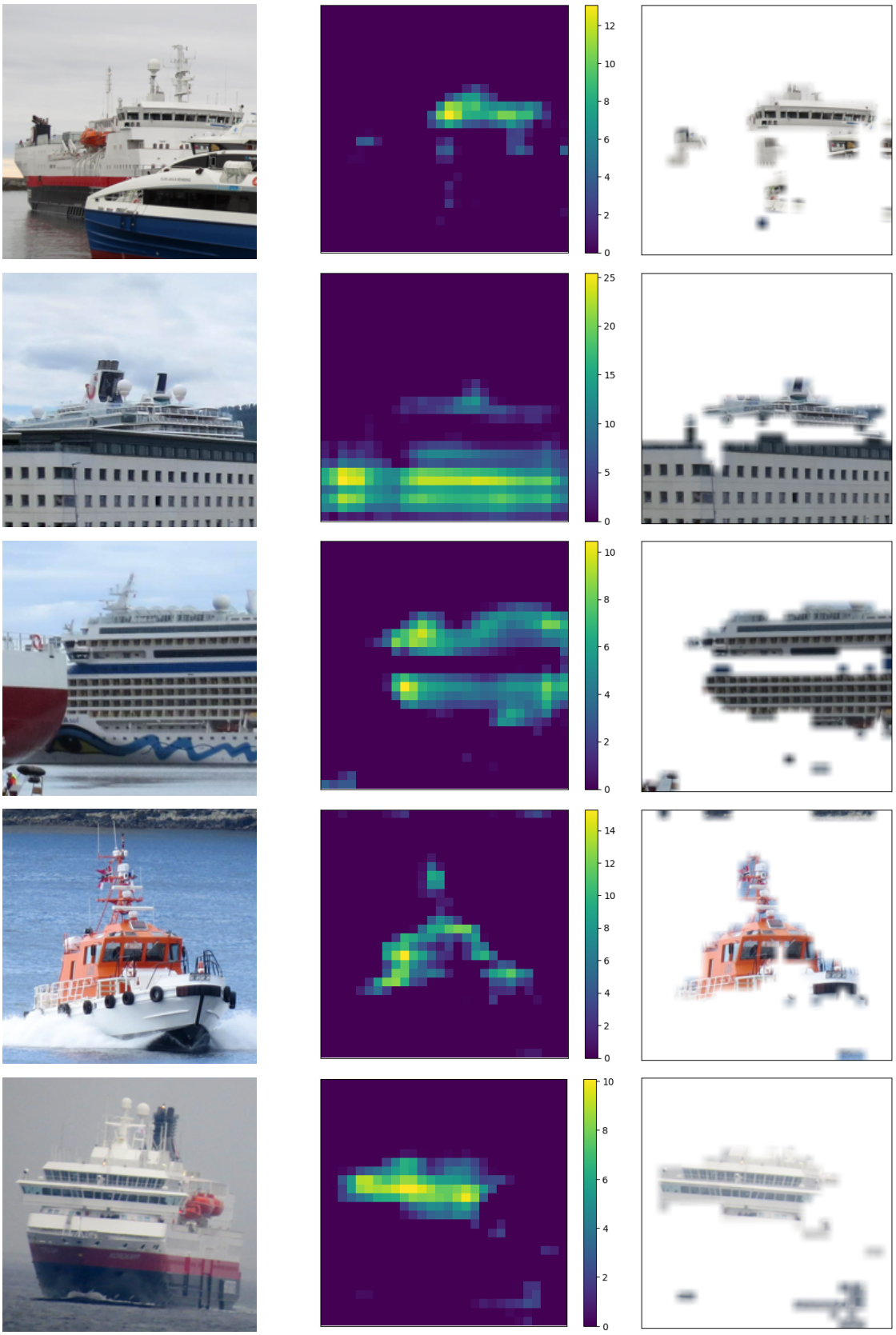


Figure 6.11: Example of filter 258 from Conv4-3 of VGG16 that behaves as a part detector for bridges on boats or windows on cruise ships. Each row is one example, where the first column is the original image, the second column is the ReLU activated feature map, and the third is the original image with an alpha mask that is one where the interpolated feature map is active and 0 everywhere else.

lets a DCNN avoid having to learn parts; it could therefore be that reducing such a bias also gives us a richer selection of part filters.

Loss Function Based On The Bhattacharyya Coefficient

The part detector approach proposed in section 4.7 is based around utilizing the distance between two distributions representing the maximum activations. However, as shown in Figure 4.28, the distance for the majority of filters is quite low. It could be interesting to develop a loss function that encourages the network to maintain a stronger separation between the maximum distributions. This idea can more intuitively be viewed as creating a loss function that pushes the network to have more class-specialized filters.

We believe the Bhattacharyya coefficient can be used as part of the loss function mentioned above. The Bhattacharyya coefficient is related to the Bhattacharyya distance, but instead of producing a measure of distribution similarity, it produces a measure of distribution overlap. The overlap between the distributions is given as normalized output, which is to say $BC \in [0, 1]$. See Equation 6.5 for a mathematical definition of the Bhattacharyya coefficient.

$$BC = \int \sqrt{p(x)q(x)}\, dx \tag{6.5}$$

In our view, the Bhattacharyya coefficient should primarily be used to augment one of the archetypal loss functions, such as cross-entropy. With this in mind, we believe the loss formulation shown in Equation 6.6 is suitable for inducing increased filter specialization. The coefficient γ is meant to serve as a hyperparameter that allows tuning the contribution to the loss caused by converging distributions.

$$\mathcal{L} = \mathcal{L}_{CE} + \gamma \mathcal{L}_{BC} \tag{6.6}$$

The main idea is to track the temporal changes in the activation maximums during training and penalize converging distributions. However, it is evident from Equation 6.5 that we need a continuous distribution while the activation maximums distributions are discrete. It is, of course, possible to redefine Equation 6.5 to handle discrete distributions using a sum instead of an integral, but it introduces problems concerning the gradient.

Our suggestion is to fit a normal distribution to the maximum activation histogram. The normal distribution can then be used for all calculations related to the loss. The histograms can be updated continuously by tracking the bin position of all images for all filters. To clarify, the bin position of an image is simply the bin position of the maximum activation for a specific filter. Each time an image changes bin position, we decrement the previous bin and increment the new bin. By using this method, we have established how one can track changes. Therefore, it is now possible to derive the gradient of the loss $\mathcal{L}_{BC}$ with respect to a filter weight w. The variables σ and μ are the variance and

mean, respectively, with the subscript denoting to which distribution it belongs. The variable x represents the maximum activation from an input image. We will not show the derivation of the gradient here but instead refer the reader to Appendix E.

$$\frac{\partial \mathcal{L}_{BC}}{\partial w} \stackrel{\text{def}}{=} \frac{\theta e^{\psi}}{4\pi\sigma_p\sigma_q\sqrt{\phi}} \tag{6.7}$$

$$\phi \stackrel{\text{def}}{=} \frac{1}{2\pi\sigma_p\sigma_q}\exp(-\frac{1}{2}\frac{((\sum\sum wx)-\mu_p)^2}{\sigma_p^2} - \frac{1}{2}\frac{((\sum\sum wx)-\mu_q)^2}{\sigma_q^2})$$

$$\psi \stackrel{\text{def}}{=} -\frac{1}{2}\frac{((\sum\sum wx)-\mu_p)^2}{\sigma_p^2} - \frac{1}{2}\frac{((\sum\sum wx)-\mu_q)^2}{\sigma_q^2}$$

$$\theta \stackrel{\text{def}}{=} -\frac{x((\sum\sum wx)-\mu_p)^2}{\sigma_p^2} - \frac{x((\sum\sum wx)-\mu_q)^2}{\sigma_q^2}$$

Unfortunately, we did not have sufficient time to test the loss function. While we believe the function has the potential to encourage a higher degree of filter specialization, it would introduce the need for additional training. Additionally, it might be desirable to have filters that detect shared parts. The proposed approach is built upon estimating the actual distribution by fitting a normal distribution. Therefore, it is conceivable that the approach incurs too significant inaccuracies, making it unviable.

6.3 Future Work

We recommend that future work continues generating part detections by using our network dissection-inspired approach from section 4.7 since we found this to be the most viable way to generate part detections. For a joint-structure system, we do not make any specific recommendations beyond looking further into spatial relationship modeling. We have already written about this in section 5.1 as a result of a literature search we did. We have, however, not had time to properly analyze the methods in practical experiments, which means some methods might be dead-ends. Future work should look into these and figure out what works and does not work for a joint-structure system. In section 6.2 we also introduce many different problems that should be addressed by future work. Some of these tasks are naturally more important than others, and we also mention specific suggestions as to what tasks future work should be focused on below.

6.3.1 Expansion Of Filter Selection

The first suggestion is for an expansion of filter selection since we see that our approach currently cannot find all relevant filters. This can be clearly seen for our combined car probability map examples in Figure 5.5. It seems to us that we manage to find very prominent car parts such as wheels and windows, but we also miss texturally homogeneous

regions such as the chassis of cars. We think this is a consequence of only picking filters with large Bhattacharyya distances. Since texturally homogeneous regions are more likely to be a shared concept between many classes, filters that represent them would also have small Bhattacharyya distances, meaning they will not be considered. Future work should therefore look into also finding these filters because they can be important in describing a class even though they represent a shared concept.

It could also be that these filters do not exist in the Conv4-3 layer we attempted to use in our practical experiments. Existing work shows that lower-level concepts such as textures are found in earlier layers of DCNNs, which means filters that detect car doors could be found there instead [Bau+17; Yos+15]. Therefore, future work should also look into multiple layers when finding filters because it is very likely that different concepts are found at different layers.

6.3.2 Expanding Into Detection

The second suggestion is for the joint-structure classifiers to be further expanded into detectors. We believe that Computer Vision systems today should build upon the ability to localize, and while we mainly managed to complete the classification part of detection, our approaches still needs to perform localization. Some discussion and experiments that we make related to detection can be found in chapter 7.

6.3.3 Handling Object Scales

The third suggestion is to look into how to expand the object scales which the system can process. As it currently stands, the system is limited in the size of objects it can recognize. For instance, if the wheels typically consist of n pixels, it is not given that the system can recognize a wheel consisting of n^3 pixels. This suggestion is linked to the second suggestion in the sense that considering more layers will give access to different sizes of receptive fields. However, this only represents an improvement and not a complete solution to the problem.

6.3.4 Modern Architectures

The fourth and final suggestion is to modernize our part detector approach. When finding filters, we only attempted to use a VGG16, an older DCNN model, and state-of-the-art has moved on since then. More impressive results and better part detectors can likely be found in newer architectures like AutoEncoders or Generative Adversarial Networks.

Chapter 7

From Classification To Detection: Concepts, Approaches, And Experiments

This chapter discusses how the joint-structure classifiers can be extended to perform object detection. We create a chapter to discuss the extension because we want to call attention to the importance of moving beyond classification if our approach is to find use in real applications.

7.1 Concepts Related To Detection

To understand how our classifiers can be extended to perform detection, we examine the review paper by Zou et al. [Zou+19]. The following subsections introduce the different detection concepts with which we are familiar. We only summarize some concepts Zou et al. mention related to performing detection. The review paper also covers topics related to context and speed-ups, which can be of interest for future work.

7.1.1 Sliding Window Detection

An easy way to extend image classifiers for tasks that require localization is to use a sliding window approach. The idea is to apply a classifier trained for a specific image size $N \times N$ to each location of a larger image of size $M \times M$, where $M > N$. Each region of the larger image must then be assigned a classification score. Such a sliding window classifier is illustrated in Figure 7.2.

There are different ways that the output classification score can be mapped back to the larger image. One way is to assign each pixel the classifier is centered with the current classification output. In that case, each pixel in the original $M \times M$ image gets assigned only one classification score. Another way is to assign the entire $N \times N$ region the classifier sees with the current classification output. This form of mapping requires further aggregation because it will generate multiple predictions for each position of the original

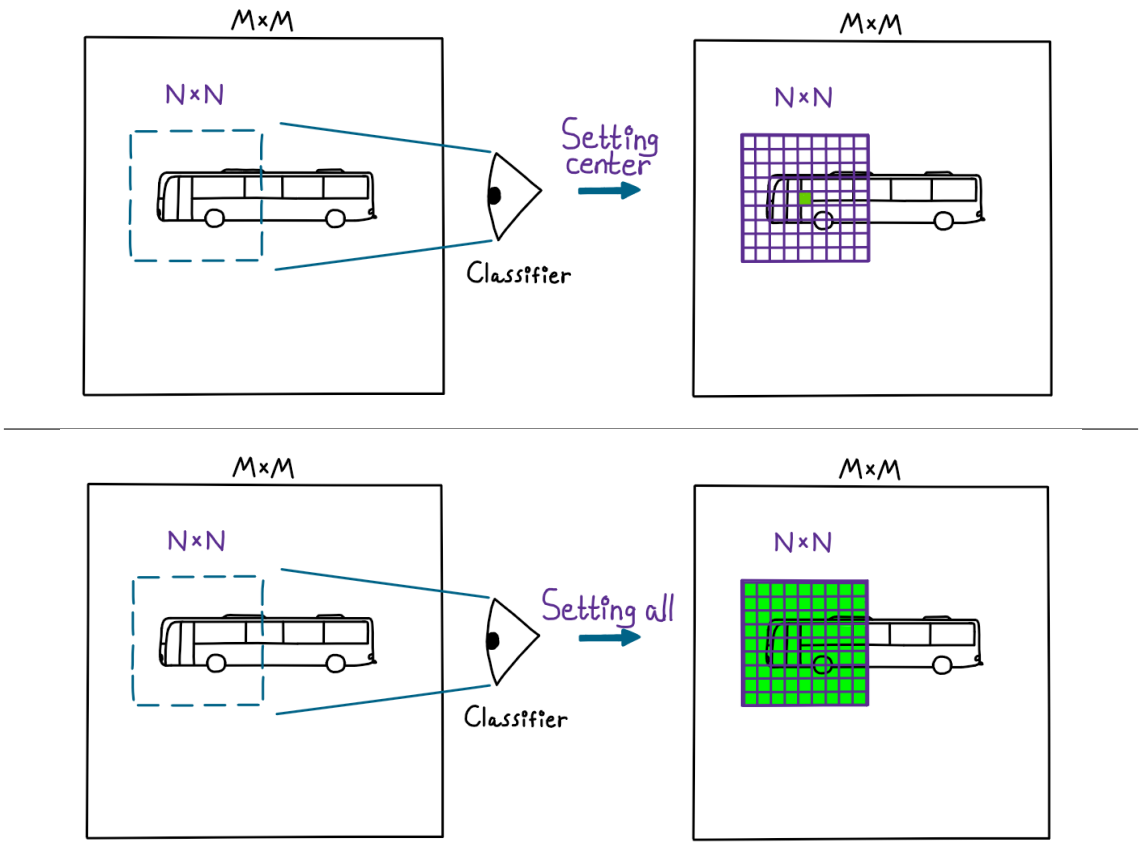


Figure 7.1: Example of two different ways to assign the decision output from a sliding window classifier. In this case, the classifier outputs a binary decision, but probabilities can also be assigned. In the top row, the pixel at the center of the sliding window is assigned the classifier's output. In the bottom row, the entire region of pixels inside the sliding window is assigned the classifier's output. In the case of an entire region being set, further aggregation would be required because each pixel will have multiple decisions.

$M \times M$ image. Examples of ways to aggregate could, for example, be simple mathematical operations like taking the mean, maximum and minimum or using a heuristic like applying a majority vote. We illustrate the different methods of assigning the classifier decision in Figure 7.1.

7.1.2 Image Pyramid Detection

The primary issue with a sliding window approach is the limitations related to object size. Since the sliding window classifier takes in a specific input size, it will be unfamiliar with the smaller and larger object sizes that can appear. In the case of objects much smaller than the sliding window, feature mixing[1] will happen. For objects larger than the sliding window, the results would depend on the training and robustness of the classifier, such as if the classifier can recognize that portions of an object belong to a larger object. In our approach, we used a VGG16 which was pre-trained on ImageNet. Therefore, if we were to perform sliding window detection with our approach, we would most likely see the best detection performance on cars close to the size used in ImageNet.

[1] We described feature mixing related to classification as a visual task in section 2.1

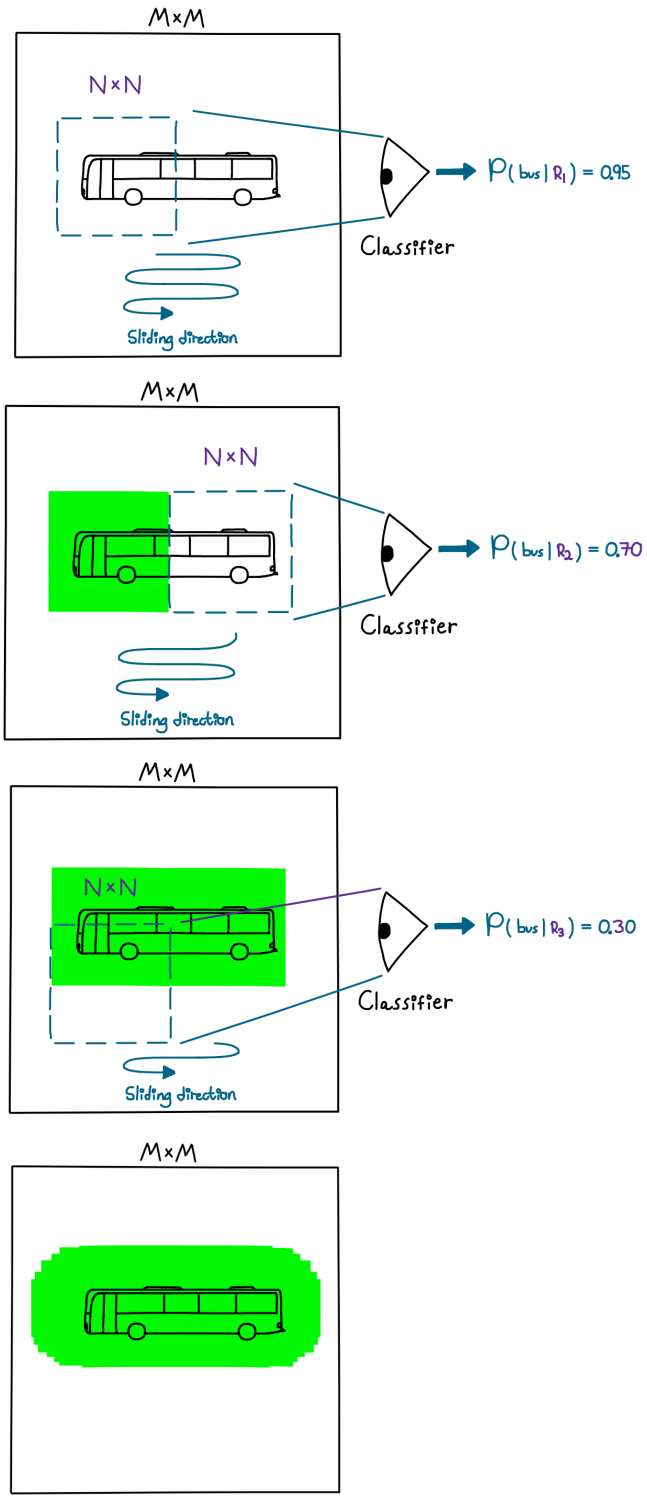


Figure 7.2: Example of how a classifier can be used to localize an object. The rows represent a random progression of the sliding window going chronologically from top to bottom. The entire sub-region is classified in this example when the classifier generates a probability > 0.50. The example uses the max operator to resolve multiple assigned values per pixel. The final row shows how the final detection result might look.

One way to address the size limitation of sliding window approaches is to create image pyramids, i.e., a stack of images at different scales. The largest image is at the bottom of such a stack, and the images become smaller as one traverses up the stack. By scanning through the entire pyramid of images with a sliding window, the classifier will have the possibility of seeing objects that were otherwise too large in the smaller images. The image pyramid detection is illustrated in Figure 7.3. A downside of pyramid detection is that the approach requires more computations than a baseline sliding window approach.

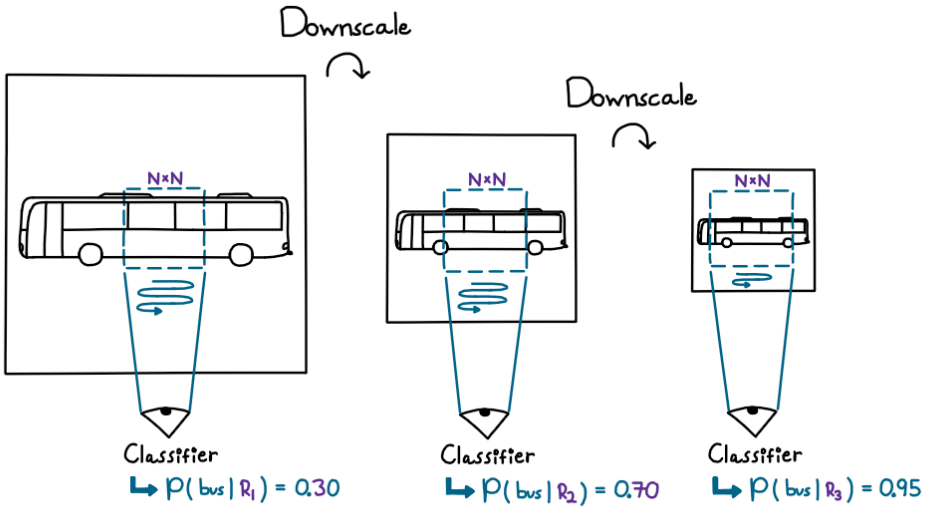


Figure 7.3: Example of how a pyramid of image scales can help a classifier to recognize objects which are too large otherwise.

7.1.3 Region Proposal Detection

Another method of performing detection with a classifier is using a generator that proposes regions where ”any” objects are located and then using the classifier only on the proposed regions. The main idea behind region proposals is for the generator and classifier to complement each other. The generator should propose all objects with little computational power, even if some proposals are false, which translates into high recall and low precision. The classifier that checks the proposed regions is usually more computationally expensive and very precise in its classification. A well-known early approach for region proposal is the selective search algorithm by Uijlings et al. [Uij+13]. We illustrate the idea of using region proposals in Figure 7.4.

Generating proposals can naturally be extended to our work because we do not do anything that deviates from existing architectures. We illustrate the idea of using region proposals in Figure 7.4. This detection is an improvement over sliding windows because the method can focus the heavy computations on only regions that are good candidates. One potential issue of adapting region proposals is that our explainable system could rely on using unexplainable proposals if generated by deep networks or some other hidden generator. However, this would only be for the errors that happen as a result of the

region proposals. Suppose the failures caused by the region proposals are rarer than the failures caused by the explainable system. In that case, it could still be okay to employ an inexplainable generator since most failures would still be explainable.

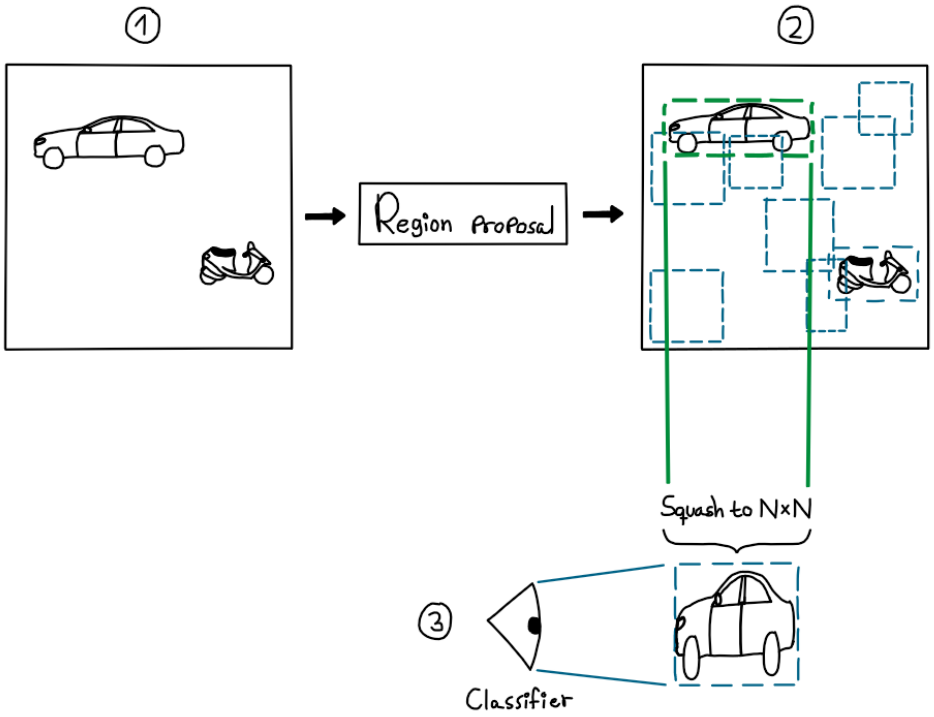


Figure 7.4: Example of how region proposals can be used for detection. One generator is used to localize "any" object, and the image patches with objects are passed down to a classifier.

7.1.4 Bounding Box Regression Detection

Object detection through regression is another technique that predicts bounding boxes around a preset amount of output regions. This is typically seen in the single-stage deep networks mentioned in chapter 3, e.g. YOLO [Red+16].

The modern approaches that perform detection through regression mainly use neural networks to generate the bounding box predictions directly from internal features. Suppose we are to adopt a form of regression. In that case, part detections should not be processed through a neural network because it would make the system's decisions unexplainable again, i.e., defeating the whole point. Adopting explainable bounding box regression would require learning some form of a relation between parts and how they match an overall object through an explainable method. For example, the left side of the bounding box is always to the left of the wheels, and the top is always above the roof. The paper by Sun et al. [SKY22] can be used for some inspiration on how to do this. They use shape priors, which could be used together with part detections as pre-defined regions to reason about an object's bounding box.

One benefit of the modern regression approaches is the removal of maximum sizes of

objects that can be detected. Since regression-based methods process whole images by scaling them down to fit in a deep network and then predict bounding boxes from internal features, they are not constrained by maximum object sizes like a sliding window classifier is. However, they still suffer from being unable to detect smaller objects. This inability likely happens because the receptive field at later layers is large and contains many different non-object contexts for smaller objects. It seems that the ways to alleviate the issue of finding small objects are either through using anchor boxes[2] or using internal features from earlier layers.

There seem to be some benefits related to deep network regression approaches, and we see the following as essential points that can be adopted in future work:

(1) Using a DCNN to generate part detections allows us to solve the problem of maximum object scales like the modern regression approaches because images can be scaled down to fit the input size of the network.

(2) We already considered multi-layered part detectors in section 6.2.3. This appears to be essential for robustness when detecting smaller objects and should be investigated.

(3) We can adopt regression by using our part detections as pre-defined proposals to calculate around. This matches the modern regression approaches as they also use pre-defined regions. However, we need to find a way to avoid relying on neural networks to do the regression. One idea, as mentioned earlier, is to use a shape prior or some other explainable method. It would be wise for future work to study existing literature on how to perform explainable bounding box regression from part detections.

(4) The fact that images can be scaled down and multiple layers can be used to detect smaller objects suggests that there could be an ideal input size of network architectures that is computationally efficient and generates good enough part detectors for smaller scales in the earlier layers.

7.1.5 Summarizing Our Thoughts

We have highlighted some detection approaches in the previous paragraphs. It seems to us that the focus for future work should be to investigate detection through either region proposals or bounding box regression, simply because these appear not to have the disadvantages of the other approaches. Sliding windows suffer from being computationally expensive and do not handle objects at larger scales, while pyramid detection appears more like a band-aid solution for sliding windows and is even more computationally expensive.

7.2 Sliding Window Detection Experiment

To briefly test detection, we implemented the simplest sliding window operation possible and downloaded open-source images from Wikimedia to evaluate the performance. The

[2] Anchor boxes are a pre-defined set of bounding box templates that vary in shape and scale, during predictions they are refined to match objects by predicting offsets

classifier we used for our sliding window was the Bag of Words classifier presented in section 5.3. From the measured results in section 6.1 we see that setting a threshold of $t = 0.25$ appeared to function well for the classifier, so we also used this threshold when detecting. Since many different steps are included in our approach, we also illustrate the entire process in Figure 7.5 as a reminder.

When sliding our classifier over the images, we classify 224x224 sub-regions since that is the input size for the VGG16. If a classification appears, we attempt both methods of assignment, i.e., we try both marking the entire sub-region in the sliding window and only marking the center pixel. To combine pixels where multiple classifications appear, we use the max operator. We slide our classifier across the entire image, meaning we use a step size of 1. We also add 112 pixels of zero padding to the images before sliding the classifier across. The result of our detection experiment can be seen in Figure 7.6, Figure 7.7, and Figure 7.8

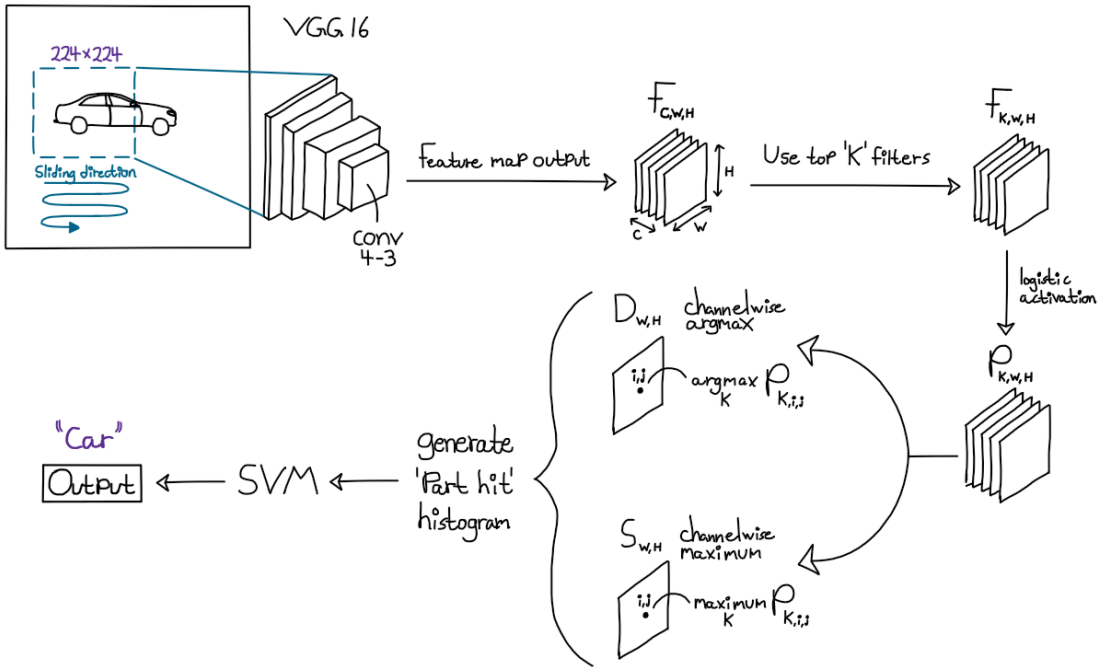


Figure 7.5: The full processing pipeline we have developed. We first select the filters we found to activate for car parts in Conv 4-3. The filter outputs are activated by a logistic function which makes them probabilistic. The probabilities are further processed with a simple heuristic to generate 'part hits', and is then classified using an SVM based BoW approach.

Figure 7.6: *Original image from [Com22] - Public domain, via Wikimedia Commons*
Example of sliding window detection. The first row is the original image with the classifier window size marked with a yellow box. The second row shows the locations of the original image that the classifier fired over, with the left image being center pixel assignment and the right image being region assignment. The third row shows the assignment images overlaid on the original image with red color.

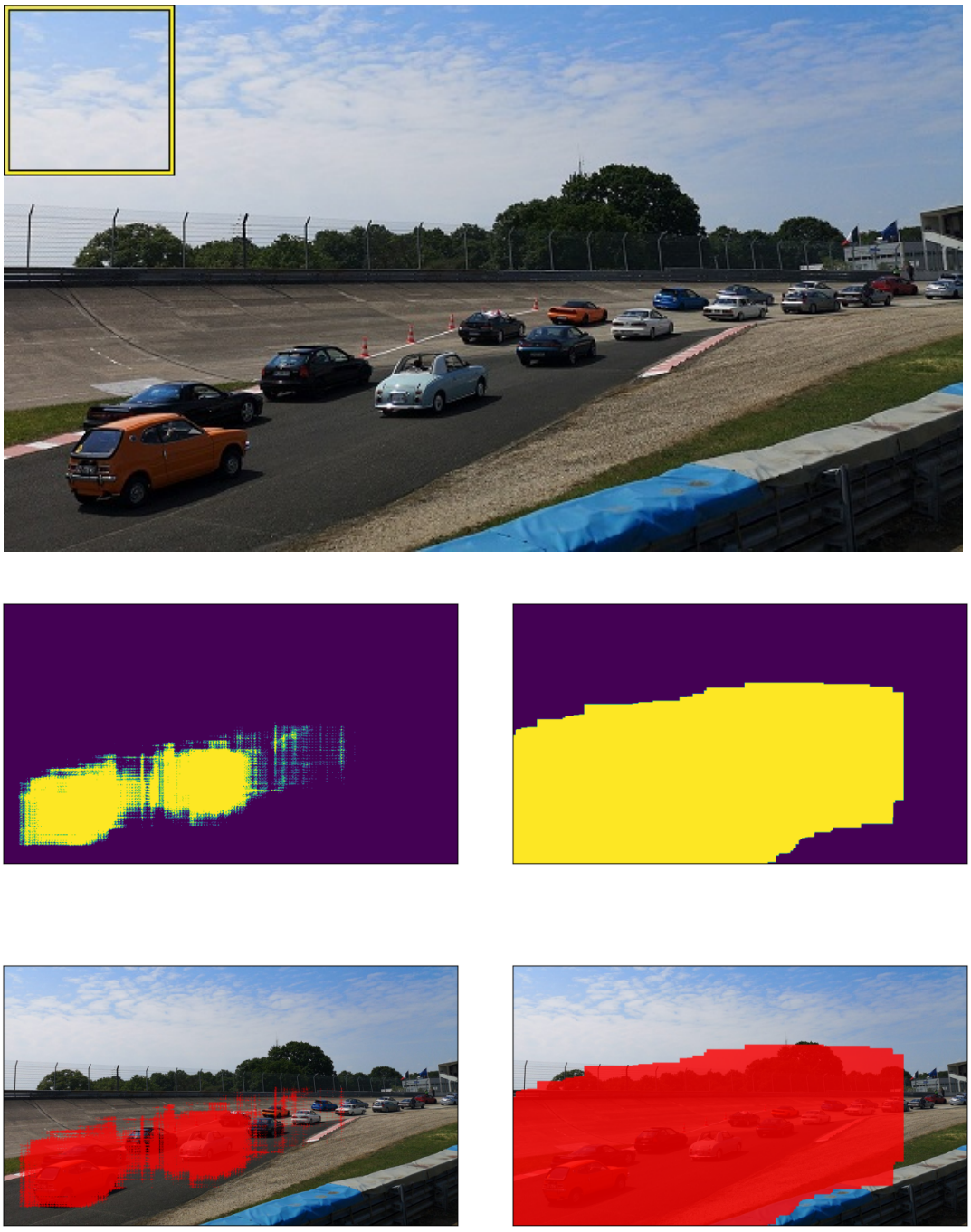


Figure 7.7: *Original image from [Com21] - Public domain, via Wikimedia Commons*
Example of sliding window detection. The first row is the original image with the classifier window size marked with a yellow box. The second row shows the locations of the original image that the classifier fired over, with the left image being center pixel assignment and the right image being region assignment. The third row shows the assignment images overlaid on the original image with red color.

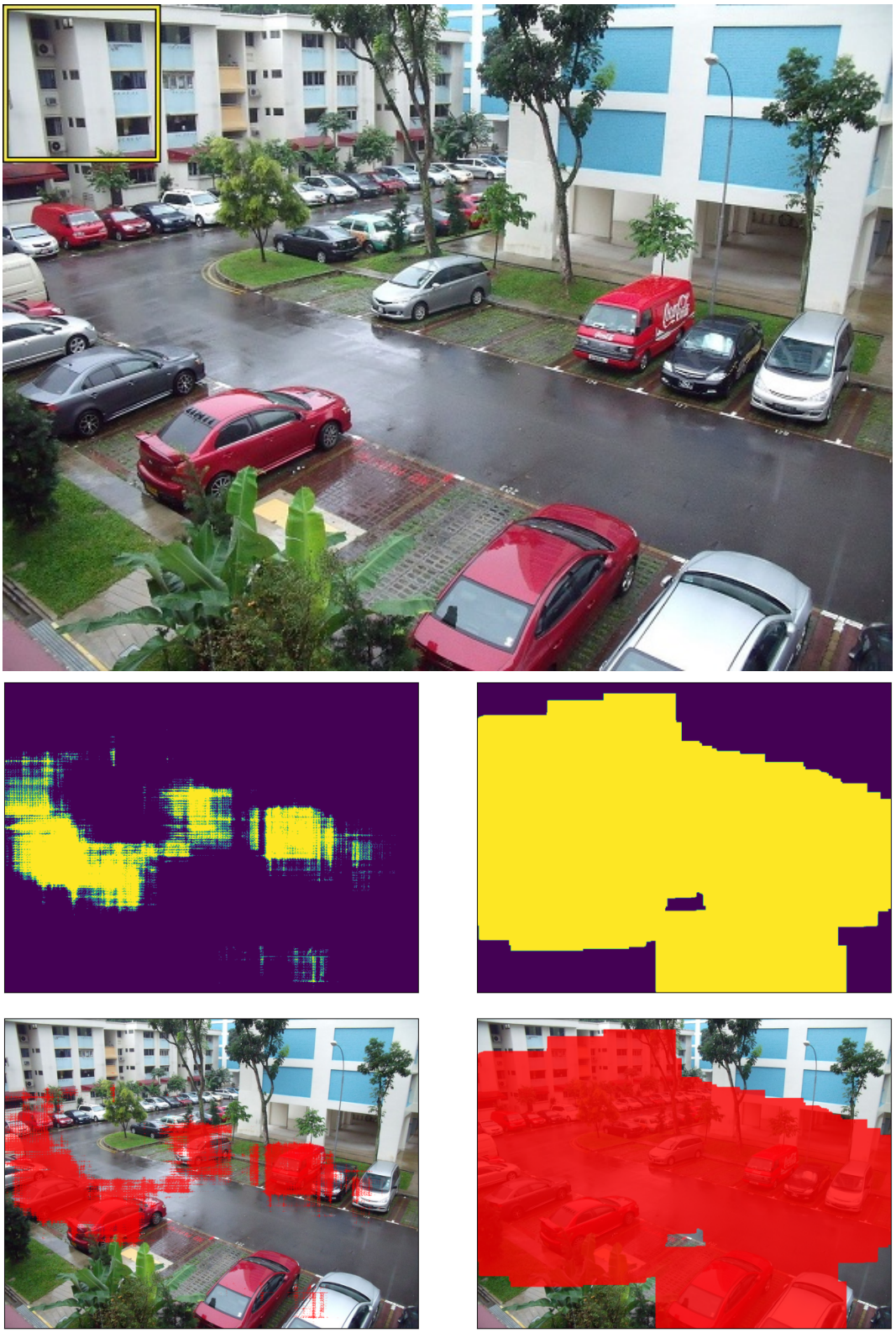


Figure 7.8: *Original image from [Com20a] - Public domain, via Wikimedia Commons*
Example of sliding window detection. The first row is the original image with the classifier window size marked with a yellow box. The second row shows the locations of the original image that the classifier fired over, with the left image being center pixel assignment and the right image being region assignment. The third row shows the assignment images overlaid on the original image with red color.

Chapter 8

Conclusion

This thesis has been devoted to investigating whether object detection can be improved using compositionality. The primary motivation was to adapt the potential benefits of using a structured part model with the advantages of automatically learning parts using deep learning. In chapter 4 we described our pursuit to utilize the internal representations of DCNNs as part detectors. Followingly, in chapter 5, we described preliminary investigations into applying the part detectors in a joint structure classifier.

A part detector system was created based on the literature reviews conducted during the pre-project and the thesis. Several methods found in relevant papers were presented in section 4.2. We then conducted experiments on methods from papers that we considered most likely to yield results given the time frame of the thesis. We first tried "Approach A - Feature Vectors" in section 4.4, which was built on clustering feature vectors to create a dictionary of parts, based on the work presented in the papers [Wan+15; Lia+16; Wan+17a; Kor+20a; Kor+21a]. Next, we tried "Approach B - Pattern Mining" in section 4.5, where we implemented a method based on pattern mining of feature maps as proposed by the paper [Zha+19]. Then, in section 4.6, we tried "Approach C - Activation Masking" from the paper [ZWZ18]. Approach C was a training-based method that uses masks on feature maps to enforce activation uniqueness.

Finally, in section 4.7, we implemented our own method "Approach D - DCNN Filters". Our method is based upon utilizing filters in CNNs as part detectors. We find part detector filters by measuring the maximum activations of feature maps to form class distributions. This process is done for each filter in one layer of a DCNN. We then fit a logistic regressor to the distributions, which outputs a probability for the presence of a part in a location. This output can then be processed further by a joint structure system, such as the classifiers described in chapter 5.

The joint structure system, which is the system that is intended for summarizing the information from the part detection subsystem, ended up being relatively simple due to time constraints. We performed an extensive literature review on approaches to model the spatial relationships between the part hits generated from the part detectors, which we outline in section 5.1. Followingly, in section 5.2, we described a heuristic for creating

a probability map by using the part detections from section 4.7. We then described two approaches to create a joint structure classifier based on BoVW. The first BoVW approach in section 5.3 (”Approach A - Bag of Words Baseline”) used the probability map to generate histograms of part hit counts, which was then used to train an SVM for classification. The second BoVW approach (”Approach B - Bag of Visual Words Spatial Extension”), in section 5.4, was similar but incorporated spatial extensions described in [ZM10].

The novel contribution of our thesis is using filters as part detectors which we introduced in section 4.7. The idea that filters can be used as part detectors is not new, but the methodology we used for finding these filters is, to our knowledge, novel. We took inspiration from the papers of Bau et al. [Bau+17; Bau+20] and Zhang et al. [ZWZ18; Zha+20] when developing the filter-based approach. The main inspiration comes from the Bau et al. papers, which show that concepts can be mapped to filters. The inspiration taken from the Zhang et al. papers was the observation that maximum positions of feature maps have valuable properties. The contributions that set our method apart from other work are the fact that we only need image-level labeling and the use of the Bhattacharyya distance to measure class separation. We also transform filter outputs into probabilities using logistic regression.

Another contribution of our thesis is the performed literature studies. We have done extensive literature searches and read through an extensive amount of papers. Many discovered papers were not explicitly used in the practical experiments, but they are listed for convenience in Appendix B. The papers we found most interesting are those seen in the different related works sections. In chapter 3 we introduce papers related to the field of structured object detection, in section 4.2 we introduce specific papers related to generating part detections from DCNNs and in section 5.1 we introduce several papers related to modeling and analyzing spatial relationships. In chapter 4 we also attempted to re-implement some of the papers we found to generate part detections. Due to our use of a different dataset from the papers, we cannot directly test their stated results. However, our implementation experiences can still be used as a guideline on what to expect from the approaches.

We believe that the approach of using filter-based part detections with a joint-structure detection system holds significant potential. There are, however, two major limitations in our work that we consider of particular importance to address; the first limitation is that the system we created only performs classification in its current form, i.e., it is a compositional classification system. The intention was initially to develop a detector, but we did not have sufficient time. Because of this, we also provide some additional discussion about extending the current system for detection in chapter 7.

The second limitation is the limited object scales the system can consider. Two separate things cause the issue of object scale: The first is that only one layer is selected, which limits us to consider parts of a scale close to the layer’s receptive field. The second is related to the input image size, which results in the system only being able to detect parts from objects of a similar scale to ones already seen in training. For a more thorough

description of these limitations and others, we refer the reader to section 6.2.

Overall, we believe that the filter-based solution is a valuable contribution. As stated earlier, we could not implement a more complex joint structure system in the given time frame. However, we have proposed approaches for how such an endeavor may be completed in section 6.2. In addition, the part detectors and joint structure system chapters have cited numerous papers, which should be helpful. We have also indicated how we believe the research should proceed in section 6.3. We hope and believe our thesis represents a good baseline for further work in designing and implementing a joint structure system with practical applications.

Bibliography

[ACR12] Denis Allard, Alessandro Comunian, and Philippe Renard. "Probability aggregation methods in geoscience". In: *Mathematical Geosciences* 44.5 (2012), pp. 545–581.

[ANS19] André Araujo, Wade Norris, and Jack Sim. "Computing Receptive Fields of Convolutional Neural Networks". In: *Distill* (2019). https://distill.pub/2019/computing-receptive-fields. DOI: 10.23915/distill.00021.

[Bau+17] David Bau et al. "Network dissection: Quantifying interpretability of deep visual representations". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 6541–6549.

[Bau+20] David Bau et al. "Understanding the role of individual units in a deep neural network". In: *Proceedings of the National Academy of Sciences* 117.48 (2020), pp. 30071–30078.

[BB19] Wieland Brendel and Matthias Bethge. "Approximating cnns with bag-of-local-features models works surprisingly well on imagenet". In: *arXiv preprint arXiv:1904.00760* (2019).

[BCT16] Michael Blot, Matthieu Cord, and Nicolas Thome. "Max-min convolutional neural networks for image classification". In: *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2016, pp. 3678–3682.

[Bha46] A. Bhattacharyya. "On a Measure of Divergence between Two Multinomial Populations". In: *Sankhyā: The Indian Journal of Statistics (1933-1960)* 7.4 (1946), pp. 401–406. ISSN: 00364452. URL: http://www.jstor.org/stable/25047882 (visited on 05/04/2022).

[BP22] Roger Bivand and Edzer Pebesma. *Spatial Data Science with applications in R*. 2022. URL: https://keen-swartz-3146c4.netlify.app/index.html#preface.

[Cha+97] Shih-Hsu Chang et al. "Fast algorithm for point pattern matching: invariant to translations, rotations and scale changes". In: *Pattern recognition* 30.2 (1997), pp. 311–320.

[Che+14] Xianjie Chen et al. "Detect what you can: Detecting and representing objects using holistic models and body parts". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 1971–1978.

[Com20a] Wikimedia Commons. *File:CarPark.JPG — Wikimedia Commons, the free media repository*. [Online; accessed 29-June-2022]. 2020. URL: https://commons.wikimedia.org/w/index.php?title=File:CarPark.JPG&oldid=454544930.

[Com20b] Wikimedia Commons. *File:Gabor-ocr.png — Wikimedia Commons, the free media repository*. [Online; accessed 26-June-2022]. 2020. URL: https://commons.wikimedia.org/w/index.php?title=File:Gabor-ocr.png&oldid=467753310.

[Com21] Wikimedia Commons. *File:Jap'n car festival Parade JDM - 48315288952.jpg — Wikimedia Commons, the free media repository*. [Online; accessed 29-June-2022]. 2021. URL: https://commons.wikimedia.org/w/index.php?title=File:Jap%27n_car_festival_Parade_JDM_-_48315288952.jpg&oldid=587495361.

[Com22] Wikimedia Commons. *File:Coches en Mazatlán, 25 de febrero de 2022.jpg — Wikimedia Commons, the free media repository*. [Online; accessed 29-June-2022]. 2022. URL: https://commons.wikimedia.org/w/index.php?title=File:Coches_en_Mazatl%C3%A1n,_25_de_febrero_de_2022.jpg&oldid=658412833.

[Cos+20] Christian Cosgrove et al. "Robustness Out of the Box: Compositional Representations Naturally Defend Against Black-Box Patch Attacks". In: *arXiv preprint arXiv:2012.00558* (2020).

[Dai+14] Jifeng Dai et al. "Unsupervised learning of dictionaries of hierarchical compositional models". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 2505–2512.

[Deo+21] Tejas Y Deo et al. "A White-Box SVM Framework and its Swarm-Based Optimization for Supervision of Toothed Milling Cutter through Characterization of Spindle Vibrations". In: *arXiv preprint arXiv:2112.08421* (2021).

[DT05] Navneet Dalal and Bill Triggs. "Histograms of oriented gradients for human detection". In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*. Vol. 1. Ieee. 2005, pp. 886–893.

[FBL14] Sanja Fidler, Marko Boben, and Ales Leonardis. "Learning a hierarchical compositional shape vocabulary for multi-class object representation". In: *arXiv preprint arXiv:1408.5516* (2014).

[Fel+10] Pedro F Felzenszwalb et al. "Object detection with discriminatively trained part-based models". In: *IEEE transactions on pattern analysis and machine intelligence* 32.9 (2010), pp. 1627–1645.

[FGM10] Pedro F Felzenszwalb, Ross B Girshick, and David McAllester. "Cascade object detection with deformable part models". In: *2010 IEEE Computer society conference on computer vision and pattern recognition*. Ieee. 2010, pp. 2241–2248.

[FMR08] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. "A discriminatively trained, multiscale, deformable part model". In: *2008 IEEE conference on computer vision and pattern recognition*. Ieee. 2008, pp. 1–8.

[Gat+96] Anthony C Gatrell et al. "Spatial point pattern analysis and its application in geographical epidemiology". In: *Transactions of the Institute of British geographers* (1996), pp. 256–274.

[Gei+18] Robert Geirhos et al. "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness". In: *arXiv preprint arXiv:1811.12231* (2018).

[GFM11] Ross Girshick, Pedro Felzenszwalb, and David McAllester. "Object detection with grammar models". In: *Advances in neural information processing systems* 24 (2011).

[GGMF18] Abel Gonzalez-Garcia, Davide Modolo, and Vittorio Ferrari. "Objects as context for detecting their semantic parts". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 6907–6916.

[Gir+14] Ross Girshick et al. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014, pp. 580–587.

[Gir15] Ross Girshick. "Fast r-cnn". In: *Proceedings of the IEEE international conference on computer vision*. 2015, pp. 1440–1448.

[Gir+15] Ross Girshick et al. "Region-based convolutional networks for accurate object detection and segmentation". In: *IEEE transactions on pattern analysis and machine intelligence* 38.1 (2015), pp. 142–158.

[He+15] Kaiming He et al. "Spatial pyramid pooling in deep convolutional networks for visual recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 37.9 (2015), pp. 1904–1916.

[He+21] Ju He et al. "PartImageNet: A Large, High-Quality Dataset of Parts". In: *arXiv preprint arXiv:2112.00933* (2021).

[HKY21] Ju He, Adam Kortylewski, and Alan Yuille. "COMPAS: Representation Learning with Compositional Part Sharing for Few-Shot Classification". In: *arXiv preprint arXiv:2101.11878* (2021).

[HP19] Xiangteng He and Yuxin Peng. "Fine-grained visual-textual representation learning". In: *IEEE Transactions on Circuits and Systems for Video Technology* 30.2 (2019), pp. 520–531.

[Hua+16] Shaoli Huang et al. "Part-stacked cnn for fine-grained visual categorization". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 1173–1182.

[Iee] *Requesting Permission to Reuse IEEE Material*. URL: `https://www.ieee.org/publications/rights/reqperm.html#obtaining-permission-through-rightslink`.

[Ji+21] Jinsheng Ji et al. "Adversarial erasing attention for fine-grained image classification". In: *Multimedia Tools and Applications* 80.15 (2021), pp. 22867–22889.

[Kor+19] Adam Kortylewski et al. "Greedy structure learning of hierarchical compositional models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 11612–11621.

[Kor+20a] Adam Kortylewski et al. "Combining compositional models and deep networks for robust object classification under occlusion". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2020, pp. 1333–1341.

[Kor+20b] Adam Kortylewski et al. "Compositional convolutional neural networks: A deep architecture with innate robustness to partial occlusion". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 8940–8949.

[Kor+21a] Adam Kortylewski et al. "Compositional convolutional neural networks: A robust and interpretable model for object recognition under occlusion". In: *International Journal of Computer Vision* 129.3 (2021), pp. 736–760.

[Kor+21b] Adam Kortylewski et al. "Compositional generative networks and robustness to perceptible image changes". In: *2021 55th Annual Conference on Information Sciences and Systems (CISS)*. IEEE. 2021, pp. 1–8.

[KPR20] Jung Uk Kim, Sungjune Park, and Yong Man Ro. "Towards human-like interpretable object detection via spatial relation encoding". In: *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2020, pp. 3284–3288.

[KR21] William Kaaby and Stig Andre Rosenlund. *Specialization project report*. Project report in TDT4501. Department of Computer Science, NTNU – Norwegian University of Science and Technology, 2021.

[Kra+13] Jonathan Krause et al. "3d object representations for fine-grained categorization". In: *Proceedings of the IEEE international conference on computer vision workshops*. 2013, pp. 554–561.

[Lap+19] Sebastian Lapuschkin et al. "Unmasking Clever Hans predictors and assessing what machines really learn". In: *Nature communications* 10.1 (2019), pp. 1–8.

[Lee+20] Cheng-Han Lee et al. "MaskGAN: Towards Diverse and Interactive Facial Image Manipulation". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.

[Li+22] Zhe Li et al. "Robust deep learning object recognition models rely on low frequency information in natural images". In: *bioRxiv* (2022).

[Lia+16] Renjie Liao et al. "Learning deep parsimonious representations". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. 2016, pp. 5083–5091.

[Lin+14] Tsung-Yi Lin et al. "Microsoft coco: Common objects in context". In: *European conference on computer vision*. Springer. 2014, pp. 740–755.

[Lin+17a] Tsung-Yi Lin et al. "Feature pyramid networks for object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2117–2125.

[Lin+17b] Tsung-Yi Lin et al. "Focal loss for dense object detection". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.

[Liu+16] Wei Liu et al. "Ssd: Single shot multibox detector". In: *European conference on computer vision*. Springer. 2016, pp. 21–37.

[LMH03] Baihua Li, Qinggang Meng, and Horst Holstein. "Point pattern matching and applications-a review". In: *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme-System Security and Assurance (Cat. No. 03CH37483)*. Vol. 1. IEEE. 2003, pp. 729–736.

[Low04] David G Lowe. "Distinctive image features from scale-invariant keypoints". In: *International journal of computer vision* 60.2 (2004), pp. 91–110.

[LSP06] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories".

In: *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*. Vol. 2. IEEE. 2006, pp. 2169–2178.

[Luo+16] Wenjie Luo et al. "Understanding the effective receptive field in deep convolutional neural networks". In: *Advances in neural information processing systems* 29 (2016).

[NLV16] David Novotny, Diane Larlus, and Andrea Vedaldi. "I have seen enough: Transferring parts across categories". In: *Proceedings of the British Machine Vision Conference (BMVC)*. Vol. 7. 2016.

[NM15] Mohammad Naeem and Pascal Matsakis. "Relative position descriptors". In: *Proceedings of the International Conference on Pattern Recognition Applications and Methods*. Vol. 1. 2015, pp. 286–295.

[OB07] Bjorn Ommer and Joachim M Buhmann. "Learning the compositional nature of visual objects". In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2007, pp. 1–8.

[Ots79] Nobuyuki Otsu. "A threshold selection method from gray-level histograms". In: *IEEE transactions on systems, man, and cybernetics* 9.1 (1979), pp. 62–66.

[Red+16] Joseph Redmon et al. "You only look once: Unified, real-time object detection". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 779–788.

[Ren+15] Shaoqing Ren et al. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems* 28 (2015).

[Rou87] Peter J Rousseeuw. "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis". In: *Journal of computational and applied mathematics* 20 (1987), pp. 53–65.

[Sch67] Fred C Schweppe. "On the Bhattacharyya distance and the divergence between Gaussian processes". In: *Information and Control* 11.4 (1967), pp. 373–395.

[SG20] Farhad Shakerin and Gopal Gupta. "White-box induction from SVM models: Explainable ai with logic programming". In: *Theory and Practice of Logic Programming* 20.5 (2020), pp. 656–670.

[SG21] Axel Sauer and Andreas Geiger. "Counterfactual generative networks". In: *arXiv preprint arXiv:2101.06046* (2021).

[SKY22] Yihong Sun, Adam Kortylewski, and Alan Yuille. "Amodal Segmentation Through Out-of-Task and Out-of-Distribution Generalization With a Bayesian

Model". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 1215–1224.

[The89] Charles W Therrien. *Decision estimation and classification: an introduction to pattern recognition and related topics*. John Wiley & Sons, Inc., 1989.

[Tor] *Resize*. URL: `https://pytorch.org/vision/main/generated/torchvision.transforms.functional.resize.html`.

[TRS21] Nontawat Tritrong, Pitchaporn Rewatbowornwong, and Supasorn Suwajanakorn. "Repurposing gans for one-shot semantic part segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 4475–4485.

[Uij+13] Jasper RR Uijlings et al. "Selective search for object recognition". In: *International journal of computer vision* 104.2 (2013), pp. 154–171.

[VJ01] Paul Viola and Michael Jones. "Rapid object detection using a boosted cascade of simple features". In: *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*. Vol. 1. Ieee. 2001, pp. I–I.

[VJ04] Paul Viola and Michael J Jones. "Robust real-time face detection". In: *International journal of computer vision* 57.2 (2004), pp. 137–154.

[Wah+11] Catherine Wah et al. "The caltech-ucsd birds-200-2011 dataset". In: (2011).

[Wan+15] Jianyu Wang et al. "Unsupervised learning of object semantic parts from internal states of cnns by population encoding". In: *arXiv preprint arXiv:1511.06855* (2015).

[Wan+17a] Jianyu Wang et al. "Detecting semantic parts on partially occluded objects". In: *arXiv preprint arXiv:1707.07819* (2017).

[Wan+17b] Jianyu Wang et al. "Visual concepts and compositional voting". In: *arXiv preprint arXiv:1711.04451* (2017).

[Wan+20] Angtian Wang et al. "Robust object detection under occlusion with context-aware compositionalnets". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020, pp. 12645–12654.

[WM13] Thorsten Wiegand and Kirk A Moloney. *Handbook of spatial point-pattern analysis in ecology*. CRC press, 2013.

[WYL21] Chien-Yao Wang, I-Hau Yeh, and Hong-Yuan Mark Liao. "You only learn one representation: Unified network for multiple tasks". In: *arXiv preprint arXiv:2105.04206* (2021).

[Xu+19] Hang Xu et al. "Spatial-aware graph relation network for large-scale object detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9298–9307.

[Yan+20] Tiantian Yan et al. "Progressive learning for weakly supervised fine-grained classification". In: *Signal Processing* 171 (2020), p. 107519.

[Yao+18] Xiaojing Yao et al. "A spatial co-location mining algorithm that includes adaptive proximity improvements and distant instance references". In: *International Journal of Geographical Information Science* 32.5 (2018), pp. 980–1005.

[YL21] Alan L Yuille and Chenxi Liu. "Deep nets: What have they ever done for vision?" In: *International Journal of Computer Vision* 129.3 (2021), pp. 781–802.

[YN10] Yi Yang and Shawn Newsam. "Bag-of-visual-words and spatial extensions for land-use classification". In: *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*. 2010, pp. 270–279.

[Yos+15] Jason Yosinski et al. "Understanding neural networks through deep visualization". In: *arXiv preprint arXiv:1506.06579* (2015).

[Yu+22] Jiahui Yu et al. "Coca: Contrastive captioners are image-text foundation models". In: *arXiv preprint arXiv:2205.01917* (2022).

[Yua+21] Xiaoding Yuan et al. "Robust instance segmentation through reasoning about multi-object occlusion". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021, pp. 11141–11150.

[Zha+18] Quanshi Zhang et al. "Interpreting cnn knowledge via an explanatory graph". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[Zha+19] Jian Zhang et al. "Unsupervised part mining for fine-grained image classification". In: *arXiv preprint arXiv:1902.09941* (2019).

[Zha+20] Quanshi Zhang et al. "Interpretable CNNs for object classification". In: *IEEE transactions on pattern analysis and machine intelligence* 43.10 (2020), pp. 3416–3431.

[Zha+21] Tian Zhang et al. "Progressive Co-Attention Network for Fine-grained Visual Classification". In: *2021 International Conference on Visual Communications and Image Processing (VCIP)*. IEEE. 2021, pp. 1–5.

[Zha+22] Hao Zhang et al. "Dino: Detr with improved denoising anchor boxes for end-to-end object detection". In: *arXiv preprint arXiv:2203.03605* (2022).

[Zho+14] Bolei Zhou et al. "Object detectors emerge in deep scene cnns". In: *arXiv preprint arXiv:1412.6856* (2014).

[ZM10] Edmond Zhang and Michael Mayo. "Improving bag-of-words model with spatial information". In: *2010 25th International Conference of Image and Vision Computing New Zealand*. IEEE. 2010, pp. 1–8.

[Zou+19] Zhengxia Zou et al. "Object detection in 20 years: A survey". In: *arXiv preprint arXiv:1905.05055* (2019).

[ZSH20] Vitalii Zhelezniak, Aleksandar Savkov, and Nils Hammerla. "Estimating Mutual Information Between Dense Word Embeddings". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020, pp. 8361–8371.

[ZWZ18] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. "Interpretable convolutional neural networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 8827–8836.

Appendix A

Work Methodology

To ensure a structured and orderly approach to our work, we decided to follow the principles outlined in Scrum. This choice was motivated by the fact that both authors are familiar with the methodology in a professional setting and have found it to work quite well. However, we did not strictly follow each principle of Scrum at all times. This flexibility to Scrum principles was an intentional choice to increase the adaptability of our workflow. Since the team only consisted of two members, it is relatively more straightforward to adjust the Scrum process.

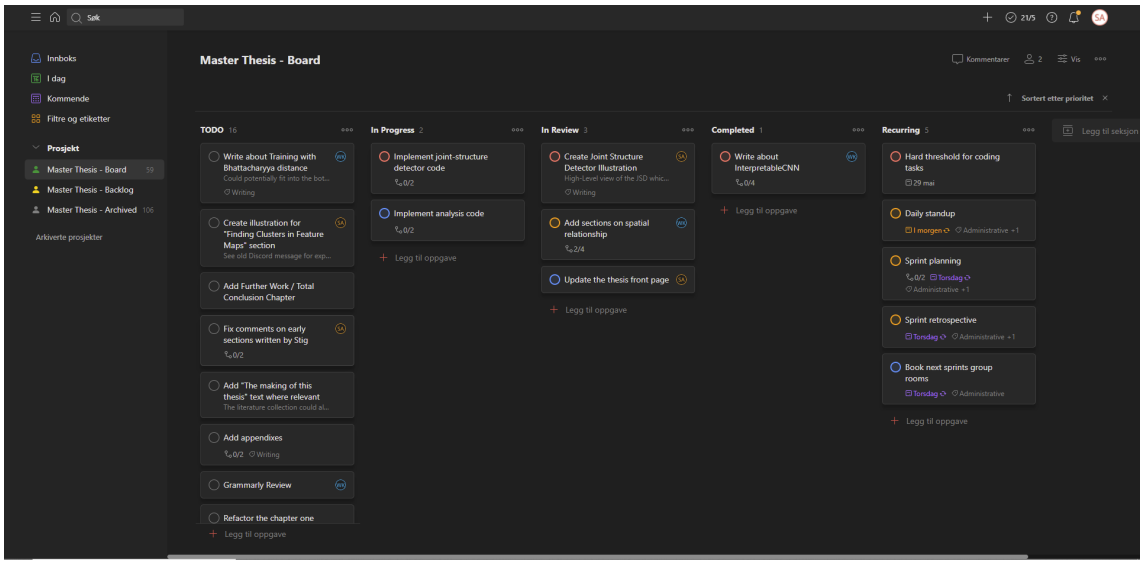


Figure A.1: This shows a snap-shot of the project board we used to structure the task items generated as part of the thesis. The board was created using a software tool called Todoist.

The centerpiece of the approach is the planning board, a snap-shot of which is shown in Figure A.1. The board was used to prioritize, schedule, and assign tasks. The board was a valuable tool to ensure that our Daily Scrum (daily meetings) were focused and efficient, and we were mostly able to adjourn the meetings before the 15-minute mark. Each Sprint (work period) had a duration of two weeks. Each sprint was initiated by two meetings; Sprint Planning and retrospective meetings. During Sprint Planning, we created work items and did an initial prioritization of the items. The retrospective meetings allowed

us to discuss which aspects of the work in the previous sprint we thought worked well and which things we would like to improve. We also had weekly meetings with our supervisor, Rudolf Mester. It was mainly in the context of these meetings that the flexible implementation of Scrum proved helpful. Once in a while, these meetings would produce changes in priorities which would be difficult to account for if one strictly follows Scrum. However, since we were pretty flexible concerning Scrum, we could adapt without great difficulty.

Appendix B

Literature Collections

B.1 Pre-Project Research Journal

Technical Report ~~AVG-XXX-XX~~

Research journal over Compositional models

William Emanuel Skreien Kaaby Stig Andre Rosenlund

V1.0, July 2, 2022
July 2, 2022

Contents

1 Record of papers

1.0.1 Reviews and Comparison papers

Visual Analogy: Deep Learning Versus Compositional Models Cosgrove et al. 2020

This paper discusses reasoning through visual analogy and compares the human approach to three different architectures: Relation network, Siamese network and compositional models. They find that compositional models produces the reasoning process most similar to the human approach. I think this paper is relevant as it may give some useful insights into how we can apply/think about the mentioned architectures.

Compositional Generative Networks and Robustness to Perceptible Image Changes Kortylewski et al. 2021b

Paper talks about the robustness of the new Compositional Convolutional Neural Networks. Relevant as a first pass read because it specifically relates directly to the paper that introduces this new type of networks in Kortylewski et al. 2021a.

CNN Architectures for Geometric Transformation-Invariant Feature Representation in Computer Vision: A Review Mumuni and Mumuni 2021

This paper reviews CNN approaches to achieve transformation-invariant feature representation. The paper motivates the search for transformation-invariant feature representation by showing that CNNs tends to struggle with misclassification of objects which have been subjected to affine transformations. The paper of course focuses on invariant representations in CNNs, but also gives a brief overview of invariance in other methods. I think this paper would give an useful overview of things to consider when dealing with transformation-invariant feature representation.

Occlusion Handling in Generic Object Detection: A Review Saleh, Szénási, and Vámossy 2021

The paper goes through the different ways of handling occlusion. In general terms it is relevant, but only for a light read unless we are looking for more specific ideas related to this topic.

Deep nets: What have they ever done for vision? Yuille and Liu 2021

Deep Networks revolutionized field, but differs from biological vision and have issues related to adversarial noise, viewpoint changes, occlusion, context sensitivity, stereo vision hazards and dataset biases.

We want to train and generalize robust networks that solve these problems, but the paradigm of the day is to create large datasets and "train away" issues. This is not a real solution, as there is a combinatorial explosion in visual problems due to endless spatial configurations and variance in properties like luminance that can occur in visual data.

Compositional models can be a solution to this as they learn the fundamental structure and rules of how something is put together, thus generalizing to combinatorial situations. A bonus is that these models also are explainable by nature.

Robustness Out of the Box: Compositional Representations Naturally Defend Against Black-Box Patch Attacks Cosgrove et al. 2020

The paper shows that Compositional Convolutional Neural Networks (CCNN) are robust towards black box patch attacks and outperform adversarial training defenses. Paper talks about how CCNNs struggle with similar classes, and introduces a new methodology to help against the problem called "part-based fine-tuning".

The paper cites the main relevant paper Kortylewski et al. 2021a, and the predecessors Kortylewski et al. 2020b and Wang et al. 2020

1.0.2 Pre Deep Learning papers

Voting by grouping dependent parts Yarlagadda, Monroy, and Ommer 2010

The paper discusses how Hough Voting methods are problematic because mutually dependent local observations are independently voting for global object properties. The paper proposes to expand on Hough Voting Methods by grouping mutually dependent parts.

Learning the compositional nature of visual objects Ommer and Buhmann 2007

The paper learns compositionality by extracting "atomic parts" and selecting the relevant ones from images. A bayesian model is trained with these atomic parts to learn compositional shape and category, for inference the model is used to categorize and localize objects.

Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories Lazebnik, Schmid, and Ponce 2006

The paper devises a method for scene recognition by extracting histograms from image regions. The regions are extraced in a pyramid pattern (the regions get smaller). According to the paper this improves the performance over bag of words at the time. Seeing as this type of approach gives better performance, the paper might be relevant if this type of pyramid extraction could be used in the newer compositional models somehow. Perhaps a relationship between parts is built up between pyramids? (Human region which is larger → Head region which is smaller → Nose region which is even smaller → etc...)

Object categorization by compositional graphical models Ommer and Buhmann 2005

The paper extracts region histogram features around interest points in images. These histogram features are used to learn up a codebook. A bayesian model is used to model compositionality relations, and performs categorization and localization.

1.0.3 Bag of Words papers

A Bayesian Hierarchical Model for Learning Natural Scene Categories Fei-Fei and Perona 2005

Each image is represented through subdivision of the image into a number of patches, each of which is represented by a codeword found in the generated vocabulary. The learning goal is to determine which word distribution corresponds to a given scene category. In other words, learn a scene model for all defined scenes (13).

Inference is done using a Bayesian decision method; given an unknown image consisting of a given distribution of patches (words), the probability of each scene is calculated based on the learned scene models. To solve the issue of a scene being able to contain multiple different "elements" (A cityscape may or may not contain trees for instance), the concept of themes is introduced. This results in the following abstraction hierarchy: scene > theme > codeword. The algorithm used is called latent Dirichlet allocation. A number of methods for region detection were attempted. Patches can be represented using either pixel gray values or a SIFT vector.

Video Google: A Text Retrieval Approach to Object Matching in Videos Sivic and Zisserman 2003

The paper uses affine invariant regions represented as SIFT descriptors (The Yellow and Cyan ellipses). IE. when 'region' is mentioned they mean the ellipses. One type of region is based on feature points and the other is based around a watershed version of the image.

A codebook is trained by generating regions for each training frame and clustering together similar regions by K-means. The codebook can then be used for lookup.

1.0.4 Fisher Vector papers

Image classification with the fisher vector: Theory and practice Sánchez et al. 2013

Alternative patch method from BoVW based around the "Fisher Kernel". A quote from the paper is: “...The FK combines the benefits of generative and discriminative approaches to pattern classification by deriving a kernel from a generative model of the data. In a nutshell, it consists in characterizing a sample by its deviation from the generative model....“.

1.0.5 Compositionality and Parts papers

Repurposing gans for one-shot semantic part segmentation Tritrong, Rewatbowornwong, and Suwajanakorn 2021

This paper explores whether GANs learn meaningful structural parts of objects. The aim of the paper is to create a model that can perform part segmentation using a very limited set of data with part labels. To do this, a pre-trained GAN is used. This seems quite interesting and should be explored further as it is a way to learn object parts by weak supervision.

COMPAS: Representation Learning with Compositional Part Sharing for Few-Shot Classification He, Kortylewski, and Yuille 2021

This paper explores the use of compositional models in few-shot image classification. The method they propose consists of two stages: meta-learning and meta-testing. Meta-learning is a stage wherein a dictionary of parts and a dictionary of spatial activation patterns of said parts are learned. They utilize the method described in Kortylewski et al. 2020b for learning the part dictionary. During the meta-testing stage, object representations are learned using the dictionaries learned in the previous stage. This paper is obviously relevant, and basically represents an extension of the earlier papers by Yuille.

Robust Instance Segmentation through Reasoning about Multi-Object Occlusion Yuan et al. 2021

Very relevant paper that expands on Kortylewski et al. 2021a by introducing instance segmentation with a reasoning mechanism that finds out about ordering of occlusions. The papers also mentions that they expand on Sun, Kortylewski, and Yuille 2021 by taking advantage of mutual relationships between close objects.

SR: This seems to be the newest paper directly related to the other papers in the Yuille & Kortylewski path. It builds upon the work done in the earlier papers Kortylewski et al. 2020a; Kortylewski et al. 2021a

Compositional convolutional neural networks: A robust and interpretable model for object recognition under occlusion Kortylewski et al. 2021a

Paper is about the combining of a Compositional Model with a Convolutional Neural Network into a model that is named "Compositional Convolutional Neural Network". The Compositional Convolutional Neural Network is directly trainable as it is differentiable, and can use popular backbones like VGG-16, ResNet50, etc. The compositional model is placed on the end of the backbone and acts as a generative model being fed the last convolutional layer feature activations. Seems very relevant as it is a modern compositional based model.

The paper cites Kortylewski et al. 2020a, which is a paper where a very simple combination between a DCNN and Compositional Model is made. A lot of the paper itself is an expansion of the work done in that paper, so they are very connected.

SR: This is one of the main papers that introduces the new architecture called "Compositional Convolutional Neural Networks". It expands on earlier versions of the papers Kortylewski et al. 2020b; Wang et al. 2020, but states themselves that this version is more refined and combines the two papers. We have therefore decided not to add these to the journal, but they are added to the paper folder in Onedrive.
The paper also has a section where they talk about related approaches in '2.3 Deep Compositional Models in Computer Vision' and argue against them/explain why this paper has the better approach. Some papers from the record are in that section. They for example discuss the idea about using the Deep Compositional Network in Tabernik et al. 2016, and argue against this. In general this is an important section to investigate and judge the conclusions drawn.

Weakly-Supervised Amodal Instance Segmentation with Compositional Priors Sun, Kortylewski, and Yuille 2021

The paper expands on Compositional Convolutional Neural Networks (CCNN) by explicitly representing objects as composition of parts. This allows the CCNN the ability to perform amodal instance segmentation (perception of occluded objects structure) and modal instance segmentation (perception of visible parts). Some supervision is required as objects are required to have bounding boxes in the dataset.

The following is new extensions:

1. Learning compositional shape priors of objects in varying 3D poses from modal bounding box supervision.
2. Predicting instance segmentation by integrating the compositional shape priors into the part-voting mechanism in the CompositionalNets.
3. Predicting amodal completion for both the bounding box and the instance segmentation mask by implementing compositional feature alignment in CompositionalNets.

The paper also cites the main relevant paper Kortylewski et al. 2021a, and the predecessors Kortylewski et al. 2020b and Wang et al. 2020

Independent Prototype Propagation Graph for Compositional Zero-Shot Recognition Ruis 2021

This paper proposes a method for zero-shot detection by composing learned object (shapes) and attribute (texture) primitives. An example given in the paper is learning a novel composition (zebra) using the object primitives from a horse and attribute primitives from a tiger. The method described in the paper is based on a GNN approach.

Unsupervised Part Discovery via Feature Alignment Guo et al. 2020

The paper shows approach for unsupervised learning of part detectors by using CNN-Backbone feature maps. Uses image of object and finds similar images of similar objects. Then uses feature alignment on the similar images to enforce coherence between these feature maps in training.

SR: This paper is interesting as it is another take on how to do part discovery from DCNN features

Combining compositional models and deep networks for robust object classification under occlusion Kortylewski et al. 2020a

The paper creates a mixed model that consists of both a Deep Convolutional Neural Network (DCNN) and Compositional Model (CM). The DCNN has problems with occlusion and mask attacks, while the CM has problems with being discriminative, so a combination uses the strength of both models.

The process is done in three steps;

1. In the first step a DCNN is trained for image classification, and afterwards the DCNN features are clustered into dictionaries. The dictionary components resemble object part detectors, and learn the spatial distribution of parts for each object class.
2. To account for different 3D poses / viewpoints of objects a mixture of compositional models is performed.
3. During runtime the classification of an image is done by DCNN, and depending on the uncertainty is, the compositional model could be used instead.

The paper cites many different papers, but specifically from the journal are papers like Kortylewski et al. 2019, Fidler, Boben, and Leonardis 2014 and Dai et al. 2014. The methodology they use to cluster DCNN features is cited from Wang et al. 2015, Liao et al. 2016 and Wang et al. 2017.

Greedy structure learning of hierarchical compositional models Kortylewski et al. 2019

Paper that learns a Hierarchical Compositional Model (HCM) from a set of images which contains the object. The method avoids making strong a-priori assumptions about the object‘s geometric structure which they say prior methods had to make. The method can also learn from training data that has not been segmented to separate the object from the background.

Uses a greedy structure learning framework to learn the HCM. The learning is done in two phases: (1) Bottom-up part learning. (2) Top-down model composition.

Uses an Active Basis Model (ABM) under the learning process. The ABM is a probabilistic generative model that models an object‘s variability in shape and appearance. In the paper the model is also generalized into a multi-layered compositional model.

Detecting semantic parts on partially occluded objects Wang et al. 2017

A voting based algorithm that uses the 'visual concepts' (parts) defined in Wang et al. 2015 to detect overall objects. Spatial relationships is also handled in this algorithm. Relevant as it is one of the DCNN feature clustering papers cited by Kortylewski et al. 2020a.

SR: This is an example of a simple algorithm that uses DCNN feature parts to predict objects by voting

Learning deep parsimonious representations Liao et al. 2016

The paper seems to propose a kind of regularization that encourages 'parsimonious' (i think this word means 'small number of'/ 'sparse') representations. Relevant as it is one of the DCNN feature clustering papers cited by Kortylewski et al. 2020a.

Towards deep compositional networks Tabernik et al. 2016

Paper that introduces a variant of Convolutional Neural Networks called Compositional Network that also model spatial relationship between features. The new Compositional Network is based on a normal neural network setup, but instead of modeling a neuron just as a weighted sum, they also add a 2D Gaussian feature for each unit that is learned. The network can be trained like any regular neural network with loss function and backpropogation.

SR: This paper is interesting because it might be used for learning better parts since the network units contain more information. The theory is then that instead of DCNN features we could perhaps try to extract parts from this type of network. We are unsure if this is actually possible and useful, but it could be worth investigating

Unsupervised learning of object semantic parts from internal states of cnns by population encoding Wang et al. 2015

The paper shows how to learn part detectors from DCNN features by clustering features on the fourth layer. Clustering is done by 'kmeans++' with K=number of filters in the layer, and then a greedy cluster merging algorithm to find a better K. To generate the features the network processes a number of images, and the features are then sampled. Relevant as it is one of the DCNN feature clustering papers cited by Kortylewski et al. 2020a.

Unsupervised learning of dictionaries of hierarchical compositional models Dai et al. 2014

Paper that learns hierarchical compositional models that are structured as a template that consists of a sub-group of part templates that shift locations and orientations relative to each other. Each part template is in turn composed of gabor wavelets that can also shift their locations and orientations relative to each other.

Uses a learning algorithm that iterates between two steps: (1) Template matching from bottom-up and template localization from top-down. (2) Dictionary re-learning by a shared matching process

Learning a hierarchical compositional shape vocabulary for multi-class object representation Fidler, Boben, and Leonardis 2014

Paper presents a framework on how to train a Compositional Deep Network through a cost function like a normal DNN. The model learns explicit structures through a hierarchical compositional shape vocabulary. The approach uses contour fragments and learns their frequent spatial positions. At a higher level these are combined into more complex and specific compositions.

1.0.6 Miscellaneous papers

Exploring Simple Siamese Representation Learning Chen and He 2021

This paper finds that siamese networks are capable of discovering meaningful representations without using methods that are commonly used (negative sampling pairs, large batches, momentum encoders) to hinder collapsing outputs (converging to constant output). Collapsing outputs can, according to the paper, be hindered by using a simple gradient-stop operation. I think overall that siamese networks shows some promise; this paper in particular is therefore relevant due to it showing how siamese networks can be simplified.

Differentiable Patch Selection for Image Recognition Cordonnier et al. 2021

Relates to removing redundant information from input data - Proposes a method for selecting the most relevant parts of inputs to reduce computation time. This is aimed towards high-resolution input images. Not directly relevant perhaps, but is useful for data driven methods in general.

Learning graph embeddings for compositional zero-shot learning Naeem et al. 2021

Relates to handling novel label/caption compositions - Proposes a method based on GNN, called Compostional Graph Embedding, which allows for generalization to unseen compositions. While this might not necessarily be directly relevant, I think some of the references might offer useful insights into applying GNNs for compositional modeling.

GIRAFFE: Representing Scenes as Compositional Generative Neural Feature Fields Niemeyer and Geiger 2021

This paper explores the usage of 3D information for controllable image synthesis. They propose a method called GIRAFFE, which utilizes a compositional 3D representation directly in the generative model. They find that using this 3D representation yields a more controllable and accurate image synthesis. Scenes are represented as something they call generative neural feature fields. I think it might be worth having a closer look at how they implemented the 3D representation as this might have some utility with respect to modeling the spatial relationship of parts.

Counterfactual generative networks Sauer and Geiger 2021

The paper introduces the problem of Neural Networks learning context like background information to cheat instead of learning object shape and texture. To remedy this they introduce a GAN which learns three specific things: (1) Background. (2) Object shape. (3) Object texture.

This could be relevant as Compositional Models have usually been extracting features from a DCNN backbone which has been trained with these weaknesses. The interesting observation could be to see what improvements could be made if we instead extracted features from a generator that has gone through GAN training with these tree categories.

Dense label encoding for boundary discontinuity free rotation detection Yang et al. 2021

The paper introduces a method to encode labels with rotation information. This seems to be very relevant for something that we can expand part detection with, because this type of encoded information would be useful for rotation between parts. It could also be used not only for rotation, but perhaps some sort of encoding schema could also help integrate other information we want to

encode between parts (spatial relationship for example?). A label encoded with rotation information would probably be useful anyway even if we cannot find other uses.

Open-vocabulary object detection using captions Zareian et al. 2021

The paper does zero shot object detection by learning from a word vocabulary It seems that they use a image caption dataset to learn a visual semantic space. This idea in itself is very interesting for part detection. What if we could exploit some type of text based model to find part relations and descriptions to learn better parts during training. For example, to find parts on a bus we could automatically find from a language space that a bus "consists" of "wheels, chassis, ..., etc". If we then again are able to look up further what describes a wheel "black colored, round shape, .., etc", we could then also have a learned meaning of these adjectives. If the model could learn what a round shape with black color would describe it would strongly react to the buss wheels, and we would therefore find much more meaningful parts than the current compositional approaches.

SR: The idea of combining learned text domain spaces with visual models is really interesting for unsupervised generation of more relevant and understandable parts

Zero-Shot Instance Segmentation Zheng et al. 2021

Relates to the problem of recognizing unseen object classes. The paper proposes a method for detecting and segmenting unseen class instances, named zero-shot instance segmentation. Uses word vectors and relate these to the unseen class to be able to segment in similar fashion if word has similar meaning. Being able to recognize and segment instances of unseen classes might have some utility within the learning phase of a compositional model.

Capsule Graph Neural Network Xinyi and Chen 2018

This paper proposes a model that merges aspects from both GNNs and Capsule Neural Networks. It uses capsules to more efficiently encode detailed node properties, which produces higher-quality graph embeddings. I find it somewhat difficult to evaluate this paper since I am familiar with neither GNNs or Capsule Neural Networks. Therefore, the classification of "Relevant" is mostly based on the assumption that since the two architectures are relevant individually, then a combination of them is probably also relevant.

1.0.7 Undecided relevance papers

Compositional Explanations for Image Classifiers Chockler, Kroening, and Sun 2021

The paper creates a method related to finding explanations related to finding why a object is classified as a certain class by using causality. This is done specifically for occluded images, but other than that the paper does not seem too relevant.

GANmut: Learning Interpretable Conditional Space for Gamut of Emotions d'Apolito et al. 2021

Relates to emotion modeling - This paper proposes a method for generating a continuous representation of emotions using basic categorical emotions (Angry, happy, fearful etc.) It might perhaps offer some insights with respect to using the latent space expressed by the encoder of a AE (or similar). The basic categorical emotions might instead be thought of as the constituent themes of a given category with the vector components being word encodings. Probably just nonsense, but who knows...

Part-aware Panoptic Segmentation Geus et al. 2021

This paper proposes a method that merges panoptic segmentation (segment objects and background classes) and part segmentation. This might be of some relevance, particularly with respect to learning how to differentiate between the parts of an object and the background.

DRANet - Disentangling Representation and Adaptation Networks for Unsupervised Cross-Domain Adaptation Lee, Cho, and Im 2021

Paper is about doing cross-adaptation through learned domain representations. Seems to be relevant as it could be used for "domain" adaptation between different type of part repesentations perhaps? F.eks if we could adapt what it means to be spatially related into one domain and separate out part specific stuff into another disentangled space. Overall seems to be relevant if we end up with domain learning.

Diverse Part Discovery: Occluded Person Re-Identification With Part-Aware Transformer Li et al. 2021b

This paper proposes a method for doing person re-identification of occluded persons using a part-aware transformer. The transformer is used for part discovery. Only identity labels are used for learning part prototypes, meaning no part labels are required. Part-aware masks are generated by calculating the similarity between pixels in the feature map and the part prototypes. The part-aware transformer might have some utility for us. Overall, I think this paper might be relevant, but should be considered low priority for now.

Pose Recognition with Cascade Transformers Li et al. 2021a

Relates to human 2D pose estimation - utilizes the encoder part of transformers to perform regression-based keypoint and person detection. The reason I label this as maybe relevant is that looking into 2D pose estimation perhaps might produce some useful insights into learning the spatial relationships between parts in objects. Looking into feature encoding in transformers might also be of some interest.

From Synthetic to Real: Unsupervised Domain Adaptation for Animal Pose Estimation Li and Lee 2021

Relates to animal 2D pose estimation - a method for reducing the domain gap between synthetic animal data and real animal data is proposed. I think the method for domain gap reduction might have some potential if we attempt to learn using synthetic data (Synthetic playground/safari idea).

Convolutional Hough Matching Networks Min and Cho 2021

This paper proposes a method for establishing visual correspondence between images. In other words, it matches similar object regions in two input images. This can perhaps be of some use with regards to learning a dictionary of codewords or similar. Should be considered low priority for now.

Unveiling the Potential of Structure Preserving for Weakly Supervised Object Localization Pan et al. 2021

Paper talks about method of doing weakly supervised object localization. Does not seem to be something we can specifically use for a compositional approach.

Closed-form factorization of latent semantics in gans Shen and Zhou 2021

Relates to unsupervised detection of semantically meaningful dimensions in the latent space of GANs. Can be useful if we decide to look into the posibility of using latent space to describe the relationship between parts and categories.

Using Shape to Categorize: Low-Shot Learning with an Explicit Shape Bias Stojanov, Thai, and Rehg 2021

Relates to learning mapping between 2D input to 3D object space - Learns a discriminative embedding space using 3D models. The embedding is used to learn how to map images into it. This might be relevant if we attempt to utilize synthetic data to learn a part dictionary.

NeMo: Neural Mesh Models of Contrastive Features for Robust 3D Pose Estimation Stojanov, Thai, and Rehg 2021

Paper that describes a method for 3D-pose estimation. Generates a mesh and which is rendered and compared to the feature map from a CNN to do 3D pose estimation.

Discrimination-Aware Mechanism for Fine-Grained Representation Learning Xu et al. 2021

Relates to increasing the capability of models for intra-class discrimination - Proposes a method for encouraging models to extract more discriminative visual cues. Most relevant if we limit the classification capability of the system to one group of objects (boats for instance).

Comparative Study Between Deep Learning and Bag of Visual Words for Wild-Animal Recognition Okafor et al. 2016

The paper tries Visual Bag of Words models against Deep Convolutional Neural networks and finds that they outperform Bag of Words. They also mention BoW models outperforming Autoencoders

(and restricted Boltzmann machines), however this should probably be looked into further due to the age of the paper. The paper seems to be interesting although a bit outdated.

1.0.8 Not relevant papers

Bottom-Up Human Pose Estimation Via Disentangled Keypoint Regression Geng et al. 2021

Proposes a regression-based method for finding keypoints that is more accurate than previous regression-based methods. Overall I think the proposed method is not really applicable to our case.

Deep learning for visual understanding: A review Guo et al. 2016

The paper is general and goes through many different Deep Learning methodologies for Computer Vision. Not very specific for the situation.

SimPLE: Similar Pseudo Label Exploitation for Semi-Supervised Classification Hu et al. 2021

This paper proposes a method for semi-supervised learning. The model is first trained on a labeled dataset using weak augmentations. The model then does self-supervision by calculating the loss of prediction between heavily augmented data and weakly augmented data (the same unlabeled data), where the latter functions as the ground-truth. While interesting, I do not think it is very relevant for a compositional approach.

Weakly Supervised Instance Segmentation for Videos with Temporal Mask Consistency Liu et al. 2021

The paper talks about improving weakly supervised instance segmentation by moving the training from images to videos. I am going to say that this is not directly relevant as it is just a paper that discusses how to improve by better training, and not anything directly related to compositional models or part detectors.

Open World Compositional Zero-Shot Learning Mancini et al. 2021

The paper discusses compositional zero-shot learning (CZSL) and proposes a method that outperforms the previous SOTA method in an open world setting. The goal of CZSL "[...] is to learn a set of states and object while generalizing to unseen compositions". An example of such a state-object pair is for instance "old dog". To handle novel cases, the proposed method has a shared embedding space of the images and compositions. To discard implausible compositions a feasibility score is calculated using similarities among primitives. In other words, it will discard compositions like "ripe dog" from an embedding space consisting of dog, tomato and their compositions. This paper is probably not relevant as I do not think the approach to handle unseen compositions is really applicable to shape compositions.

SOLD: Self-supervised Occlusion-aware Line Description and Detection Pautrat et al. 2021

Proposes a method for line detection and description in images using deep learning. Mostly relevant for scene related tasks.

Bibliography

Chen, Xinlei and Kaiming He (2021). "Exploring simple siamese representation learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15750–15758.

Chockler, Hana, Daniel Kroening, and Youcheng Sun (2021). "Compositional Explanations for Image Classifiers". In: *arXiv preprint arXiv:2103.03622*.

Cordonnier, Jean-Baptiste et al. (2021). "Differentiable Patch Selection for Image Recognition". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2351–2360.

Cosgrove, Christian et al. (2020). *Robustness Out of the Box: Compositional Representations Naturally Defend Against Black-Box Patch Attacks*. arXiv: 2012.00558 [cs.CV].

Dai, Jifeng et al. (2014). "Unsupervised learning of dictionaries of hierarchical compositional models". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2505–2512.

d'Apolito, Stefano et al. (2021). "GANmut: Learning Interpretable Conditional Space for Gamut of Emotions". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 568–577.

Fei-Fei, Li and Pietro Perona (2005). "A bayesian hierarchical model for learning natural scene categories". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 2. IEEE, pp. 524–531.

Fidler, Sanja, Marko Boben, and Ales Leonardis (2014). "Learning a hierarchical compositional shape vocabulary for multi-class object representation". In: *arXiv preprint arXiv:1408.5516*.

Geng, Zigang et al. (2021). "Bottom-Up Human Pose Estimation via Disentangled Keypoint Regression". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14676–14686.

Geus, Daan de et al. (2021). "Part-aware Panoptic Segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5485–5494.

Guo, Mengqi et al. (2020). *Unsupervised Part Discovery via Feature Alignment*. arXiv: 2012.00313 [cs.CV].

Guo, Yanming et al. (2016). "Deep learning for visual understanding: A review". In: *Neurocomputing* 187, pp. 27–48.

He, Ju, Adam Kortylewski, and Alan Yuille (2021). "COMPAS: Representation Learning with Compositional Part Sharing for Few-Shot Classification". In: *arXiv preprint arXiv:2101.11878*.

Hu, Zijian et al. (2021). "SimPLE: Similar Pseudo Label Exploitation for Semi-Supervised Classification". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15099–15108.

Kortylewski, A et al. (2020a). "Combining compositional models and deep networks for robust object classification under occlusion (2020)". In: *arXiv preprint arxiv:1905.11826*.

Kortylewski, Adam et al. (2019). "Greedy structure learning of hierarchical compositional models". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11612–11621.

Kortylewski, Adam et al. (2020b). "Compositional convolutional neural networks: A deep architecture with innate robustness to partial occlusion". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8940–8949.

Kortylewski, Adam et al. (2021a). "Compositional convolutional neural networks: A robust and interpretable model for object recognition under occlusion". In: *International Journal of Computer Vision* 129.3, pp. 736–760.

Kortylewski, Adam et al. (2021b). "Compositional Generative Networks and Robustness to Perceptible Image Changes". In: *2021 55th Annual Conference on Information Sciences and Systems (CISS)*. IEEE, pp. 1–8.

Lazebnik, Svetlana, Cordelia Schmid, and Jean Ponce (2006). "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories". In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 2. IEEE, pp. 2169–2178.

Lee, Seunghun, Sunghyun Cho, and Sunghoon Im (2021). "DRANet: Disentangling Representation and Adaptation Networks for Unsupervised Cross-Domain Adaptation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15252–15261.

Li, Chen and Gim Hee Lee (2021). "From Synthetic to Real: Unsupervised Domain Adaptation for Animal Pose Estimation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1482–1491.

Li, Ke et al. (2021a). "Pose Recognition with Cascade Transformers". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1944–1953.

Li, Yulin et al. (2021b). "Diverse Part Discovery: Occluded Person Re-Identification With Part-Aware Transformer". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2898–2907.

Liao, Renjie et al. (2016). "Learning deep parsimonious representations". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*, pp. 5083–5091.

Liu, Qing et al. (2021). "Weakly Supervised Instance Segmentation for Videos with Temporal Mask Consistency". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13968–13978.

Mancini, Massimiliano et al. (2021). "Open World Compositional Zero-Shot Learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5222–5230.

Min, Juhong and Minsu Cho (2021). "Convolutional Hough Matching Networks". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2940–2950.

Mumuni, Alhassan and Fuseini Mumuni (2021). "CNN Architectures for Geometric Transformation-Invariant Feature Representation in Computer Vision: A Review". In: *SN Computer Science* 2.5, pp. 1–23.

Naeem, Muhammad Ferjad et al. (2021). "Learning graph embeddings for compositional zero-shot learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 953–962.

Niemeyer, Michael and Andreas Geiger (2021). "Giraffe: Representing scenes as compositional generative neural feature fields". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11453–11464.

Okafor, Emmanuel et al. (2016). "Comparative study between deep learning and bag of visual words for wild-animal recognition". In: *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, pp. 1–8.

Ommer, Björn and Joachim M Buhmann (2005). "Object categorization by compositional graphical models". In: *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*. Springer, pp. 235–250.

Ommer, Bjorn and Joachim M Buhmann (2007). "Learning the compositional nature of visual objects". In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1–8.

Pan, Xingjia et al. (2021). "Unveiling the Potential of Structure Preserving for Weakly Supervised Object Localization". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11642–11651.

Pautrat, Rémi et al. (2021). "SOLD2: Self-Supervised Occlusion-Aware Line Description and Detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11368–11378.

Ruis, FA (2021). "Independent prototype propagation graph for compositional zero-shot recognition". MA thesis. University of Twente.

Saleh, Kaziwa, Sándor Szénási, and Zoltán Vámossy (2021). "Occlusion Handling in Generic Object Detection: A Review". In: *2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*. IEEE, pp. 000477–000484.

Sánchez, Jorge et al. (2013). "Image classification with the fisher vector: Theory and practice". In: *International journal of computer vision* 105.3, pp. 222–245.

Sauer, Axel and Andreas Geiger (2021). "Counterfactual generative networks". In: *arXiv preprint arXiv:2101.06046*.

Shen, Yujun and Bolei Zhou (2021). "Closed-form factorization of latent semantics in gans". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1532–1540.

Sivic, Josef and Andrew Zisserman (2003). "Video Google: A text retrieval approach to object matching in videos". In: *Computer Vision, IEEE International Conference on*. Vol. 3. IEEE Computer Society, pp. 1470–1470.

Stojanov, Stefan, Anh Thai, and James M Rehg (2021). "Using Shape to Categorize: Low-Shot Learning with an Explicit Shape Bias". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1798–1808.

Sun, Yihong, Adam Kortylewski, and Alan Yuille (2021). *Weakly-Supervised Amodal Instance Segmentation with Compositional Priors*. arXiv: `2010.13175 [cs.CV]`.

Tabernik, Domen et al. (2016). "Towards deep compositional networks". In: *2016 23rd international conference on pattern recognition (ICPR)*. IEEE, pp. 3470–3475.

Tritrong, Nontawat, Pitchaporn Rewatbowornwong, and Supasorn Suwajanakorn (2021). "Repurposing gans for one-shot semantic part segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4475–4485.

Wang, Angtian et al. (2020). "Robust object detection under occlusion with context-aware compositionalnets". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12645–12654.

Wang, Jianyu et al. (2015). "Unsupervised learning of object semantic parts from internal states of cnns by population encoding". In: *arXiv preprint arXiv:1511.06855*.

Wang, Jianyu et al. (2017). "Detecting semantic parts on partially occluded objects". In: *arXiv preprint arXiv:1707.07819*.

Xinyi, Zhang and Lihui Chen (2018). "Capsule graph neural network". In: *International conference on learning representations*.

Xu, Furong et al. (2021). "Discrimination-Aware Mechanism for Fine-Grained Representation Learning". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 813–822.

Yang, Xue et al. (2021). "Dense label encoding for boundary discontinuity free rotation detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15819–15829.

Yarlagadda, Pradeep, Antonio Monroy, and Björn Ommer (2010). "Voting by grouping dependent parts". In: *European Conference on Computer Vision*. Springer, pp. 197–210.

Yuan, Xiaoding et al. (2021). "Robust Instance Segmentation through Reasoning about Multi-Object Occlusion". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11141–11150.

Yuille, Alan L and Chenxi Liu (2021). "Deep nets: What have they ever done for vision?" In: *International Journal of Computer Vision* 129.3, pp. 781–802.

Zareian, Alireza et al. (2021). "Open-vocabulary object detection using captions". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14393–14402.

Zheng, Ye et al. (2021). "Zero-Shot Instance Segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2593–2602.

B.2 Re-Investigating Part Detectors

Literature Collection Report: Further investigation of part detectors

Stig Andre Rosenlund
William Emanuel Skreien Kaaby

July 2, 2022

1 Bibliography

1.1 Applicable papers - High

Title: "Interpretable convolutional neural networks"
Authors: Zhang, Wu, and Zhu (2018)
Citations: 462

Abstract:
This paper proposes a method to modify a traditional convolutional neural network (CNN) into an interpretable CNN, in order to clarify knowledge representations in high conv-layers of the CNN. In an interpretable CNN, each filter in a high conv-layer represents a specific object part. Our interpretable CNNs use the same training data as ordinary CNNs without a need for any annotations of object parts or textures for supervision. The interpretable CNN automatically assigns each filter in a high conv-layer with an object part during the learning process. We can apply our method to different types of CNNs with various structures. The explicit knowledge representation in an interpretable CNN can help people understand the logic inside a CNN, i.e. what patterns are memorized by the CNN for prediction. Experiments have shown that filters in an interpretable CNN are more semantically meaningful than those in a traditional CNN. The code is available at `https://github.com/zqs1022/interpretableCNN`.

Comment:
This paper improves the interpretability of CNNs by making the filters in feature maps at the higher convolution layers more semantically meaningful. This is interesting because it has the effect of restricting a filter to only activate for specific visual queues instead of multiple as we have experienced in the pre-project. Figure **??** shows an example of the difference between interpretable and non-interpretable feature maps.
This can be useful for a part detector approach based on DCNN features. The current understanding is that this should either work on its own, where the filters in a CNN would start behaving as part-detectors themselves. It could also be that this approach would work well as a combination with the Yuille approach, since we would have more well defined feature vectors when we cluster.
A noteworthy discovery in this paper is their use of receptive fields. Their receptive fields are not theoretical as the ones we used in the Yuille approach, but more like how Rudolf described them in an earlier meeting. They specifically cite the paper Zhou et al. 2014 for how they extract these receptive fields.
The method also seems to be simple to implement, as the only change is a loss function introduced in end-to-end training.

Title: "Interpretable CNNs for object classification"
Authors: Zhang et al. (2020)
Citations: 10

Abstract:
This paper proposes a generic method to learn interpretable convolutional filters in a deep convolutional neural network (CNN) for object classification, where each interpretable filter encodes features of a specific object part. Our method does not require additional annotations of object parts or textures for supervision. Instead, we use the same training data as traditional CNNs. Our method automatically assigns each interpretable filter in a high conv-layer with an object part of a certain category during the learning process. Such explicit knowledge representations in conv-layers of the CNN help people clarify the logic encoded in the CNN, i.e., answering what patterns the CNN extracts from an input image and uses for prediction. We have tested our method using different benchmark CNNs with various architectures to demonstrate the broad applicability of our method. Experiments have shown that our interpretable filters are much more semantically meaningful than traditional filters.

Comment:
This is the same paper as (Zhang, Wu, and Zhu 2018), but it is a newer version and has what seems to be some small updates and extras.

Title: "The application of two-level attention models in deep convolutional neural network for fine-grained image classification"
Authors: Xiao et al. (2015)
Citations: 716

Abstract:
Fine-grained classification is challenging because categories can only be discriminated by subtle and local differences. Variances in the pose, scale or rotation usually make the problem more difficult. Most fine-grained classification systems follow the pipeline of finding foreground object or object parts (where) to extract discriminative features (what). In this paper, we propose to apply visual attention to finegrained classification task using deep neural network. Our pipeline integrates three types of attention: the bottom-up attention that propose candidate patches, the object-level top-down attention that selects relevant patches to a certain object, and the part-level top-down attention that localizes discriminative parts. We combine these attentions to train domain-specific deep nets, then use it to improve both the what and where aspects. Importantly, we avoid using expensive annotations like bounding box or part information from end-to-end. The weak supervision constraint makes our work easier to generalize. We have verified the effectiveness of the method on the subsets of ILSVRC2012 dataset and CUB200 2011 dataset. Our pipeline delivered significant improvements and achieved the best accuracy under the weakest supervision condition. The performance is competitive against other methods that rely on additional annotations.

Comment:
This paper also extracts part detectors from Neural Network internals like the Yuille approach, but they have some difference in the methodology. The interesting take-aways would be to investigate how their approach compares to Yuille's approach. A brief description of their approach is that they use spectral clustering on a similarity matrix which is made from cosine similarity. This partitions whole filters into different groups, which again differs from Yuille where they chose to cluster over feature vectors and not whole filters.
It is also interesting how the paper has a high recall bottom up approach for first detecting objects, and then refine them afterwards with a top down approach.

Title: "Object-part attention model for fine-grained image classification"
Authors: Peng, He, and Zhao (2017)
Citations: 236

Abstract:
Fine-grained image classification is to recognize hundreds of subcategories belonging to the same basic-level category, such as 200 subcategories belonging to the bird, which is highly challenging due to large variance in the same subcategory and small variance among different subcategories. Existing methods generally first locate the objects or parts and then discriminate which subcategory the image belongs to. However, they mainly have two limitations: 1) relying on object or part annotations which are heavily labor consuming; and 2) ignoring the spatial relationships between the object and its parts as well as among these parts, both of which are significantly helpful for finding discriminative parts. Therefore, this paper proposes the object-part attention model (OPAM) for weakly supervised finegrained image classification and the main novelties are: 1) objectpart attention model integrates two level attentions: object-level attention localizes objects of images, and part-level attention selects discriminative parts of object. Both are jointly employed to learn multi-view and multi-scale features to enhance their mutual promotion; and 2) Object-part spatial constraint model combines two spatial constraints: object spatial constraint ensures selected parts highly representative and part spatial constraint eliminates redundancy and enhances discrimination of selected parts. Both are jointly employed to exploit the subtle and local differences for distinguishing the subcategories. Importantly, neither object nor part annotations are used in our proposed approach, which avoids the heavy labor consumption of labeling. Compared with more than ten state-of-the-art methods on four widely-used datasets, our OPAM approach achieves the best performance.

Comment:
This paper describes a two-level attention approach for doing fine-grained image classification. Of particular interest is their approach to modeling the spatial relationship between parts and their parent object in an explicit manner. This approach does not require object or part annotations, instead extracting saliency maps (CAM) to detect objects via the object-level attention model. Parts are extracted from the saliency maps using two different spatial constraints. Like Yuille, they perform clustering on CNN layers to group parts. However, they use spectral clustering instead of K-Means++. Another difference is that they select parts before performing clustering of the CNN; the purpose of clustering is to align the selected parts. It might be valuable to conduct a more in-depth comparison of the merits of each approach.

Title: "Extraction of an Explanatory Graph to Interpret a CNN"
Authors: Zhang et al. (2021)
Citations: 9

Abstract:
This paper introduces an explanatory graph representation to reveal object parts encoded inside convolutional layers of a CNN. Given a pre-trained CNN, each filter1 in a conv-layer usually represents a mixture of object parts. We develop a simple yet effective method to learn an explanatory graph, which automatically disentangles object parts from each filter without any part annotations. Specifically, given the feature map of a filter, we mine neural activations from the feature map, which correspond to different object parts. The explanatory graph is constructed to organize each mined part as a graph node. Each edge connects two nodes, whose corresponding object parts usually co-activate and keep a stable spatial relationship. Experiments show that each graph node consistently represented the same object part through different images, which boosted the transferability of CNN features. The explanatory graph transferred features of object parts to the task of part localization, and our method significantly outperformed other approaches.

Comment:
This paper shows how to construct an explanation graph from disentangled filters in a CNN. This is interesting because it mimicks some of the ideas we came up with in the pre-project of modeling part-dependence between layers. Figure **??** shows how the explainable graph is built up from the different convolutional layers, and how they at higher levels represent more high level concepts, while at lower levels represent simpler concepts. We could take inspiration from this paper when creating a joint structure detector.
An idea can be to take inspiration and combine this paper with (Xiao et al. 2015). By taking advantage of the top-down and bottom-up approach in (Xiao et al. 2015). We could first scan an image with a high-recall CNN and refine detections afterwards with a joint structure detector/explanation graph. The strength of this approach is that we take advantage of a normal CNN's ability to detect, and then a joint structure detectors ability to avoid false positives (This might not be unique though, and could be a bit too similar to some of the other papers in this list).

Title: "Part detector discovery in deep convolutional neural networks"
Authors: Simon, Rodner, and Denzler (2014)
Citations: 62

Abstract:
Current fine-grained classification approaches often rely on a robust localization of object parts to extract localized feature representations suitable for discrimination. However, part localization is a challenging task due to the large variation of appearance and pose. In this paper, we show how pre-trained convolutional neural networks can be used for robust and efficient object part discovery and localization without the necessity to actually train the network on the current dataset. Our approach called "part detector discovery" (PDD) is based on analyzing the gradient maps of the network outputs and finding activation centers spatially related to annotated semantic parts or bounding boxes. This allows us not just to obtain excellent performance on the CUB200-2011 dataset, but in contrast to previous approaches also to perform detection and bird classification jointly without requiring a given bounding box annotation during testing and ground-truth parts during training. The code is available at `http://www.inf-cv.uni-jena.de/part_discovery` and `https://github.com/cvjena/PartDetectorDisovery`

Comment:
This paper aims to use gradient maps to discover parts. It represents an early attempt at using CNNs to detect parts. They also develop a method for part-based classification using a SVM. While we think this paper is somewhat outdated, it might be a useful resource for learning how one might use a gradient map instead of a saliency map.

Title: "Progressive Co-Attention Network for Fine-grained Visual Classification"
Authors: Zhang et al. (2021)
Citations: 2

Abstract:
Fine-grained visual classification aims to recognize images belonging to multiple sub-categories within a same category. It is a challenging task due to the inherently subtle variations among highly-confused categories. Most existing methods only take an individual image as input, which may limit the ability of models to recognize contrastive clues from different images. In this paper, we propose an effective method called progressive co-attention network (PCA-Net) to tackle this problem. Specifically, we calculate the channel-wise similarity by encouraging interaction between the feature channels within same-category image pairs to capture the common discriminative features. Considering that complementary information is also crucial for recognition, we erase the prominent areas enhanced by the channel interaction to force the network to focus on other discriminative regions. The proposed model has achieved competitive results on three fine-grained visual classification benchmark datasets: CUB-200-2011, Stanford Cars, and FGVC Aircraft.

Comment:
This paper aims to classify objects by modeling the interaction between the channels of two models when given two different images of the same sub-category. This approach leads to discovery of features of greater discriminative value. These features do seem to correspond to parts, but it is used in a more implicit manner compared to some other papers. While a novel approach, there might be some concerns related to interpretability. The data is fully encoded in the layers of the CNNs and the resulting features are sent to a fully connected layer. This paper can be quite useful, but it depends on the requirements of the final solution.

Title: "Unsupervised part mining for fine-grained image classification"
Authors: Zhang et al. (2019)
Citations: 10

Abstract:
Fine-grained image classification remains challenging due to the large intra-class variance and small inter-class variance. Since the subtle visual differences are only in local regions of discriminative parts among sub-categories, part localization is a key issue for fine-grained image classification. Most existing approaches localize object or parts in an image with object or part annotations, which are expensive and labor-consuming. To tackle this issue, we propose a fully unsupervised part mining (UPM) approach to localize the discriminative parts without even image-level annotations, which largely improves the fine-grained classification performance. We first utilize pattern mining techniques to discover frequent patterns, i.e., co-occurrence highlighted regions, in the feature maps extracted from a pre-trained convolutional neural network (CNN) model. Inspired by the fact that these relevant meaningful patterns typically hold appearance and spatial consistency, we then cluster the mined regions to obtain the cluster centers and the discriminative parts surrounding the cluster centers are generated. Importantly, any annotations and sophisticated training procedures are not used in our proposed part localization approach. Finally, a multi-stream classification network is built for aggregating the original, object-level and part-level features simultaneously. Compared with other state-of-the-art approaches, our UPM approach achieves the competitive performance.

Comment:
This paper aims to apply pattern mining on a CNN for the purposes of discovering discriminative parts. The area greatest interest in this paper is the steps performed before part clustering. In contrast to Yuille, they do not directly cluster feature vectors, instead performing pattern mining operations on feature maps. After the maps have been mined, the extracted information is merged into a support map. The support map is then clustered using K-Means. We should investigate the mining process more thoroughly. This method is fully unsupervised.

Title: "Fine-grained visual-textual representation learning"
Authors: He and Peng (2019)
Citations: 19

Abstract:
Fine-grained visual categorization is to recognize hundreds of subcategories belonging to the same basic-level category, which is a highly challenging task due to the quite subtle and local visual distinctions among similar subcategories. Most existing methods generally learn part detectors to discover discriminative regions for better categorization performance. However, not all parts are beneficial and indispensable for visual categorization, and the setting of part detector number heavily relies on prior knowledge as well as experimental validation. As is known to all, when we describe the object of an image via textual descriptions, we mainly focus on the pivotal characteristics and rarely pay attention to common characteristics as well as the background areas. This is an involuntary transfer from human visual attention to textual attention, which leads to the fact that textual attention tells us how many and which parts are discriminative and significant to categorization. So, textual attention could help us to discover visual attention in the image. Inspired by this, we propose a fine-grained visual-textual representation learning (VTRL) approach, and its main contributions are: 1) fine-grained visual-textual pattern mining devotes to discovering discriminative visual-textual pairwise information for boosting categorization performance through jointly modeling vision and text with generative adversarial networks, which automatically and adaptively discovers discriminative parts and 2) VTRL jointly combines visual and textual information, which preserves the intra-modality and inter-modality information to generate complementary fine-grained representation, as well as further improves categorization performance. Comprehensive experimental results on the widely used CUB-200-2011 and Oxford Flowers-102 datasets demonstrate the effectiveness of our VTRL approach, which achieves the best categorization accuracy compared with the state-of-the-art methods.

Comment:
This paper aims to leverage the information of natural language descriptions to focus on the most discriminative parts of an object. Our experience with automatic part discovery from the pre-project is that a method for pruning parts of low discriminative value is needed to avoid large dictionaries of parts that are difficult to interpret. Applying textual information represents an intriguing approach to achieve this. Furthermore, the method does not require object nor part annotations, which is desirable.

1.2 Applicable papers - Medium

Title: "Progressive learning for weakly supervised fine-grained classification"
Authors: Yan et al. (2020)
Citations: 3

Abstract:
Despite fine-grained image classification has made considerable progress, it still remains a challenging task due to the difficulty of finding subtle distinctions. Most existing methods solve this problem by selecting the top-N highest scores' discriminative patches from candidate patches at one time. However, since the classification network often highlights small and sparse regions, the selected patches with the lower rank may contain noise information. To address this problem and ensure the diversity of fine-grained features, we propose a progressive patch localization module (PPL) to find the discriminative patches more accurately. Specifically, this work employs the classification model to find first most discriminative patch, then removes the most salient region to help the localization of the next most discriminative patch, and the top-K discriminative patches can be found by repeating this procedure. In addition, in order to further improve the representational power of patch-level features, we propose a feature calibration module (FCM). This module employs the global information to selectively emphasize discriminative features and suppress useless information, which can obtain more robust and discriminative local feature representations and then help classification network achieve better performance. Extensive experiments are conducted to show the substantial improvements of our method on three benchmark datasets.

Comment:
This paper proposes a method that progressively finds the most discriminative image patches in an input image. They call this step progressive patch localization, and in essence represents a method for finding K patches which are most discriminative. This should ensure that the issue of large and uninterpretable dictionaries which we encountered in the pre-project will not occur. The second step, feature calibration, represents a refinement of the features found in the previous step.

Title: "Training Interpretable Convolutional Neural Networks by Differentiating Class-specific Filters"
Authors: Liang et al. (2020)
Citations: 8

Abstract:
Convolutional neural networks (CNNs) have been successfully used in a range of tasks. However, CNNs are often viewed as "blackbox" and lack of interpretability. One main reason is due to the filter-class entanglement – an intricate many-to-many correspondence between filters and classes. Most existing works attempt post-hoc interpretation on a pre-trained model, while neglecting to reduce the entanglement underlying the model. In contrast, we focus on alleviating filter-class entanglement during training. Inspired by cellular differentiation, we propose a novel strategy to train interpretable CNNs by encouraging class-specific filters, among which each filter responds to only one (or few) class. Concretely, we design a learnable sparse Class-Specific Gate (CSG) structure to assign each filter with one (or few) class in a flexible way. The gate allows a filter's activation to pass only when the input samples come from the specific class. Extensive experiments demonstrate the fabulous performance of our method in generating a sparse and highly classrelated representation of the input, which leads to stronger interpretability. Moreover, comparing with the standard training strategy, our model displays benefits in applications like object localization and adversarial sample detection. Code link: `https://github.com/hyliang96/CSGCNN`.

Comment:
This paper shares similarities with (Zhang, Wu, and Zhu 2018) by modifying the internal filters of a neural network to be specific and not many-to-many detectors. The difference is that this paper focuses on class specific filters instead of part specific filters however, and might therefore be less relevant. It could be that we can combine the approaches and create class specific and part specific filters together somehow, but this is just speculation without further investigation.

Title: "Objects as context for detecting their semantic parts"
Authors: Gonzalez-Garcia, Modolo, and Ferrari (2018)
Citations: 9

Abstract:
We present a semantic part detection approach that effectively leverages object information. We use the object appearance and its class as indicators of what parts to expect. We also model the expected relative location of parts inside the objects based on their appearance. We achieve this with a new network module, called OffsetNet, that efficiently predicts a variable number of part locations within a given object. Our model incorporates all these cues to detect parts in the context of their objects. This leads to considerably higher performance for the challenging task of part detection compared to using part appearance alone (+5 mAP on the PASCAL-Part dataset). We also compare to other part detection methods on both PASCAL-Part and CUB200-2011 datasets.

Comment:
The paper uses object as context to model underlying parts. They seem to share how they model parts by expected location with Yuille, but lack the overall explainability aspect of the other papers since they only use neural networks. The paper is therefore less attractive to investigate further.

Title: "Adversarial erasing attention for fine-grained image classification"
Authors: Ji et al. (2021)
Citations: 2

Abstract:
Recognizing fine-grained subcategories is a challenging task due to the large intra-class diversities and small inter-class variances of the fine-grained images. The common thought is to find out the parts that can distinguish similar subcategories efficiently. Most previous works rely on the manual annotations or attention technologies to localize the discriminative parts and have achieved great progress. However, these manual annotations are demanding in practical applications and some complicated constrains on the loss functions have to be adopted to localize the discriminative parts for building multi-view feature representations. To handle the challenges above, the strategy of adversarial erasing is applied on the attention module in this paper, which learns to localize different discriminative parts by erasing the most one from the image. Without the complicated loss functions, the proposed attention module can localize the discriminative parts more efficiently. Different from many part based methods, the classification network which consists of three subnetworks is introduced, and the subnetworks are trained by the original image and two discriminative parts respectively. Moreover, features learned from the three subnetworks are then fused in a more efficiently way to build better feature representations. Four mostly used datasets of CUB-200-2011, Stanford Dogs, Stanford Cars and FGVC-Aircraft are utilized to evaluate the proposed method and experimental results show that it can outperform some state-of-the-art methods without using the manual annotations.

Comment:
This paper proposes a method that forces a network to learn the most discriminative regions by progressively erasing the most discriminative region. Overall, this method appears relatively similar to (Yan et al. 2020).

1.3 Applicable papers - Low

Title: "Learning semantic part-based models from google images"
Authors: Modolo and Ferrari (2017)
Citations: 12

Abstract:
We propose a technique to train semantic part-based models of object classes from Google Images. Our models encompass the appearance of parts and their spatial arrangement on the object, specific to each viewpoint. We learn these rich models by collecting training instances for both parts and objects, and automatically connecting the two levels. Our framework works incrementally, by learning from easy examples first, and then gradually adapting to harder ones. A key benefit of this approach is that it requires no manual part location annotations. We evaluate our models on the challenging PASCAL-Part dataset [1] and show how their performance increases at every step of the learning, with the final models more than doubling the performance of directly training from images retrieved by querying for part names (from 12.9 to 27.2 AP). Moreover, we show that our part models can help object detection performance by enriching the R-CNN detector with parts.

Comment:
The paper proposes to use google images to learn different viewpoints when creating part based models. Overall the paper does not seem too interesting over the fact that they collect images from the Internet.

1.4 Ancilliary papers

Title: "Object detectors emerge in deep scene cnns"
Authors: Zhou et al. (2014)
Citations: 1096

Abstract:
With the success of new computational architectures for visual processing, such as convolutional neural networks (CNN) and access to image databases with millions of labeled examples (e.g., ImageNet, Places), the state of the art in computer vision is advancing rapidly. One important factor for continued progress is to understand the representations that are learned by the inner layers of these deep architectures. Here we show that object detectors emerge from training CNNs to perform scene classification. As scenes are composed of objects, the CNN for scene classification automatically discovers meaningful objects detectors, representative of the learned scene categories. With object detectors emerging as a result of learning to recognize scenes, our work demonstrates that the same network can perform both scene recognition and object localization in a single forward-pass, without ever having been explicitly taught the notion of objects.

Comment:
This paper is interesting in how they find receptive fields, and it is likely very relevant for whatever approach we end up using. There could also be different receptive field approaches we could pursue.

Title: "An attention-driven hierarchical multi-scale representation for visual recognition"
Authors: Wharton, Behera, and Bera (2021)
Citations: 0

Abstract:
Convolutional Neural Networks (CNNs) have revolutionized the understanding of visual content. This is mainly due to their ability to break down an image into smaller pieces, extract multi-scale localized features and compose them to construct highly expressive representations for decision making. However, the convolution operation is unable to capture long-range dependencies such as arbitrary relations between pixels since it operates on a fixed-size window. Therefore, it may not be suitable for discriminating subtle changes (e.g. fine-grained visual recognition). To this end, our proposed method captures the high-level long-range dependencies by exploring Graph Convolutional Networks (GCNs), which aggregate information by establishing relationships among multiscale hierarchical regions. These regions consist of smaller (closer look) to larger (far look), and the dependency between regions is modeled by an innovative attention-driven message propagation, guided by the graph structure to emphasize the neighborhoods of a given region. Our approach is simple yet extremely effective in solving both the finegrained and generic visual classification problems. It outperforms the state-of-the-arts with a significant margin on three and is very competitive on other two datasets.

Comment:
... No comment ...

Title: "CGPart: A Part Segmentation Dataset Based on 3D Computer Graphics Models"
Authors: Liu et al. (2021)
Citations: 1

Abstract:
Convolutional Neural Networks (CNNs) have revolutionized the understanding of visual content. This is mainly due to their ability to break down an image into smaller pieces, extract multi-scale localized features and compose them to construct highly expressive representations for decision making. However, the convolution operation is unable to capture long-range dependencies such as arbitrary relations between pixels since it operates on a fixed-size window. Therefore, it may not be suitable for discriminating subtle changes (e.g. fine-grained visual recognition). To this end, our proposed method captures the high-level long-range dependencies by exploring Graph Convolutional Networks (GCNs), which aggregate information by establishing relationships among multiscale hierarchical regions. These regions consist of smaller (closer look) to larger (far look), and the dependency between regions is modeled by an innovative attention-driven message propagation, guided by the graph structure to emphasize the neighborhoods of a given region. Our approach is simple yet extremely effective in solving both the finegrained and generic visual classification problems. It outperforms the state-of-the-arts with a significant margin on three and is very competitive on other two datasets.

Comment:
... No comment ...

1.5 Surveys and Reviews

Title: "Visual interpretability for deep learning: a survey"
Authors: Zhang and Zhu (2018)
Citations: 521

Abstract:
This paper reviews recent studies in understanding neural-network representations and learning neural networks with interpretable/disentangled middlelayer representations. Although deep neural networks have exhibited superior performance in various tasks, the interpretability is always the Achilles' heel of deep neural networks. At present, deep neural networks obtain high discrimination power at the cost of low interpretability of their black-box representations. We believe that high model interpretability may help people to break several bottlenecks of deep learning, e.g. learning from very few annotations, learning via human-computer communications at the semantic level, and semantically debugging network representations. We focus on convolutional neural networks (CNNs), and we revisit the visualization of CNN representations, methods of diagnosing representations of pre-trained CNNs, approaches for disentangling pre-trained CNN representations, learning of CNNs with disentangled representations, and middle-to-end learning based on model interpretability. Finally, we discuss prospective trends in explainable artificial intelligence.

Comment:
... No comment ...

Title: "A survey on neural network interpretability"
Authors: Zhang et al. (2021)
Citations: 41

Abstract:
Along with the great success of deep neural networks, there is also growing concern about their black-box nature. The interpretability issue affects people's trust on deep learning systems. It is also related to many ethical problems, e.g., algorithmic discrimination. Moreover, interpretability is a desired property for deep networks to become powerful tools in other research fields, e.g., drug discovery and genomics. In this survey, we conduct a comprehensive review of the neural network interpretability research. We first clarify the definition of interpretability as it has been used in many different contexts. Then we elaborate on the importance of interpretability and propose a novel taxonomy organized along three dimensions: type of engagement (passive vs. active interpretation approaches), the type of explanation, and the focus (from local to global interpretability). This taxonomy provides a meaningful 3D view of distribution of papers from the relevant literature as two of the dimensions are not simply categorical but allow ordinal subcategories. Finally, we summarize the existing interpretability evaluation methods and suggest possible research directions inspired by our new taxonomy.

Comment:
... No comment ...

Title: "What do CNN neurons learn: Visualization & Clustering"
Authors: Dai (2020)
Citations: 1

Abstract:
In recent years convolutional neural networks (CNN) have shown striking progress in various tasks. However, despite the high performance, the training and prediction process remains to be a black box, leaving it a mystery to extract what neurons learn in CNN. In this paper, we address the problem of interpreting a CNN from the aspects of the input image's focus and preference, and the neurons' domination, activation and contribution to a concrete final prediction. Specifically, we use two techniques – visualization and clustering – to tackle the problems above. Visualization means the method of gradient descent on image pixel, and in clustering section two algorithms are proposed to cluster respectively over image categories and network neurons. Experiments and quantitative analyses have demonstrated the effectiveness of the two methods in explaining the question: what do neurons learn.

Comment:
... No comment ...

2 Conclusions

- Comment: Part detectors from DCNN seem to be more common than initially thought.
- Implement the interpretable filters from (Zhang, Wu, and Zhu 2018). Investigate how filters act as part detectors and how the feature vectors would cluster with the Yuille approach. Also, check if the class-specific filters can be as valuable as the part-specific filters.
- Take inspiration from bottom-up (high recall) and then refine with part-detectors to remove false positives.
- Take inspiration from retrieving parts by spatial constraints and saliency maps.
- Take inspiration from the explanation graph and attempt to model the dependencies between the convolution layers. It could also be interesting if combined with the discriminative filter paper (Would this be similar to the original contour papers we read but more flexible because of CNN?)
- Take inspiration from modeling the interaction between two networks. Either to find discriminative regions of interest or perhaps make the method more interpretable.
- Take inspiration from pattern mining to extract data from DCNN layers.
- Investigate if text-based semantic part detectors can be implemented (Do we have code, is it hard, etc.). For example, it can be interesting to use for boats, as technical documents describing visual queues exist.
- Take inspiration from the methodology of covering up most discriminative parts to find other parts.
- Implement a receptive field or saliency map for visualizing part activations.

References

Dai, Haoyue (2020). "What do CNN neurons learn: Visualization & Clustering". In: *arXiv preprint arXiv:2010.11725*.

Gonzalez-Garcia, Abel, Davide Modolo, and Vittorio Ferrari (2018). "Objects as context for detecting their semantic parts". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6907–6916.

He, Xiangteng and Yuxin Peng (2019). "Fine-grained visual-textual representation learning". In: *IEEE Transactions on Circuits and Systems for Video Technology* 30.2, pp. 520–531.

Ji, Jinsheng et al. (2021). "Adversarial erasing attention for fine-grained image classification". In: *Multimedia Tools and Applications* 80.15, pp. 22867–22889.

Liang, Haoyu et al. (2020). "Training Interpretable Convolutional Neural Networks by Differentiating Class-specific Filters". In: *European Conference on Computer Vision*. Springer, pp. 622–638.

Liu, Qing et al. (2021). "CGPart: A Part Segmentation Dataset Based on 3D Computer Graphics Models". In: *arXiv preprint arXiv:2103.14098*.

Modolo, Davide and Vittorio Ferrari (2017). "Learning semantic part-based models from google images". In: *IEEE transactions on pattern analysis and machine intelligence* 40.6, pp. 1502–1509.

Peng, Yuxin, Xiangteng He, and Junjie Zhao (2017). "Object-part attention model for fine-grained image classification". In: *IEEE Transactions on Image Processing* 27.3, pp. 1487–1500.

Simon, Marcel, Erik Rodner, and Joachim Denzler (2014). "Part detector discovery in deep convolutional neural networks". In: *Asian Conference on Computer Vision*. Springer, pp. 162–177.

Wharton, Zachary, Ardhendu Behera, and Asish Bera (2021). "An attention-driven hierarchical multi-scale representation for visual recognition". In: *arXiv preprint arXiv:2110.12178*.

Xiao, Tianjun et al. (2015). "The application of two-level attention models in deep convolutional neural network for fine-grained image classification". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 842–850.

Yan, Tiantian et al. (2020). "Progressive learning for weakly supervised fine-grained classification". In: *Signal Processing* 171, p. 107519.

Zhang, Jian et al. (2019). "Unsupervised part mining for fine-grained image classification". In: *arXiv preprint arXiv:1902.09941*.

Zhang, Quanshi, Ying Nian Wu, and Song-Chun Zhu (2018). "Interpretable convolutional neural networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8827–8836.

Zhang, Quanshi and Song-Chun Zhu (2018). "Visual interpretability for deep learning: a survey". In: *arXiv preprint arXiv:1802.00614*.

Zhang, Quanshi et al. (2020). "Interpretable CNNs for object classification". In: *IEEE transactions on pattern analysis and machine intelligence* 43.10, pp. 3416–3431.

Zhang, Quanshi et al. (2021a). "Extraction of an Explanatory Graph to Interpret a CNN". In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 43.11, pp. 3863–3877.

Zhang, Tian et al. (2021b). "Progressive Co-Attention Network for Fine-grained Visual Classification". In: *arXiv preprint arXiv:2101.08527*.

Zhang, Yu et al. (2021c). "A survey on neural network interpretability". In: *IEEE Transactions on Emerging Topics in Computational Intelligence*.

Zhou, Bolei et al. (2014). "Object detectors emerge in deep scene cnns". In: *arXiv preprint arXiv:1412.6856*.

B.3 Investigation Of Constellation Models

Literature Collection Report: Investigation of Constellation Models

Stig Andre Rosenlund
William Emanuel Skreien Kaaby

July 2, 2022

1 Bibliography

1.1 High Relevancy

Title: "Spatial-aware graph relation network for large-scale object detection"
Authors: Xu et al. (2019)
Citations: 55

Abstract:
How to proper encode high-order object relation in the detection system without any external knowledge? How to leverage the information between co-occurrence and loca- tions of objects for better reasoning? These questions are key challenges towards large-scale object detection system that aims to recognize thousands of objects entangled with complex spatial and semantic relationships nowadays. Distilling key relations that may affect object recogni- tion is crucially important since treating each region sepa- rately leads to a big performance drop when facing heavy long-tail data distributions and plenty of confusing cate- gories. Recent works try to encode relation by construct- ing graphs, e.g. using handcraft linguistic knowledge be- tween classes or implicitly learning a fully-connected graph between regions. However, the handcraft linguistic knowl- edge cannot be individualized for each image due to the se- mantic gap between linguistic and visual context while the fully-connected graph is inefficient and noisy by incorpo- rating redundant and distracted relations/edges from irrel- evant objects and backgrounds. In this work, we introduce a Spatial-aware Graph Relation Network (SGRN) to adap- tive discover and incorporate key semantic and spatial rela- tionships for reasoning over each object. Our method con- siders the relative location layouts and interactions among which can be easily injected into any detection pipelines to boost the performance. Specifically, our SGRN integrates a graph learner module for learning a interpatable sparse graph structure to encode relevant contextual regions and a spatial graph reasoning module with learnable spatial Gaussian kernels to perform graph inference with spatial awareness. Extensive experiments verify the effectiveness of our method, e.g. achieving around 32% improvement on VG(3000 classes) and 28% on ADE in terms of mAP.

Comment:
This paper proposes a method of using a graph convolutional neural network to model the spatial spatial relationship between regions in the input. The authors claim that the method easily integrates with any modern visual systems. This method, to my understanding, does not provide a final hypothesis, but instead provides more information which can be used. Therefore, I think this approach is worth investigating further.

Title: "Towards human-like interpretable object detection via spatial relation encoding"
Authors: Kim, Park, and Ro (2020)
Citations: 2

Abstract:
The performance of recent deep neural networks in various computer vision areas such as object detection has increased significantly. Along with such advances, attempts to visualize and interpret the networks have been made in order to understand how a network predicts a certain result. However, there is a lack of research on ways to improve the interpretability of networks' features. In this paper, we propose a spatial relation reasoning (SRR) framework to encode interpretable networks' features, especially an object detector, by mimicking the human visual cognition system. The SRR consists of the spatial feature encoder (SFE) and the graph-based spatial relation encoder (GSRE) to consider spatial relationships between different parts of an object. So that, object detectors can encode spatially-related object features enabling humanlike visual interpretation. We verified the proposed framework with general object detectors on public datasets - PASCAL VOC and MS COCO.

Comment:
This paper proposes a method that tries to mimick the human visual system by explicity reason about spatial information. The authors claim that the approach easily can be attached to any two-stage region-based object detector. However, based on the diagram, is seems to me that the only true requirement is that the model it is attached to incorporates a method for generating region proposals. This is of course something that VGG-16 by default does not do but it would be possible to implement this or perhaps just approximate the output of a region proposal network. Overall, I think this approach seems promising and should be looked into further.

Title: "A coarse-to-fine taxonomy of constellations for fast multi-class object detection"
Authors: Fidler, Boben, and Leonardis (2010)
Citations: 31

Abstract:
In order for recognition systems to scale to a larger number of object categories building visual class taxonomies is important to achieve running times logarithmic in the number of classes [1, 2]. In this paper we propose a novel approach for speeding up recognition times of multi-class part-based object representations. The main idea is to construct a taxonomy of constellation models cascaded from coarse-to-fine resolution and use it in recognition with an efficient search strategy. The taxonomy is built automatically in a way to minimize the number of expected computations during recognition by optimizing the cost-to-power ratio [3]. The structure and the depth of the taxonomy is not pre-determined but is inferred from the data. The approach is utilized on the hierarchy-of-parts model [4] achieving efficiency in both, the representation of the structure of objects as well as in the number of modeled object classes. We achieve speed-up even for a small number of object classes on the ETHZ and TUD dataset. On a larger scale, our approach achieves detection time that is logarithmic in the number of classes.

Comment:
This paper proposes a method the learns a hierarchical representation of objects and parts. The root node represents the coarsest representation in the tree, while the leaf nodes are the fine representations. The approach essentially considers three aspects: a feature from an input image, its location in said image and occlusion. It seems to me that it would be possible to replace their method of extracting features with our planned approach based on an Interpretable CNN layer. Therefore, this paper should be investigated further.

1.2 Medium Relevancy

Title: "Part-based R-CNNs for fine-grained category detection"
Authors: Zhang et al. (2014)
Citations: 1073

Abstract:
Semantic part localization can facilitate fine-grained categorization by explicitly isolating subtle appearance differences associated with specific object parts. Methods for pose-normalized representations have been proposed, but generally presume bounding box annotations at test time due to the difficulty of object detection.We propose a model for fine-grained categorization that overcomes these limitations by leveraging deep convolutional features computed on bottom-up region proposals. Our method learns whole-object and part detectors, enforces learned geometric constraints between them, and predicts a fine-grained category from a pose-normalized representation. Experiments on the Caltech- UCSD bird dataset confirm that our method outperforms state-of-the-art fine-grained categorization methods in an end-to-end evaluation without requiring a bounding box at test time.

Comment:
This paper proposes a method using a R-CNN to perform object detection and part localization. The approach requires both object and part annotations at training time. The method generates n part proposals. From among these candidates a number of parts are selected using a geometric constraint function. They experiment with two different functions for defining constraints: a mixture of Gaussian models and a K-Nearest Neighbors with a Gaussian model fitted to the neighbours. The part detectors are not particularly interesting, but it may be worthwhile considering the applicability of the geometric constraint in our implementation of a Constellation model.

Title: "Part and appearance sharing: Recursive Compositional Models for multi-view"
Authors: Zhu et al. (2010)
Citations: 131

Abstract:
We propose Recursive Compositional Models (RCMs) for simultaneous multi-view multi-object detection and parsing (e.g. view estimation and determining the posi- tions of the object subparts). We represent the set of ob- jects by a family of RCMs where each RCM is a probability distribution defined over a hierarchical graph which corre- sponds to a specific object and viewpoint. An RCM is con- structed from a hierarchy of subparts/subgraphs which are learnt from training data. Part-sharing is used so that dif- ferent RCMs are encouraged to share subparts/subgraphs which yields a compact representation for the set of objects and which enables efficient inference and learning from a limited number of training samples. In addition, we use appearance-sharing so that RCMs for the same object, but different viewpoints, share similar appearance cues which also helps efficient learning. RCMs lead to a multi-view multi-object detection system. We illustrate RCMs on four public datasets and achieve state-of-the-art performance.

Comment:
This paper proposes a method to develop a hierarchical representation of objects and their constituent parts using something they term as Recursive Compositional Models (RCM). These hierachies of RCMs can share part and appearance representations between themselves. The method seems to operate on image patches, and it is not immediatly obvious how this could be extended to work on feature maps. It would probably be more worthwhile to pursue other approaches, although hierarchical approaches such as the one proposed in this paper would be desireable. However, I do not want to completly rule out this approach so I place it in medium relevancy. I should also mention that Yuille is one of the co-authors of the paper.

1.3 Low Relevancy

Title: "Which and how many regions to gaze: Focus discriminative regions for fine-grained visual categorization"
Authors: He, Peng, and Zhao (2019)
Citations: 34

Abstract:
Fine-grained visual categorization (FGVC) aims to discriminate similar subcategories that belong to the same superclass. Since the distinctions among similar subcategories are quite subtle and local, it is highly challenging to distinguish them from each other even for humans. So the localization of distinctions is essential for fine-grained visual categorization, and there are two pivotal problems: (1) Which regions are discriminative and representative to distinguish from other subcategories? (2) How many discriminative regions are necessary to achieve the best categorization performance? It is still difficult to address these two problems adaptively and intelligently. Artificial prior and experimental validation are widely used in existing mainstream methods to discover which and how many regions to gaze. However, their applications extremely restrict the usability and scalability of the methods. To address the above two problems, this paper proposes a multi-scale and multi-granularity deep reinforcement learning approach (M2DRL), which learns multi-granularity discriminative region attention and multi-scale region-based feature representation. Its main contributions are as follows: (1) Multi-granularity discriminative localization is proposed to localize the distinctions via a two-stage deep reinforcement learning approach, which discovers the discriminative regions with multiple granularities in a hierarchical manner ("which problem"), and determines the number of discriminative regions in an automatic and adaptive manner ("how many problem"). (2) Multiscale representation learning helps to localize regions in different scales as well as encode images in different scales, boosting the fine-grained visual categorization performance. (3) Semantic reward function is proposed to drive M2DRL to fully capture the salient and conceptual visual information, via jointly considering attention and category information in the reward function. It allows the deep reinforcement learning to localize the distinctions in a weakly supervised manner or even an unsupervised manner. (4) Unsupervised discriminative localization is further explored to avoid the heavy labor consumption of annotating, and extremely strengthen the usability and scalability of our M2DRL approach. Compared with state-of-the-art methods on two widely-used fine-grained visual categorization datasets, our M2DRL approach achieves the best categorization accuracy.

Comment:
This paper proposes a method that uses reinforcement learning to learn discriminative features and to handle different scales. Overall, the process seems quite involved and difficult to adapt to our requirements.

Title: "Attention CoupleNet: Fully convolutional attention coupling network for object detection"
Authors: Zhu et al. (2018)
Citations: 88

Abstract:
The field of object detection has made great progress in recent years. Most of these improvements are derived from using a more sophisticated convolutional neural network. However, in the case of humans, the attention mechanism, global structure information, and local details of objects all play an important role for detecting an object. In this paper, we propose a novel fully convolutional network, named as Attention CoupleNet, to incorporate the attention-related information and global and local information of objects to improve the detection performance. Specifically, we first design a cascade attention structure to perceive the global scene of the image and generate class-agnostic attention maps. Then the attention maps are encoded into the network to acquire object-aware features. Next, we propose a unique fully convolutional coupling structure to couple global structure and local parts of the object to further formulate a discriminative feature representation. To fully explore the global and local properties, we also design different coupling strategies and normalization ways to make full use of the complementary advantages between the global and local information. Extensive experiments demonstrate the effectiveness of our approach. We achieve state-of-the-art results on all three challenging data sets, i.e., a mAP of 85.7% on VOC07, 84.3% on VOC12, and 35.4% on COCO. Codes are publicly available at https://github.com/tshizys/CoupleNet.

Comment:
This paper proposes a variation of CNN which more explicitly model the relationship between local and global structures. This approach is not really applicable for our case since it would not work well with what we have done so far. Furthermore, the proposed method does not seem to offer increased interpreatability.

1.4 Ancilliary

Title: "Fine-grained categorization via CNN-based automatic extraction and integration of object-level and part-level features"
Authors: Sun, Sun, and Yeung (2017)
Citations: 11

Abstract:
Fine-grained categorization can benefit from part-based features which reveal subtle visual differences between object categories. Handcrafted features have been widely used for part detection and classification. Although a recent trend seeks to learn such features automatically using powerful deep learning models such as convolutional neural networks (CNN), their training and possibly also testing require manually provided annotations which are costly to obtain. To relax these requirements, we assume in this study a general problem setting in which the raw images are only provided with objectlevel class labels for model training with no other side information needed. Specifically, by extracting and interpreting the hierarchical hidden layer features learned by a CNN, we propose an elaborate CNN-based system for fine-grained categorization. When evaluated on the Caltech-UCSD Birds-200-2011, FGVC-Aircraft, Cars and Stanford dogs datasets under the setting that only object-level class labels are used for training and no other annotations are available for both training and testing, our method achieves impressive performance that is superior or comparable to the state of the art. Moreover, it sheds some light on ingenious use of the hierarchical features learned by CNN which has wide applicability well beyond the current fine-grained categorization task.

Comment:
This paper proposes a method using multiple CNNs to detect parts and objects. The approach only requires object-level annotation at training time. The proposed method is based upon extracting information from the feature maps of a pre-trained CNN. From these feature maps they construct object-level and part-level masks using a multi-step process. Masks are then selected if they are contigous and their centroid lies within the object-level mask. The centroids of the selected part-level masks are then clustered using K-Means to group feature maps which corresponds to the same part. They use the masks to generate object-level and part-level images which are fed into new CNNs which are used for classification. Of interest in this paper is their process of cleaning up noisy feature maps and the two algorithms for determining K in K-Means.

Title: "Visual concepts and compositional voting"
Authors: Wang et al. (2017)
Citations: 35

Abstract:
It is very attractive to formulate vision in terms of pattern theory [26], where patterns are defined hierarchically by compo- sitions of elementary building blocks. But applying pattern theory to real world images is currently less successful than discrimina- tive methods such as deep networks. Deep networks, however, are black-boxes which are hard to interpret and can easily be fooled by adding occluding objects. It is natural to wonder whether by better understanding deep networks we can extract building blocks which can be used to develop pattern theoretic models. This mo- tivates us to study the internal representations of a deep network using vehicle images from the PASCAL3D+ dataset. We use clus- tering algorithms to study the population activities of the features and extract a set of visual concepts which we show are visually tight and correspond to semantic parts of vehicles. To analyze this we annotate these vehicles by their semantic parts to create a new dataset, VehicleSemanticParts, and evaluate visual concepts as un- supervised part detectors. We show that visual concepts perform fairly well but are outperformed by supervised discriminative methods such as Support Vector Machines (SVM). We next give a more detailed analysis of visual concepts and how they relate to seman- tic parts. Following this, we use the visual concepts as building blocks for a simple pattern theoretical model, which we call com- positional voting. In this model several visual concepts combine to detect semantic parts. We show that this approach is significantly better than discriminative methods like SVM and deep networks trained specifically for semantic part detection. Finally, we return to studying occlusion by creating an annotated dataset with occlu- sion, called VehicleOcclusion, and show that compositional voting outperforms even deep networks when the amount of occlusion be- comes large.

Comment:
This paper analyses the internals of VGG16 and in many ways can be consiered an amalgamation of other work by the Yuille group. Specifcally they apply both K-Means++ and von Mises-Fisher to find visual concepts from a set of feature vectors extracted from pool4. They seem to apply a simple grid search from a set of candiate K values to find the K which yields the best result. They also define a strategy for merging clusters, however they state that the results with and without merging is very similar. They state that visual concepts can correspond to multiple semantic parts. They seem to get quite good results, but we were unable to achieve such clean results when we tried a similar approach. One thing we should consider investigating is visual pattern models, which seems to represent a significant source of inspiration to the Yuille group. I have added a couple of the papers they cite in OneDrive.

1.5 Not Relevant

Title: "Probabilistic Ranking-Aware Ensembles for Enhanced Object Detections"
Authors: Mao et al. (2021)
Citations: 1

Abstract:
Model ensembles are becoming one of the most effective approaches for improving object detection performance already optimized for a single detector. Conventional methods directly fuse bounding boxes but typically fail to consider proposal qualities when combining detectors. This leads to a new problem of confidence discrepancy for the detector ensembles. The confidence has little effect on single detectors but significantly affects detector ensembles. To address this issue, we propose a novel ensemble called the Probabilistic Ranking Aware Ensemble (PRAE) that refines the confidence of bounding boxes from detectors. By simultaneously considering the category and the location on the same validation set, we obtain a more reliable confidence based on statistical probability. We can then rank the detected bounding boxes for assembly. We also introduce a bandit approach to address the confidence imbalance problem caused by the need to deal with different numbers of boxes at different confidence levels. We use our PRAE-based nonmaximum suppression (P-NMS) to replace the conventional NMS method in ensemble learning. Experiments on the PASCAL VOC and COCO2017 datasets demonstrate that our PRAE method consistently outperforms state-of-the-art methods by significant margins.

Comment:
This paper proposes a method for improving the creation of bounding boxes in the context of ensembles. This paper is not really relevant beyond the fact that we in essence have an ensemble of part detectors. However, it is not immediatly clear how this approach could be modified to suit our needs.

1.6 Unevaluated

2 Conclusion

References

Fidler, Sanja, Marko Boben, and Aleš Leonardis (2010). "A coarse-to-fine taxonomy of constellations for fast multi-class object detection". In: *European Conference on Computer Vision*. Springer, pp. 687–700.

He, Xiangteng, Yuxin Peng, and Junjie Zhao (2019). "Which and how many regions to gaze: Focus discriminative regions for fine-grained visual categorization". In: *International Journal of Computer Vision* 127.9, pp. 1235–1255.

Kim, Jung Uk, Sungjune Park, and Yong Man Ro (2020). "Towards human-like interpretable object detection via spatial relation encoding". In: *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, pp. 3284–3288.

Mao, Mingyuan et al. (2021). "Probabilistic Ranking-Aware Ensembles for Enhanced Object Detections". In: *arXiv preprint arXiv:2105.03139*.

Sun, Ting, Lin Sun, and Dit-Yan Yeung (2017). "Fine-grained categorization via CNN-based automatic extraction and integration of object-level and part-level features". In: *Image and Vision Computing* 64, pp. 47–66.

Wang, Jianyu et al. (2017). "Visual concepts and compositional voting". In: *arXiv preprint arXiv:1711.04451*.

Xu, Hang et al. (2019). "Spatial-aware graph relation network for large-scale object detection". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9298–9307.

Zhang, Ning et al. (2014). "Part-based R-CNNs for fine-grained category detection". In: *European conference on computer vision*. Springer, pp. 834–849.

Zhu, Long et al. (2010). "Part and appearance sharing: Recursive Compositional Models for multi-view". In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, pp. 1919–1926.

Zhu, Yousong et al. (2018). "Attention CoupleNet: Fully convolutional attention coupling network for object detection". In: *IEEE Transactions on Image Processing* 28.1, pp. 113–126.

Appendix C

Copyright Permissions

C.1 Permissions From IEEE

This section contains permissions given for IEEE materials.

For an explanation of IEEE permissions see [Iee].

C.1.1 [FMR08]

C.1.2 [OB07]

C.1.3 [Dai+14]

Appendix D

Failure Situations

D.1 Approach A - Bag Of Words

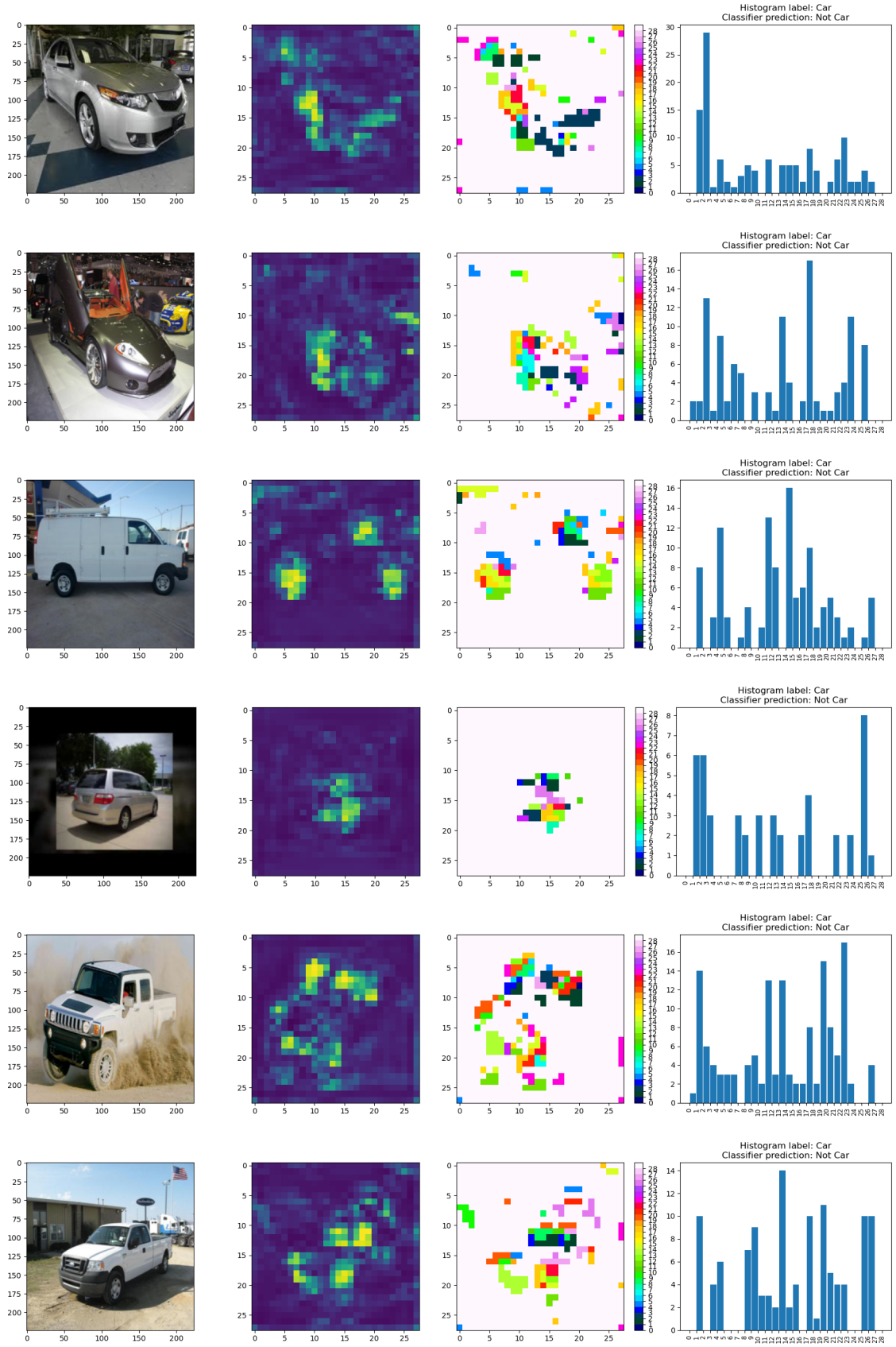


Figure D.1: The false negative predictions of the Bag of Words approach. Each row is one example. The first column is the original image, the second column is the corresponding segmentation map $S_{W,H}$, the third column is the corresponding detection map $D_{W,H}$ for $t = 0.25$ and the fourth column is the generated part hit histogram.

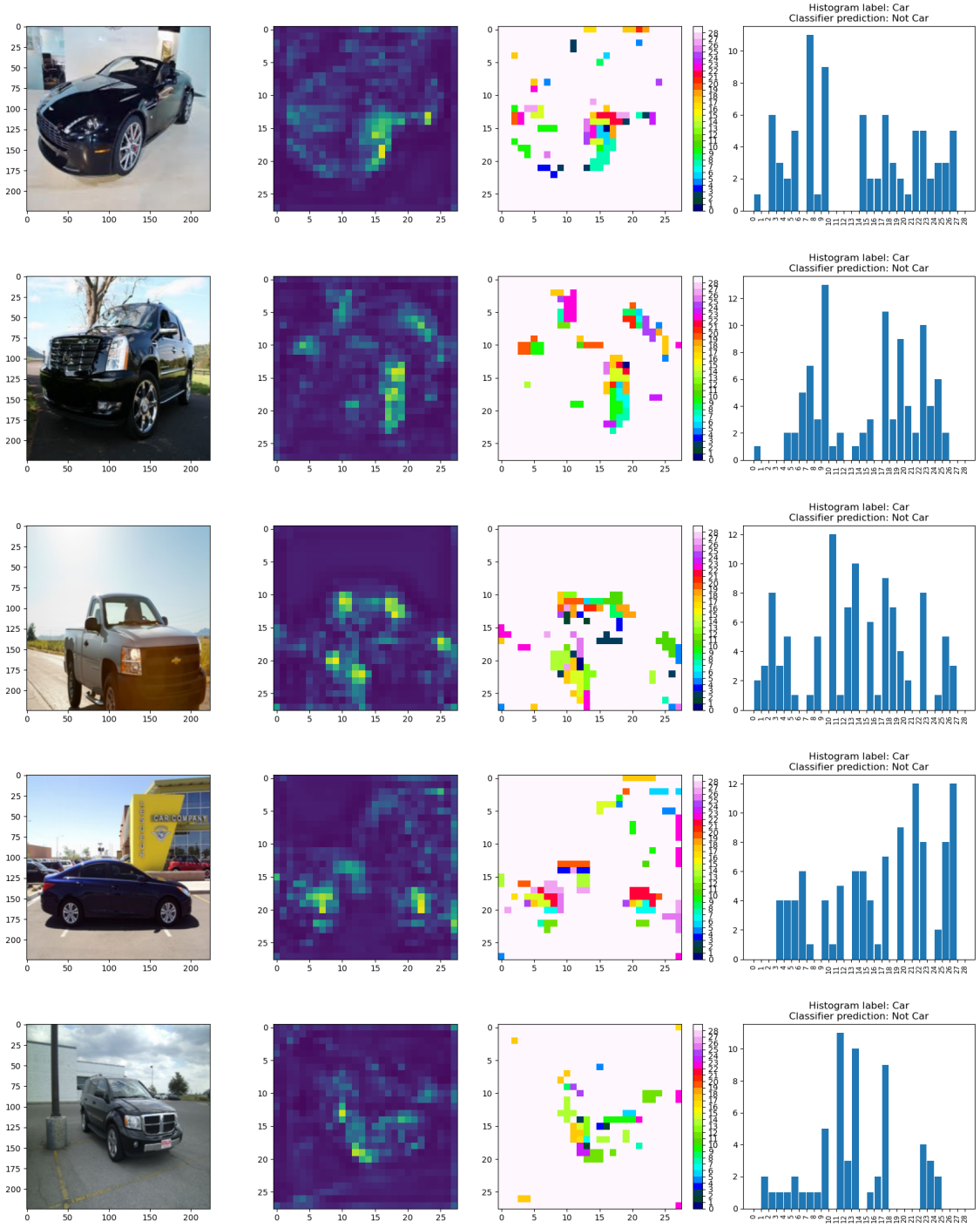


Figure D.2: The false negative predictions of the Bag of Words approach. Each row is one example. The first column is the original image, the second column is the corresponding segmentation map $S_{W,H}$, the third column is the corresponding detection map $D_{W,H}$ for $t = 0.25$ and the fourth column is the generated part hit histogram.

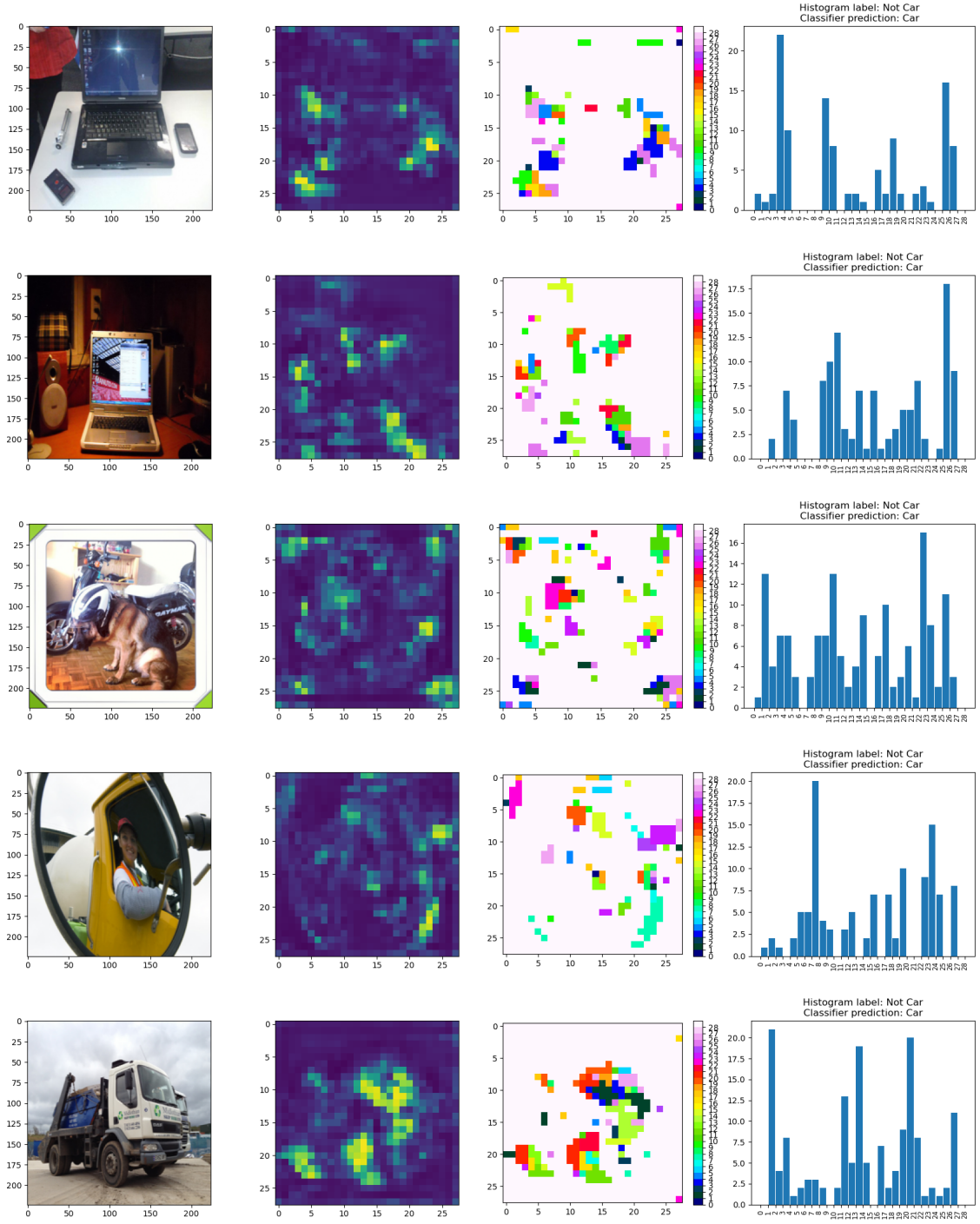


Figure D.3: The false positive predictions of the Bag of Words approach. Each row is one example. The first column is the original image, the second column is the corresponding segmentation map $S_{W,H}$, the third column is the corresponding detection map $D_{W,H}$ for $t = 0.25$ and the fourth column is the generated part hit histogram.

Appendix E

Derivation Of The Bhattacharyya Loss Function

As stated in section 6.2.5, we define the two histograms in terms of fitted normal distributions. The use of normal distributions is done so as to simplify the process of tracing the loss back to the filter weights. The two normal distributions are defined in Equation E.1 in terms of the maximum activation A.

$$p(A) \stackrel{\text{def}}{=} \frac{1}{\sigma_p\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{A-\mu_p}{\sigma_p})^2} \qquad q(A) \stackrel{\text{def}}{=} \frac{1}{\sigma_q\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{A-\mu_q}{\sigma_q})^2} \tag{E.1}$$

The Bhattacharyya Coefficient is defined in Equation E.2.

$$BC = \int \sqrt{p(A)q(A)}\, dA \tag{E.2}$$

We merge the two terms $p(A)$ and $q(A)$, the steps of which is shown below:

$$p(A)q(A) = \frac{1}{\sigma_p\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{A-\mu_p}{\sigma_p})^2} \cdot \frac{1}{\sigma_q\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{A-\mu_q}{\sigma_q})^2}$$

$$= \frac{1}{2\pi\sigma_p\sigma_q} \exp(-\frac{1}{2}\frac{(A-\mu_p)^2}{\sigma_p^2} - \frac{1}{2}\frac{(A-\mu_q)^2}{\sigma_q^2})$$

A is defined in terms of weights w and the input x using the term $A = \sum\sum wx$. This term merely represents a convolutional operation using some arbitrary kernel k where we have dropped the indices for convenience. We replace A with the new definition:

$$p(A)q(A) = \frac{1}{2\pi\sigma_p\sigma_q} exp(-\frac{1}{2}\frac{((\sum\sum wx)-\mu_p)^2}{\sigma_p^2} - \frac{1}{2}\frac{((\sum\sum wx)-\mu_q)^2}{\sigma_q^2}$$

The next step is to find the gradient from Equation E.2. We will calculate the partial derivative in terms of some weight particular $w_{i,j}$, and will again forego the indices for

the sake of convenience. The partial derivative of the loss with respect to some weight w is found using the chain rule, as shown in Equation E.3.

$$\begin{aligned}\frac{\partial \mathcal{L}_{BC}}{\partial w} &= \frac{\partial \mathcal{L}_{BC}}{\partial A}\frac{\partial A}{\partial w}(\int \sqrt{p(A)q(A)}\, dA) \\ &= \frac{\partial}{\partial w}(\sqrt{\frac{1}{2\pi\sigma_p\sigma_q}\exp(-\frac{1}{2}\frac{((\sum\sum wx)-\mu_p)^2}{\sigma_p^2} - \frac{1}{2}\frac{((\sum\sum wx)-\mu_q)^2}{\sigma_q^2})})\end{aligned} \tag{E.3}$$

We apply the chain rule to the final term of Equation E.3 to find the partial derivative with respect to w, the result of which is shown in Equation E.4.

$$\frac{\partial \mathcal{L}_{BC}}{\partial w} = \frac{1}{2\sqrt{\phi}}\frac{1}{2\pi\sigma_p\sigma_q}e^{\psi}\theta = \frac{\theta e^{\psi}}{4\pi\sigma_p\sigma_q\sqrt{\phi}} \tag{E.4}$$

$$\phi \stackrel{\text{def}}{=} \frac{1}{2\pi\sigma_p\sigma_q}\exp(-\frac{1}{2}\frac{((\sum\sum wx)-\mu_p)^2}{\sigma_p^2} - \frac{1}{2}\frac{((\sum\sum wx)-\mu_q)^2}{\sigma_q^2})$$

$$\frac{\partial \phi}{\partial w} \stackrel{\text{def}}{=} e^{\psi}\theta$$

$$\psi \stackrel{\text{def}}{=} -\frac{1}{2}\frac{((\sum\sum wx)-\mu_p)^2}{\sigma_p^2} - \frac{1}{2}\frac{((\sum\sum wx)-\mu_q)^2}{\sigma_q^2}$$

$$\theta \stackrel{\text{def}}{=} \frac{\partial \psi}{\partial w} = -\frac{x((\sum\sum wx)-\mu_p)^2}{\sigma_p^2} - \frac{x((\sum\sum wx)-\mu_q)^2}{\sigma_q^2}$$

Kaaby, Rosenlund

Structured Object Detection

NTNU

Kunnskap for en bedre verden