Applied Intelligence
https://doi.org/10.1007/s10489-021-02908-z

Deep learning based decomposition for visual navigation in industrial platforms

Youcef Djenouri[1] · Johan Hatleskog[2] · Jon Hjelmervik[1] · Elias Bjorne[3] · Trygve Utstumo[3] · Milad Mobarhan[3]

Accepted: 7 October 2021

Abstract
In the heavy asset industry, such as oil & gas, offshore personnel need to locate various equipment on the installation on a daily basis for inspection and maintenance purposes. However, locating equipment in such GPS denied environments is very time consuming due to the complexity of the environment and the large amount of equipment. To address this challenge we investigate an alternative approach to study the navigation problem based on visual imagery data instead of current ad-hoc methods where engineering drawings or large CAD models are used to find equipment. In particular, this paper investigates the combination of deep learning and decomposition for the image retrieval problem which is central for visual navigation. A convolutional neural network is first used to extract relevant features from the image database. The database is then decomposed into clusters of visually similar images, where several algorithms have been explored in order to make the clusters as independent as possible. The Bag-of-Words (BoW) approach is then applied on each cluster to build a vocabulary forest. During the searching process the vocabulary forest is exploited to find the most relevant images to the query image. To validate the usefulness of the proposed framework, intensive experiments have been carried out using both standard datasets and images from industrial environments. We show that the suggested approach outperforms the BoW-based image retrieval solutions, both in terms of computing time and accuracy. We also show the applicability of this approach on real industrial scenarios by applying the model on imagery data from offshore oil platforms.

Keywords Information retrieval · Deep learning · Decomposition · Place recognition

1 Introduction

In our everyday life we are increasingly dependent on mobile tools, like Google Maps, to find our way. In an industrial context we face the same challenge for human navigation, but we would often find ourselves inside an industrial structure without GPS reception. For instance, in the heavy asset industry such as oil & gas, offshore personnel need to locate various equipment on the oil platform on a daily basis for inspection and maintenance purposes. Locating equipment in such environments is very time consuming due to the complexity of the environment and the large amount of equipment. Ad-hoc methods are often used, by comparing area plans, Piping and Instrumentation Diagrams (P&ID) and other available information. A more efficient method being explored is having a Computer-aided design (CAD) model easily available and indexed so that equipment easily can be shown in a full CAD model. Thus, locating the relevant equipment in a full CAD assembly can be straightforward [12]. However, locating the current position of the worker is less trivial. As an industrial heavy asset facility will typically have multiple levels and one would be inside a steel and concrete structure, one does not have the luxury of having GPS to provide an initial location hint. Also since many industrial facilities have strict maintenance programs, which makes infrastructure installation in general expensive, installing such positioning infrastructure is often not an option. Thus, users typically need to compare large objects in the nearby environment with those in the CAD model to estimate their current location. This is quite inconvenient, especially because CAD models do not provide a photo-realistic representation of the environment.

In this paper, we investigate an alternative approach to address the navigation problem based on visual imagery data. At the heart of the visual navigation problem is

✉ Youcef Djenouri
youcef.djenouri@sintef.no

Extended author information available on the last page of the article.

Published online: 20 October 2021

the place recognition task, which involves recognition and localization of a given query image [5, 31]. Place recognition may be interpreted as an information retrieval problem, where the purpose is to retrieve a place (set of images) by matching a query image with images in a preexisting database. Despite many studies on the place recognition problem, both based on traditional Bag-of-Words (BoW) image retrieval solutions [3, 30, 40] and deep learning based approaches [22, 38, 41, 42], the place recognition problem still remains extremely challenging. This is especially the case for homogeneous environments where nearly identical objects occur on different locations, such as in industrial environments. Furthermore, limited hardware and strong requirements on the processing speed in the industry add additional layers of complexity to the problem. Decomposition may be an alternative way to address this challenge. Similar ideas [4, 21] have been explored, where decomposition is used to split the images into groups, and return the group of images most similar to the image query as output of the image retrieval process. These solutions are limited in accuracy where a high number of false positives are identified.

In this study, we propose a hybrid model where decomposition and convolutional neural network are combined with the traditional BoW approach. In this model, referred to as the DCNN-vForest (Decomposition Convolution Neural Network for vocabulary Forest) model, a set of database images are decomposed into several independent clusters (see Fig. 1). In this context, we adopt different clustering algorithms, such as kmeans [32], kmeans++ [14], and mini batch kmeans [37], to decompose the image database into clusters of images aiming to minimize the number of the shared features among clusters, and maximize the number of shared features within each cluster. During the searching process, only the most similar clusters are explored, which significantly speeds up the image search. This performance is reached by the fact that vForest considerably reduces the word space of the BoW solutions, yielding better accuracy than BoW with the same number of words. The main contributions of this paper are listed in the following:

1. We combine both global image features determined by a Convolutional Neural Network (CNN) and local image features determined by the Scale-Invariant Feature Transform (SIFT) extractor. Global features are used to separate the similar images from the non-similar ones, while the local features are used to describe the images inside each cluster.
2. We propose two novel strategies that use the clusters for searching the relevant images to the query image. The first strategy only explores the most similar cluster to the query image, while the second approach also uses the neighborhood information of the most similar cluster.
3. We conduct extensive analysis of the computational time and accuracy. The results show that the DCNN-vForest model outperforms the BoW-based image retrieval solutions, both in terms of computing time and accuracy. We also show the applicability of the DCNN-vForest model on real industrial scenarios by applying the model on imagery data from offshore oil platforms.

The rest of the paper is organized as follows. Section 2 reviews on the existing image retrieval and place recognition based solutions. Section 3 presents the proposed approach and its main components. Section 4 presents the experimental study and results. Section 5 presents the main finding of applying the DCNN-vForest model on imagery data from heavy asset industry. Finally, section VI concludes the paper.

2 Related work

This research focuses on two topics: image search and autonomous navigation. In the following, existing literature in the two research topics have been analyzed and reviewed.

2.1 Image search

Arandjelovic et al. [2] proposed a solution to the reverse image search problem applied to a large scale image dataset covering city scenes. They apply a weakly supervised learning algorithm, to predict the origin of the given query image. This approach outperforms the traditional reverse image based solutions, however it is not straightforward to learn an accurate model for reverse image search, in particular for large corpus like city images data. Cao et al. [9] developed Deep Visual-Semantic Quantization (DVSQ) for learning deep quantization models, and semantic information from the image database. It combines both learning deep visual-semantic embeddings and quantizers using hybrid networks and well-specified loss functions. Zhang et al. [44] proposed the Unsupervised Generative Adversarial Cross-modal Hashing (UGACH) approach to capture meaningful nearest neighbours of different modalities for cross-modal image retrieval. It used the generative adversarial network for unsupervised representation learning of image features. Liu et al. [30] proposed End-to-End BoW (E2BoW) using the deep convolutional network. Instead of performing different steps in making the vocabulary tree of the bag of words model, this algorithm investigates only a single step by learning the image features, identifies the visual words, and then determines the cosine similarity between

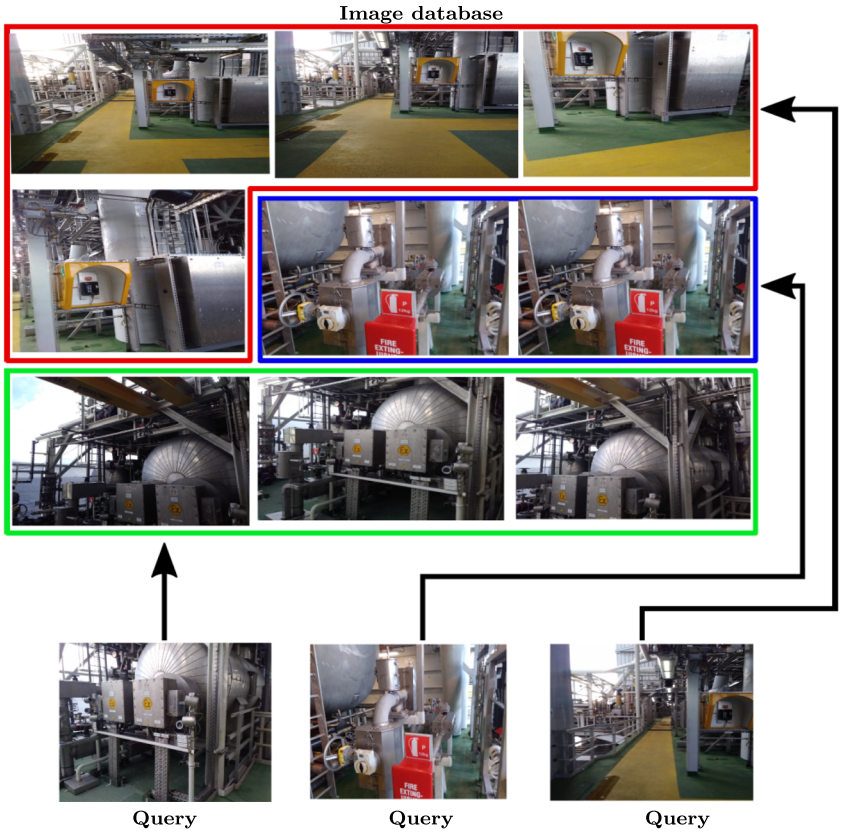

Fig. 1 Illustrative example of the DCNN-vForest model. The image database is grouped into three clusters, marked by different colors. For each query image at the bottom, only clusters with the most similar images are explored and this leads to an overall increase in performance

the image database and the query image. These approaches are costly, both in terms of computational and memory resources, and that the number of neurons of the last fully connected layer is highly depend to the number of bag words of the image database. Yang et al. [40] proposed the Hierarchical Deep Embedding (HDE) approach with the use of multiple feature extractors for retrieving the Aurora satellite images. It incorporates the local features (using SIFT algorithm), the regional and the global features (using CNN model) in constructing the vocabulary tree of the image database. This approach requires large computational effort, but it outperforms the state-of-the-art solutions in terms of accuracy. To improve the runtime performance of such an approach, Zhan et al. [43] proposed a GPU-based parallel approach in extracting the features, where one GPU-block is responsible for computing the local features using SIFT, and another GPU-block is responsible for determining the global features using CNN. This approach gives a lower computation time compared to the previous one, however, the GPU resources concurrently assigned to two jobs (local and global extractors), reduces the capability of the approach to run complex CNN architectures such as VGG19. Ahmad et al. [27] proposed a hybrid bag of words and VLAD solution. The features of the image databases are first extracted using the VLAD network, and the bag of words algorithm is then performed from the features extracted. Doan et al. [19] proposed an incremental hidden Markov model for recognizing images in autonomous driven system, which allows to exploit the temporal features of the images in the query, and study the correlation between the temporal and the spatial dimensions of the images database. In the same context, Vysotska et al. [39] deal with seasonal weather change to localize vehicles in a map by combining hashing-based image retrieval, and contextual information represented by a data association graph. Hong et al. [25] proposed a text-based algorithm for reverse image search. The textual descriptors are first generated from map images. To remove noise, the Levenshtein distance is then calculated between the recognized text, and textual descriptor of the query. The topological localization which explores both spatial and temporal information is finally adopted to recognize the place of the query. Chancan et al. [11] incorporated the image retrieval with the neuroscience-oriented model and propose a one-dimensional continuous attractor neural network with a compact, sparse two-layer neural network inspired by brain architecture. Cao et al. [8] addressed the variation of the perceptual condition issue such as all weather, times-of-day, seasons and viewpoint shifts, and developed an adapted light detection and ranging algorithm. A new scene representation is integrated by merging context and layout descriptors to reach accurate place recognition across seasons. The sequence-based

temporal consistency is also developed to handle scenes with similar objects with local structural changes. Givek [24] developed the Scale-Space Multi-View Bag of Words (SSMV-BoW) approach for addressing the overlooking spatial information limitation of the BoW. It considers multi-scaling when determining the features with the SIFT extractor. The semantic information of the visual features is also used in the image search process.

2.2 Autonomous navigation

Anwar et al. [1] suggested the use of transfer learning for reducing the training time of deep learning architectures for applications in autonomous navigation for drones. A fine-tuned process is also investigated for the last fully connected layers. Carrio et al. [10] introduced a new strategy for drone localization based on both segmentation and object detection models. The training data of the object detection model, composed of both images of flying drones and segmentation maps, are first created. The obtained bounding boxes are used for 3D position estimation of the detected drones. Sina et al. [34] proposed a solution to the navigation of constrained robots problem in a dark underground mine environment, while exploring unknown regions. It improved the vision ability of the aerial areas by minimizing the number of sensors allocated in the navigation system. De Queiroz Mendes et al. [36] developed a hybrid framework which combines the convolutional neural network and encoder-decoder architectures for autonomous navigation. It also proposed a new loss function to optimize the single image depth estimation. The integration of multiple semantic surface and depth knowledge is also investigated in the training process. Phil et al. [26] suggested the use of evolutionary algorithm [17], and in particular the particle swarm optimization in the autonomous navigation process. It simulates the behaviours of particles when exploiting the embodied dynamics, and project this simulation in the robotic system for autonomous navigation. Matthias et al. [20] developed a strategy for localization verification of the robots in autonomous navigation setting. It means checking the correctness of the current position of the robot in a real time scenario. The convolutional neural network with the recurrent neural network is used to estimate the temporal patterns when the localization is missing. A weak classifier is combined with these patterns to boost the identification of the missed localization. Lee et al. [29] combined the multi-task learning, the convolutional neural network, and controllers to improve the stability of the actual autonomous driving system. The cars on the road are first detected using both regression and classification tasks with hybrid multi-task convolutional neural network architecture. The controller algorithm is then applied to mitigate collisions in a real-time scenario. Mao et al. [35] addressed the multi-scale vehicle detection, and the overlapping objects issues in autonomous vehicle settings. It extended the YOLOv3 by improving the feature extraction process while using the inverted residuals strategy on the convolution layers. Spatial pyramid pooling blocks are also integrated for deriving the multi-scale information of each car. Finally, and in order to solve the overlapping between cars, the non maximum suppression operator is replaced by the soft non maximum suppression operator. Dinh et al. [15] used the transfer learning for improving the autonomous vehicle system on two cameras with different focal lengths. The output of the autonomous vehicle model with the parameters of the first camera is projected to the parameters of the second camera. The evolutionary computation algorithm is also integrated to find the different correlations among the parameters of both cameras.

2.3 Discussion

From this literature review, solutions to image retrieval, place recognition and autonomous navigation algorithms are divided into two categories: i) Solutions exploring traditional pipelines such as bag of words to find the relevant output according to the user settings. ii) Other solutions which explore artificial intelligence to train deep learning models in retrieving the relevant information according to different user settings. All these solutions suffer from two main issues: The first one is that the existing solutions are not able to recognize areas in homogeneous buildings, where same objects may be occurring in different places and in different rooms. The second issue is that none of the existing solutions meet our requirements in place recognition when it comes to both accuracy and runtime. Motivated by the success of decomposition-based algorithms [6, 16, 18] in solving complex problems, in the next section, we propose a DCNN-vForest algorithm to efficiently explore the image database, and accurately satisfy user queries.

3 DCNN-vForest

This section presents the proposed DCNN-vForest framework, which combines decomposition, convolutional neural network, and BoW to solve the image retrieval problem. This is a generic framework, where any decomposition and searching algorithms may be used. As illustrated in Fig. 2, the designed framework consists of three main steps:

1. **Decomposition.** Images in the database are grouped into clusters of visually similar images. In particular, the clustering aims to minimize the number of shared features across clusters, and maximize the

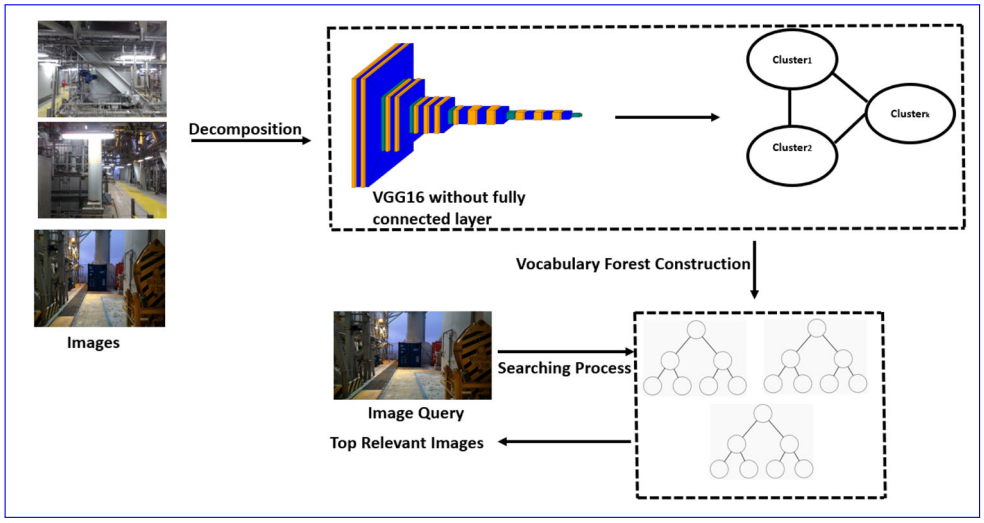

Fig. 2 DCNN-vForest Framework

number of shared features within each cluster. To efficiently explore the problem, the global features, derived using the convolutional neural network, have been clustered using three different decomposition algorithms: kmeans, kmeans++, and mini batch kmeans.

2. **Vocabulary Forest Construction.** In this step, the vocabulary forest is built based on the output of the previous step. Thus, a vocabulary tree is created for each cluster of images using the BoW approach. To accurately manage and store the vocabulary tree for each cluster, a new structure is defined, called a vocabulary forest (vForest for short). This vForest contains information related to both the vocabulary trees, and the centers of the clusters created in the decomposition step.
3. **Searching Process.** In this step the vForest is explored to find the most similar images to the query image. Instead of exploring the entire set of image features, only the most similar clusters to the image query are visited. We propose two different strategies in order to efficiently explore the vForest. The first one only targets the most similar cluster to the image query, while the second one also targets the neighbours of the most similar cluster to the image query.

The detailed explanation of each step is given in the following subsections.

3.1 Decomposition

3.1.1 Principle

The aim of this step is to divide the image database into k clusters, $C = \{C_1, C_2 ... C_k\}$, where each cluster $C_s = \{I_1^{(s)}, I_2^{(s)} ... I_{|C_s|}^{(s)}\}$ is a subset of the images I. We first compute the global features for each image in the database with the convolutional neural network using the pre-trained model of VGG16 architecture on ImageNet[1]. Using global image features ensures that dissimilar images will be assigned to different clusters. This cannot be done with local features (such as corners or edges), since even dissimilar images may share such local features. This clustering will speed up the image-retrieval process: If we assume that the clusters are fully distinct, i.e., they do not share any image features, the retrieval process could be restricted to the cluster that is most similar to the query image, and the result would be the same as if checking against all clusters. Unfortunately, fully feature-distinct clusters are unrealistic. For real-world image databases, there will always be some overlap between the features of individual clusters. However, by minimizing the feature overlap between the clusters and maximizing the feature overlap within each cluster, one can reach a configuration that is closest to the ideal case of feature-distinct clusters. More formally, we have to optimize the two following functions:

$$\begin{cases} \arg\min_{C} \sum_{i=1}^{k} \sum_{j=1}^{k} Sim(\mathcal{F}(C_i), \mathcal{F}(C_j)), i \neq j \\ \vee \\ \arg\max_{C} \sum_{s=1}^{|C|} Sim(\mathcal{F}(I_i^{(s)}), \mathcal{F}(I_j^{(s)})), \forall (i,j) \in [1..|C_s|]^2, , i \neq j. \end{cases} \quad (1)$$

Note that $\mathcal{F}(C_i)$ is the set of features of the images of the cluster C_i, $\mathcal{F}(I_i)$ is the set of features of the image I_i, and $Sim(\mathcal{F}_1, \mathcal{F}_2)$ is the similarity measure between two sets of features $\mathcal{F}_1$ and $\mathcal{F}_2$.

[1] http://www.image-net.org/

3.1.2 Decomposition operators

In the following, we define the main operators used in the decomposition process.

- **Global feature extractor.** We used VGG16 to extract the global features of the images. It is composed of several convolution and max pooling layers followed by the rectified linear activation function (ReLu), and it ends up with a fully connected layer and Softmax activation function. Extracting the global features from a fully connected layer generates a vector of 4,096 features, which is considered to be insufficient for computer vision applications. Therefore, the features are extracted from the last max pooling layer by excluding the fully connected layer. This results in a vector of 25,088 features.
- **Distance computation.** The distance between two images I_i and I_j is defined as

$$D(I_i, I_j) = \sum_{l=1}^{|\mathcal{F}(I_i)|} |\mathcal{F}(I_i^l) - \mathcal{F}(I_j^l)|, \tag{2}$$

  where $\mathcal{F}(I_j^l)$ is the l^{th} feature of the image I_i.
- **Shortest distance.** Let us consider an image I_i, the set of clusters C, and let $\mu(C)$ be the set of centroids of C. We define $D_{min}(I_i, C)$ to the shortest distance between the image I_i and the centroids of the clusters in C, which is given by

$$D_{min}(I_i, C) = \min\{D(I_i, \mu_j) | \mu_j \in \mu(C)\}. \tag{3}$$

- **Centroids updating.** Let us consider the set of images of the cluster $C_i = \{I_1^{(i)}, I_2^{(i)}, ..., I_{|C_i|}^{(i)}\}$. The aim is to find a centroid of this set which is also an image, and we define μ_i to be the centroid of cluster C_i. The features of μ_i will be the average of all feature values of the images within cluster C_i. The j^{th} feature of μ_i, noted $\mathcal{F}_j(\mu_i)$ is determined as

$$\mathcal{F}_j(\mu_i) = \frac{\sum_{l=1}^{|C_i|} \mathcal{F}_j(I_l^{(i)})}{|C_i|}, \tag{4}$$

  where $\mathcal{F}_j(I_l^{(i)})$ is the j^{th} feature of the image $I_l^{(i)}$. Note that Eq. 4 will be applied for all 25,088 elements in the feature vector of μ_i.

3.1.3 Algorithms

In the following, we propose different clustering algorithms in order to optimize the functions reported in Eq. 1, which minimize the number of shared features among clusters, and maximize the number of shared features inside each cluster.

- **kmeans for image decomposition.** kmeans for image decomposition aims to maximize the function

$$J = \sum_{i=1}^{k} \sum_{I_j^{(i)} \in C_i} D(\mathcal{F}(I_j^{(i)}), \mu_i)^2. \tag{5}$$

  First, the images are assigned randomly to the k clusters and a centroid is computed for each cluster. Then, every image is assigned to a cluster whose centroid is the closest to that image. These two steps are repeated until there is no further assignment of the images to the clusters.
- **kmeans++ for image decomposition.** The main drawback of the kmeans algorithm is the centroid updating. In order to solve this issue, kmeans++ for image decomposition is developed. It aims to explore the centroid space, and accurately update the centers of the image clusters. The shortest distance for each image is first determined. For the centroid updating, the clusters are created recursively, where at each iteration t, the image I_i will be assigned to the cluster C_t with probability

$$P(I_i, C_t) = \frac{D(I_i, \mu_t)^2}{\sum_{I_j \in I} D(I_j, m_t)^2}. \tag{6}$$

  This process will be repeated until all images are assigned to the cluster clusters. Except for centroid updating, the same kmeans process is applied.
- **Mini batch kmeans for image decomposition.** It is a variant of the kmeans algorithm which uses mini-batches in order to reduce the decomposition time. The mini-batches are subsets of the image database, and are randomly generated. The union of all mini-batches should be equal to the entire database. The use of mini-batches allows the algorithm to faster converge to the local optimum.

3.2 Vocabulary forest construction

3.2.1 Principle

After decomposing into clusters, we create a new structure called the vocabulary forest. It is an extended representation of the vocabulary tree used by the BoW-based solutions. Thus, the first two steps of the BoW pipeline (feature extraction and vocabulary tree construction) are applied on each cluster C_s, which results in a set of k vocabulary trees, where each vocabulary tree voc_s contains information related to the cluster C_s. We define the vocabulary forest vForest by a tuple $< voc, g >$, where voc is the set of k vocabulary trees, and g is the set of centroids of the clusters in C. In the following, we describe the main operators of this step, which are the SIFT extractor and construction of the vocabulary tree of each cluster.

3.2.2 SIFT extractor

This step aims to determine the local features of each image in each cluster. The SIFT extractor is a well-known algorithm to identify the most relevant features in a given image. It is decomposed into four main stages:

1. Feature point detection: The feature points are identified based on difference on Gaussian function of the image. Thus, the Gaussian of the image is first computed. Each point is then computed with its eight neighbours. The local minima and maxima will be considered as a set of keypoint candidates.
2. Feature point localization: The set of keypoint candidates are refined to derive the correct localization of the keypoints. The set of keypoints are extended to sub-pixel using the Taylor expansion.
3. Orientation assignment: For each keypoint detected in the previous stage, its 16x16 neighbours are selected. The edge orientation of each neighbour is calculated, where the angle histogram is deduced using the histogram of oriented gradients.
4. Feature descriptor generation: This stage aims to generate the descriptor of each keypoint, which consists of 128 features. The orientation histogram is calculated based on the histogram determined in the previous stage.

At the end of this step, the pairs of keypoints and descriptors are calculated for every image in each cluster. Each keypoint is characterized by its pixel coordinates, where the descriptor is composed by 128 features representing the different orientation histograms of its 16x16 neighbours. Figure 3 shows the visualization of the features using one image from the Offshore dataset, which will be used in the experimentation part.

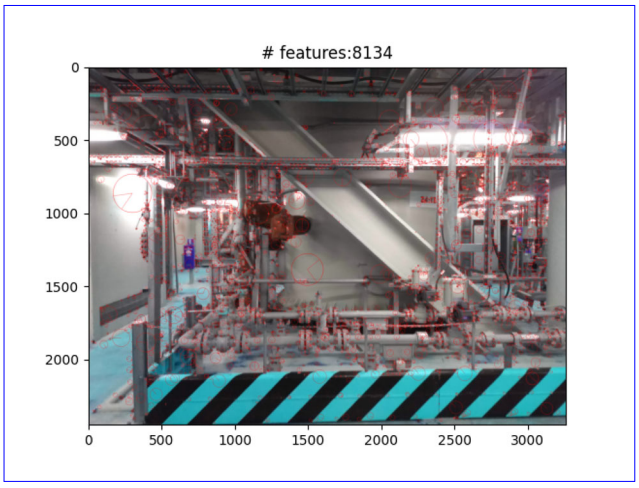

Fig. 3 SIFT Features Visualization

3.2.3 Building vocabulary trees

The collection of features extracted from a given cluster is used to compute the visual words. Here, for each cluster, the visual words from these features are determined using the hierarchical kmeans algorithm, where each center will be considered as one visual word. This results in one vocabulary tree for each cluster of images. Afterwards, each image in every cluster is represented by a frequency histogram of words, computed by exploring the vocabulary tree associated with the cluster. Different metrics can be used to determine the frequency histogram. In this research work, TFIDF (Term frequency Inverse Document Frequency) [28] is adopted to determine the frequency histogram of the image I^i assigned to the cluster C_i, and is defined as

$$TFIDF(I^i, w) = TF(I^i, w) \times IDF(w, C_i), \tag{7}$$

where

$$TF(I^i, w) = \frac{f_{I^i,w}}{\sum_{w' \in I^i} f_{I^i,w'}}, \tag{8}$$

$$IDF(w, C_i) = log\left(\frac{|C_i|}{|\{I^i \in C_i / w \in I^i\}|}\right), \tag{9}$$

and where $f_{I^i,w}$ is the frequency of the visual word w in the image I^i.

3.3 Searching process

The online processing has the goal of finding the relevant images to the given input image by querying the vForest structure. The features of the query image are extracted using both CNN and SIFT algorithms, described in Sections 3.1.2 and 3.2.2, respectively. The former is used to determine the similarity between each centroid in the vForest structure and the CNN features of the query image. The search will then be limited to the most similar clusters using the SIFT features of the query image. Different strategies may be used to explore the clusters:

1. **1-Nearest cluster neighbour.** In this strategy, we only explore the nearest cluster to the query image. To do that, we compute the similarity between the query image and each centroid, and we choose the cluster with highest similarity score.
2. **l-Nearest clusters neighbours.** In this strategy, we explore the l nearest clusters of the query image. The search starts by exploring the images of the most similar cluster to the query image, then the second most similar cluster to the query image, and so on until the l^{th} similar cluster of the query image.

After selecting the cluster(s) to be used in the searching process, the corresponding vocabulary tree(s) is explored. To find the relevant images to the image query from a given cluster, the corresponding vocabulary tree is used to compute the visual words from SIFT features of the query image. The score function between the image query and each image I^i in C_i is then calculated, which is defined using the TFIDF value of all visual words belonging to both the image query and the image I^i as

$$Score(I_q, I^i) = \sum_{w \in (I_q \vee I^i)} TFIDF(I^i, w). \quad (10)$$

The top relevant images are those with highest score values and are returned to the user. When considering multiple vocabulary trees, the same process is applied to each vocabulary tree and the most relevant images from all selected vocabulary trees are considered as relevant.

3.4 Theoretical complexity

Algorithm 1 presents the pseudo-code of DCNN-vForest framework. The decomposition and the vocabulary forest construction steps are the most time consuming tasks, however, both steps are performed only once, independently from the number of image queries. The image search on the other hand is executed for each query image, and it's execution time is crucial when used in a navigation setting. The theoretical complexity of the searching process depends on the number of clusters visited during the search process, the number of images, the number of words and the number of clusters, noted l, n, w and k, respectively. The theoretical complexity of BoW-based solutions for retrieving one query is $O(n \times w \times log(w))$. If we assume the decomposition step generates clusters with approximately the same number of images each, then the DCNN-vForest applied the BoW approach for each visited cluster with size $\frac{n}{k}$. Therefore, the theoretical complexity of DCNN-vForest for retrieving one query is $O(l \times \frac{n}{k} \times w \times log(w))$. Ideally, only a single cluster is explored, which costs $O(\frac{n}{k} \times w \times log(w))$, and in the worst case, all clusters are explored, which costs $O(n \times w \times log(w))$. From these theoretical analysis, we can argue that the lower bound of DCNN-vForest complexity time is $O(\frac{n}{k} \times w \times log(w))$, which is k times faster than BoW-based solutions, and the upper bound of DCNN-vForest complexity time is $O(n \times w \times log(w))$, which is equal to the complexity of BoW-based solutions.

Algorithm 1 DCNN-vForest.

Input: $I = \{I_1, I_2, \ldots, I_n\}$: the set of images. q: the query image.
Output: I': the set of the relevant images of the image user query q.
**********Decomposition**********
$C \leftarrow Decomposition(CNN(I))$
$vForest \leftarrow \emptyset$
for s=1 to k **do**
    $voc_s \leftarrow VocabularyTreeConstruction(C_s)$
    $g_s \leftarrow centroid(C_s)$
    $vForest \leftarrow vForest \cup \{voc_s, g_s\}$
end for
**********Searching**********
for s=1 to k **do**
    $C' \leftarrow Similarity(q, vForest)$
end for
$I' \leftarrow Searching(C', q)$
return I'

3.5 Illustration

Figure 4 illustrates the DCNN-vForest construction on images from the offshore dataset. On the top of the figure, the images are decomposed into clusters of visually similar images, using the global features extracted using the convolution neural network. In this context, each cluster presents one room or location on the offshore oil platform. For instance, the first cluster represents a location with pumps covered by blue isolation, the second cluster represents locations with containers, and the third cluster represents a hall area of the offshore oil platform. The middle row illustrates the vocabulary trees, where each tree consists of visual words generated from features representative for the corresponding cluster of images. The bottom shows the word vector for two images from the cluster with images containing pumps covered by blue isolation.

4 Results

Several experiments were conducted to evaluate the performance of the proposed DCNN-vForest framework.

Four datasets were used in the experiments. Three of them are well-known datasets, widely used for visual navigation problem, and the last one is captured from an offshore oil platform. The latter contains images representative for industrial platforms and is provided by

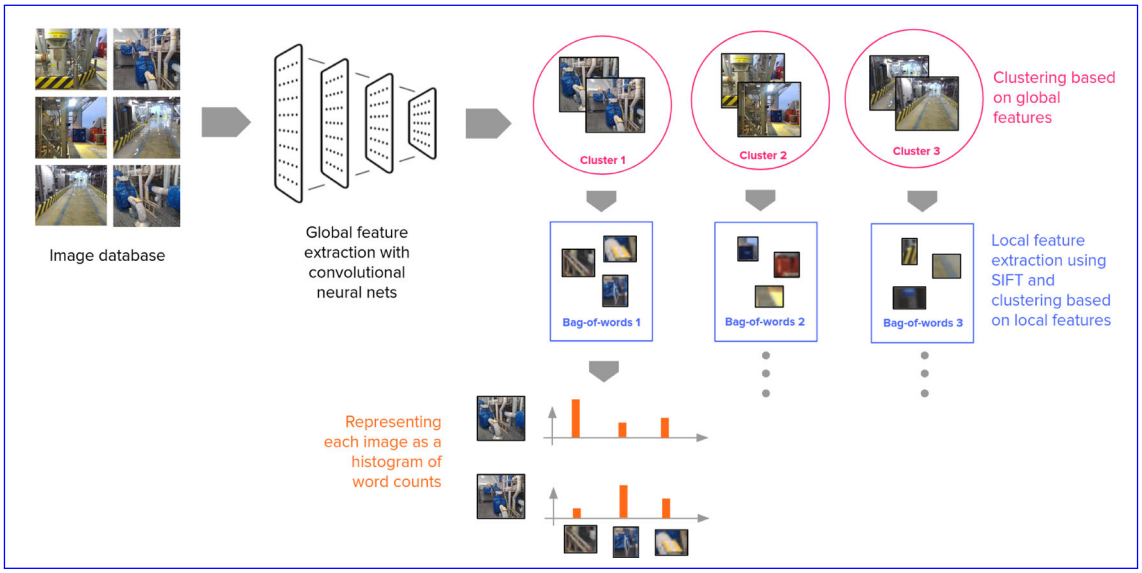

Fig. 4 DCNN-vForest Illustration

our industrial partners. The detailed description of the four datasets used in the experiments is given below:

1. **Offshore.** Dataset from an industrial installation. It contains 1,153 images. Each image is associated by a tag representing its location on the installation.
2. **Kitti.** Well-known dataset typically used in place recognition problems. The data is collected by up to 15 cars equipped with two high-resolution color and grayscale video cameras. The ground truth is provided by the GPS localization system of the cars [23]. We used 3GB of data with 3,025 images. The data is public and may be retrieved on http://www.cvlibs.net/datasets/kitti/index.php.
3. **ZUMAV.** Dataset collected using a camera equipped on a Micro Aerial Vehicle (MAV) flying over urban streets at low altitudes (5-15 meters above the ground) [33]. We used 1.37GB of data with 4,020 images. The data is public and can be retrieved from http://rpg.ifi.uzh.ch/zurichmavdataset.html.
4. **Indoor.** One of the top ten datasets used in image classification problems. It contains 67 different categories for indoor scene recognition. The number of images varies across categories, but there are at least 100 images per category. We used 1GB of the data with 3908 images divided into different categories. This dataset is public and can be retrieved from https://www.kaggle.com/itsahmad/indoor-scenes-cvpr-2019.

The characteristics of the four databases are shown in Table 1.

Each dataset is divided into two disjoints subsets, one for training, and another for testing. The training data are used to build the vocabulary forest, and the test data are considered as query images. The evaluation of the proposed framework is performed in two main steps:

1. Evaluation of decomposition step: The evaluation of the decomposition step is performed using distortion Elbow score [7]. It is the most common metric used for determining the optimal number of clusters. It is computed as the sum of the squared distance between each image and its closest centroid, which is formally defined as,

$$\text{Elbow}(C) = \sum_{i=1}^{|C|} \sum_{j=1}^{|C_i|} D(I_j^{(i)}, \mu_i) \tag{11}$$

It is also important to create balance clusters which will be addressed later in the image search step. Therefore, we propose a new measure to evaluate the decomposition step, based on the number of images in each cluster. The aim is to obtain similar number of images per cluster. It is determined by the average number of images per cluster, divided by the by the number of images of the biggest cluster. The result is between 0, and 1, where the perfectly balanced

Table 1 Data description

Database	# Images	Resolution	Size in GB	# Classes/Places
Offshore	1,153	3,264 X 2,448	0.55	6
Kitti	3,025	1,392 X 512	3.00	7
ZUMAV	4,020	1,920 x 1,080	1.37	4
Indoor	3,908	247 X 325	1.00	67

configuration will be 1. It is formally defined as,

$$\text{Balance}(C) = \frac{|I|}{|C| \times max(C)}, \quad (12)$$

where max(C) is the number of images of the biggest cluster in C.

2. Evaluation of Searching Step: Evaluation of Searching Step: Different evaluation criteria are used depending on whether the ground truth of the dataset is in the form of from classification or localisation.

In the case of classification, the well-known mAP (mean Average Precision) [13], which is defined as:

$$mAP = \frac{\sum_{i=0}^{r} AvgP(i)}{n}, \quad (13)$$

where r is the number of images to be retrieved, n is the number of all image queries, and $AvgP(i)$ is the average precision while considering the first i ranked images. The mAP criterion is chosen because it scores the fraction of selected images that are correct.

The second evaluation is established where the ground truth a position associated to each images. In this context, we propose a new measure to evaluate the results. It aims to calculate the ratio of accepted images. An image is accepted if its position is close to the position of the query image. The purpose is maximize the following function:

$$\phi(\mathcal{T}, \mathcal{R}) = \frac{\sum_{i=1}^{|\mathcal{T}|} \phi_i(\mathcal{T}_i, \mathcal{R}^{(i)})}{|\mathcal{T}|}, \quad (14)$$

where

$$\phi_i(\mathcal{T}_i, \mathcal{R}^{(i)}) = \frac{\sum_{j=1}^{n} \phi_{ij}(\mathcal{T}_i, \mathcal{R}_j^{(i)})}{n}, \quad (15)$$

and

$$\phi_{ij}(\mathcal{T}_i, \mathcal{R}_j^{(i)}) = \begin{cases} 1, & \text{if } (1/Pos(\mathcal{T}_i, \mathcal{R}_j^{(i)})) \leq d_{max} \\ 0, & \text{otherwise} \end{cases} \quad (16)$$

Note that,
$\mathcal{T}$: The set of test images.
$\mathcal{R}$: The set of sets of the most similar images to all images in $\mathcal{T}$.
$\mathcal{R}^{(i)}$: The set of most similar images to the query image $\mathcal{T}_i$.
$\mathcal{R}_j^{(i)}$: The j^{th} similar image to the query image $\mathcal{T}_i$.
$Pos(\mathcal{T}_i, \mathcal{R}_j^{(i)})$ is the difference between the position of the image $\mathcal{T}_i$, and the position of the image $\mathcal{R}_j^{(i)}$.
d_{max} is a scenario specific threshold.

All implementations are executed on a computer with a i7 CPU, coupled by a GeForce GTX 1070 GPU. We used Python 3.7.4, and scikit-learn library for building, and evaluating the clusters of images. The tuning of the DCNN-vForest parameters are first explained. The best configuration of DCNN-vForest is then compared to the BoW-based image retrieval based solutions.

4.1 Parameter setting

The aim of this experiment is to tune the parameters of the DCNN-vForest framework. In the following, the parameters setting of each step of DCNN-vForest is studied.

The quality of the decomposition is measured by evaluating both the Elbow, and the balance functions, as shown in Fig. 5. The three decomposition algorithms kmeans, kmeans++, and mini-batch kmeans are used for comparison. However, for each execution, only one algorithm of the three is selected. Starting by Elbow which is a visual metric to estimate the optimal number of clusters. It involves running the decomposition algorithm multiple times with an increasing the number of clusters (from 5 to 50) and then plotting the distortion score. As the number of clusters increases, the distortion score is decreasing. This is because the images will be closer to the centroids they are assigned to. The idea behind the Elbow metric is to identify the value of the number of clusters where the score begins to decrease most rapidly before the curve reaches a plateau. Therefore, the optimal number of clusters is selected just before reaching the plateau. For instance, the optimal number of clusters on Offshore dataset is 35 for kmeans, 25 for kmeans++, and 35 for mini-batch kmeans. We remove the mini-batch kmeans configurations for the Indoor dataset, because the distortion score is not close to monotone, and is therefore unreliable. An explanation of these result for the Indoor dataset is that the clusters obtained are unbalanced (many images are associated to a single cluster, leaving other clusters almost empty). Indeed, the balance score is also important in the decomposition results. Obtaining clusters of similar size is crucial for the searching step. Consequently, we select the configurations with high balance score. The selected configurations for each dataset is as follows: Offshore (kmeans++ with 25 clusters), Kitti (kmeans with 25 clusters), Indoor (kmeans with 15 clusters), and ZUMAV (mini-batch kmeans with 40 clusters). Table 2 shows the best configurations of the DCNN-vForest selected for each dataset.

The next experiment is to tune the parameters of the image search step. We varied the strategy used in retrieving the relevant images. Thus, we varied the percentage of the number of visited clusters from 10% to 100%, and we computed the runtime, and the accuracy of each configuration. The results are reported in Fig. 6. According

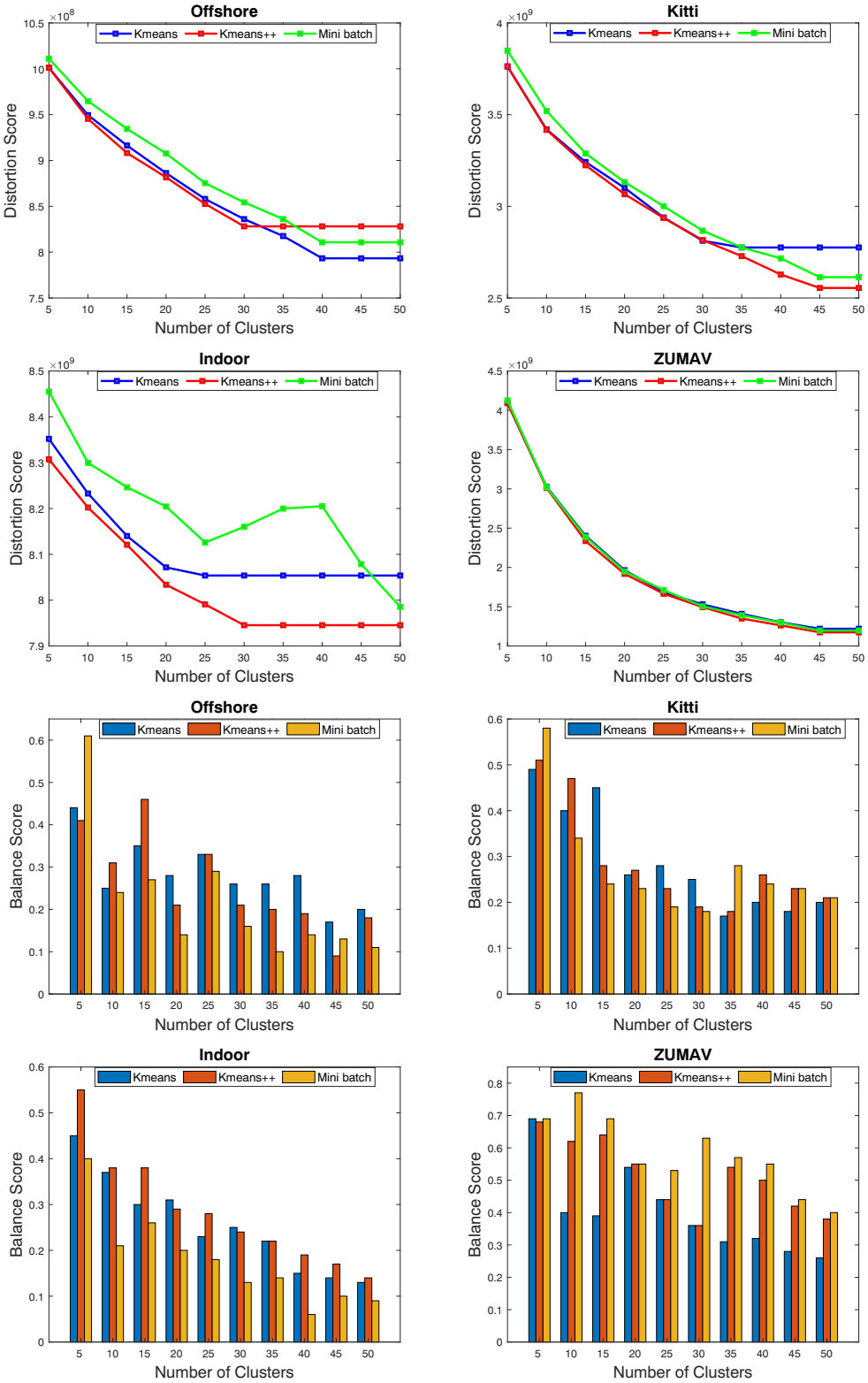

Fig. 5 Elbow and balance scores of the decomposition step with varying the number of clusters

to the results, we can remark that the accuracy of DCNN-vForest increases with increasing the number of visited clusters, until it stabilizes in a given number of visited clusters. In some databases, few number of clusters are needed to be explored to reach high accuracy as the case of Kitti, however, in some databases, high number of clusters are needed to be explored to reach high accuracy, as the case of ZUMAV. In addition, the runtime increases

Table 2 Best configurations of DCNN-vForest

Database	Best configuration
Offshore	kmeans++ with 25 clusters
Kitti	kmeans with 25 clusters
ZUMAV	mini-batch kmeans with 40 clusters
Indoor	kmeans with 15 clusters

with increasing number of visited clusters. Therefore, the number of visited clusters is selected by choosing the lowest number which gives higher accuracy value. The best values of the DCNN-vForest for each dataset is given as follows: Offshore (kmeans++ with 25 clusters, and 12 visited clusters), Kitti (kmeans with 25 clusters, and 10 visited clusters), Indoor (kmeans with 15 clusters, and 7 visited clusters), and ZUMAV (mini-batch kmeans with 40 clusters,and 16 visited clusters).

4.2 DCNN-vForest vs state-of-the-art image search algorithms

This section studies the performance of vForest compared to BoW [3], HDE [40], and SSMV-BoW [24]. Throughout this section, the parameters for clustering are kept fixed and the number of words per vocabulary tree varies from 100 to 1,000. The number of words are deliberately kept low, to reach an acceptable runtime performance in the image search process. However, each cluster of images is expected to consist of similar images, which is likely to result in good accuracy, even with the relatively small number of words.

The construction time of the vocabulary trees are shown in Fig. 7. It shows that the construction time is reduced with a factor of three in the place recognition cases and by 10% in the classification case. This difference of performance has two contributing factors: in the classification case the image database is split in fewer clusters, and therefore a larger portion of the images in each cluster, furthermore, the images in each cluster are expected to have more common similarities in place recognition problem compared to the image database for classification.

The runtime performance of the query is presented in Fig. 8. As can be seen, the query time is lower for DCNN-vForest than the other solutions (BoW, HDE, and SSMV-BoW) indicating that the ability to ignore large portions of the image databases saves more time than the overhead cost of evaluating the neural net and select the most relevant clusters. Similarly, Fig. 9 shows that the accuracy is also improved with the DCNN-vForest algorithm, for all evaluated datasets.

Figure 10 shows the retrieved images for representative query images, one from each dataset. The results reveal that the DCNN-vForest outperforms the BoW in terms of evaluation score for all cases. Furthermore, visual inspection shows that the top three relevant images are relevant to the query images in the localisation datasets, while this is not the case for the classical BoW algorithm. This is especially true for the Offshore dataset, where the BoW algorithm fails to find any relevant images. This is explained by the fact that the DCNN-vForest splits the data into clusters of visually similar images, allowing the visual words to be more representative to the images in each cluster. These results confirm the applicability of DCNN-vForest in dealing industrial offshore data for autonomous navigation systems, which is missing on state-of-the-art BoW solutions. These results confirm the applicability of DCNN-vForest when dealing with industrial offshore data for autonomous navigation systems, which is missing on state-of-the-art BoW solutions.

5 Discussions and future directions

This section discusses the main findings from the experiments using the DCNN-vForest method on the place recognition problem.

1. The first finding of this study is that the query time improvements achieved by only considering images with similar features exceeds the overhead of evaluating

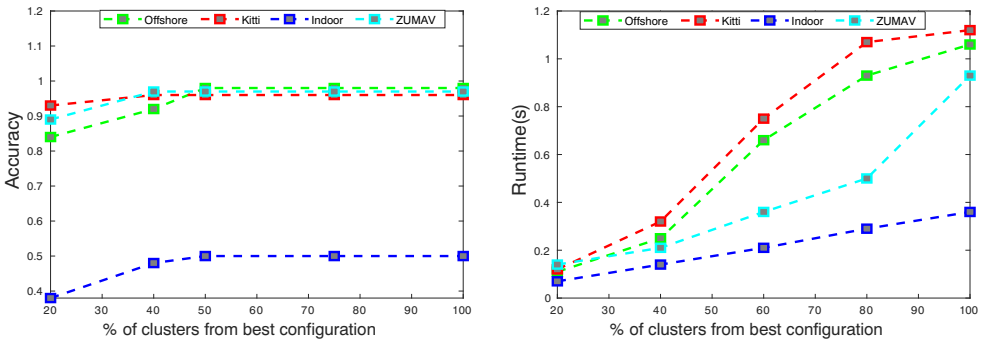

Fig. 6 Accuracy and Runtime of the image search of the DCNN-vForest with varying the number of visited clusters

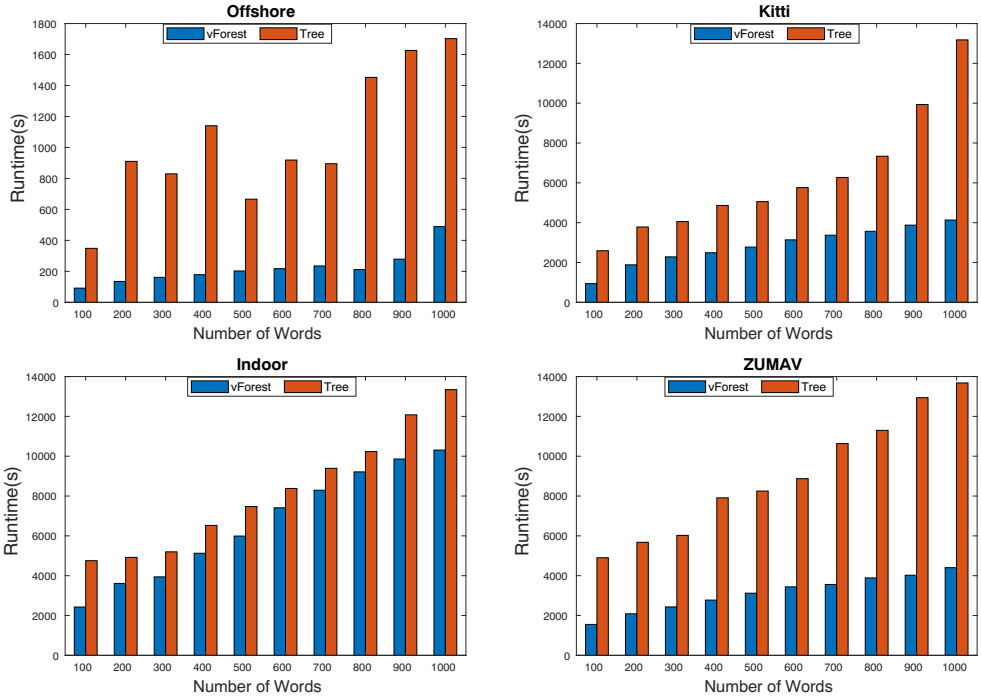

Fig. 7 Comparison of runtime of vocabulary tree, and vocabulary forest construction with different number of words

the neural net and choosing the most appropriate cluster(s). This leads to a considerable reduction in query time for all datasets. Furthermore, the runtime performance is dependent on the uniformity of the cluster sizes.

2. The second finding is that the proposed framework improves the accuracy of the BoW algorithm. This is mainly due the similarity of the images in each cluster, which makes is easier to find common features (words) describing the images. Furthermore, feature vector determined by the CNN contains information on a global level, in contrast to the SIFT feature extractor that operates on a local level only.
3. The third finding of this study is that the choice of decomposition algorithm and number of clusters are crucial for the performance of the proposed

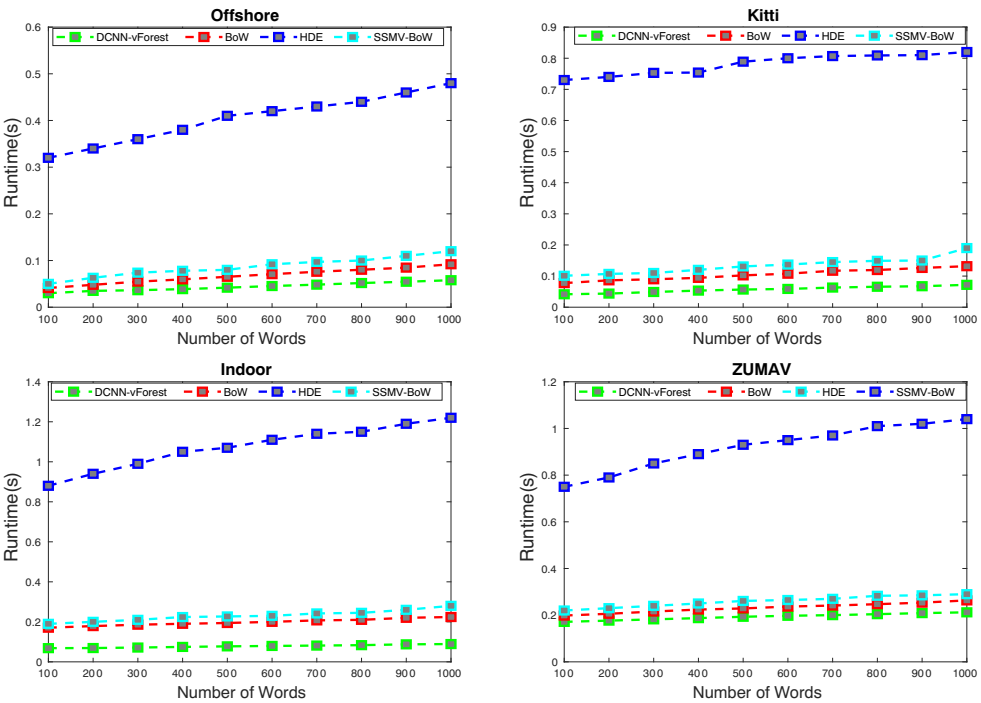

Fig. 8 Runtime of the image search of the DCNN-vForest, and State-of-the-art Image Search Algorithms

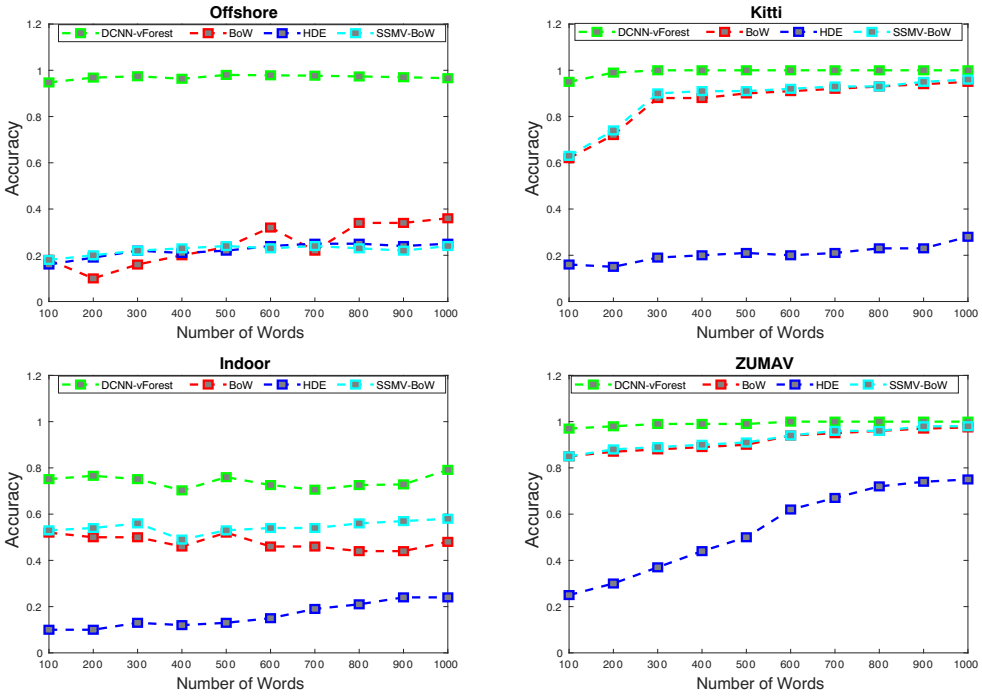

Fig. 9 DCNN-vForest Vs State-of-the-art Image Search Algorithms: Accuracy

method. According to our experiments, and analysis on different datasets, there is a high relation between the data correlation and the parameters setting of the decomposition step. More correlated datasets can benefit from a higher number of clusters, separating the data more accurately.

4. The last finding of this study is that the BoW components influence on the results of the image search. For instance, large number of visual words used in building the vocabulary tree of each cluster increase the accuracy performance, however it needs considerable amount of time and memory for processing and storing the vocabulary trees.

The results presented in this paper is promising, and opens up for further studies on:

1. Decomposition algorithms that creating more balanced clusters and separates the dataset better. In this research work, three kmeans-based algorithms are explored to decompose the image database into clusters of similar images, an interesting direction for future work is to study the adaptation of other decomposition algorithms such as, density-based algorithms, hierarchical algorithms, and fuzzy-based decomposition algorithms, or methods from other fields such as entity resolution and/or record linkage.
2. Auto tuning of parameters, including number of clusters and number of words in each vocabulary tree and how many clusters to visit per query. In this paper, a brute force parameter sweep is performed for each dataset, which is highly time consuming. One possibility is to apply meta learning to learn the different parameters of the DCNN-vForest. The learning stage is done from properties extracted from the training image databases such as the number of images, the number of pixels, the image features...). A challenge here is to design the training data and to learn the parameters required.
3. Applying the DCNN-vForest method in real life applications to further validate its performance and applicability. This will provide further insight in the performance of the proposed method. One promising application visual localization of robots in industrial environments. DCNN-vForest is promising here, where one can access a large image data base including similar images taken of the same location from different views.

6 Conclusion

This paper propose a new algorithm for using image retrieval to efficiently solve the challenge of place recognition. The goal is to determine the location from where the query image was taken by using the location of the most similar image. The proposed method is a hybrid approach for image retrieval by combining deep learning and decomposition frameworks. It integrates the convolution neural network to extract the relevant features of the image database. The extracted features are used to divide the whole image database into clusters, each of which contains similar images. A vocabulary tree is then

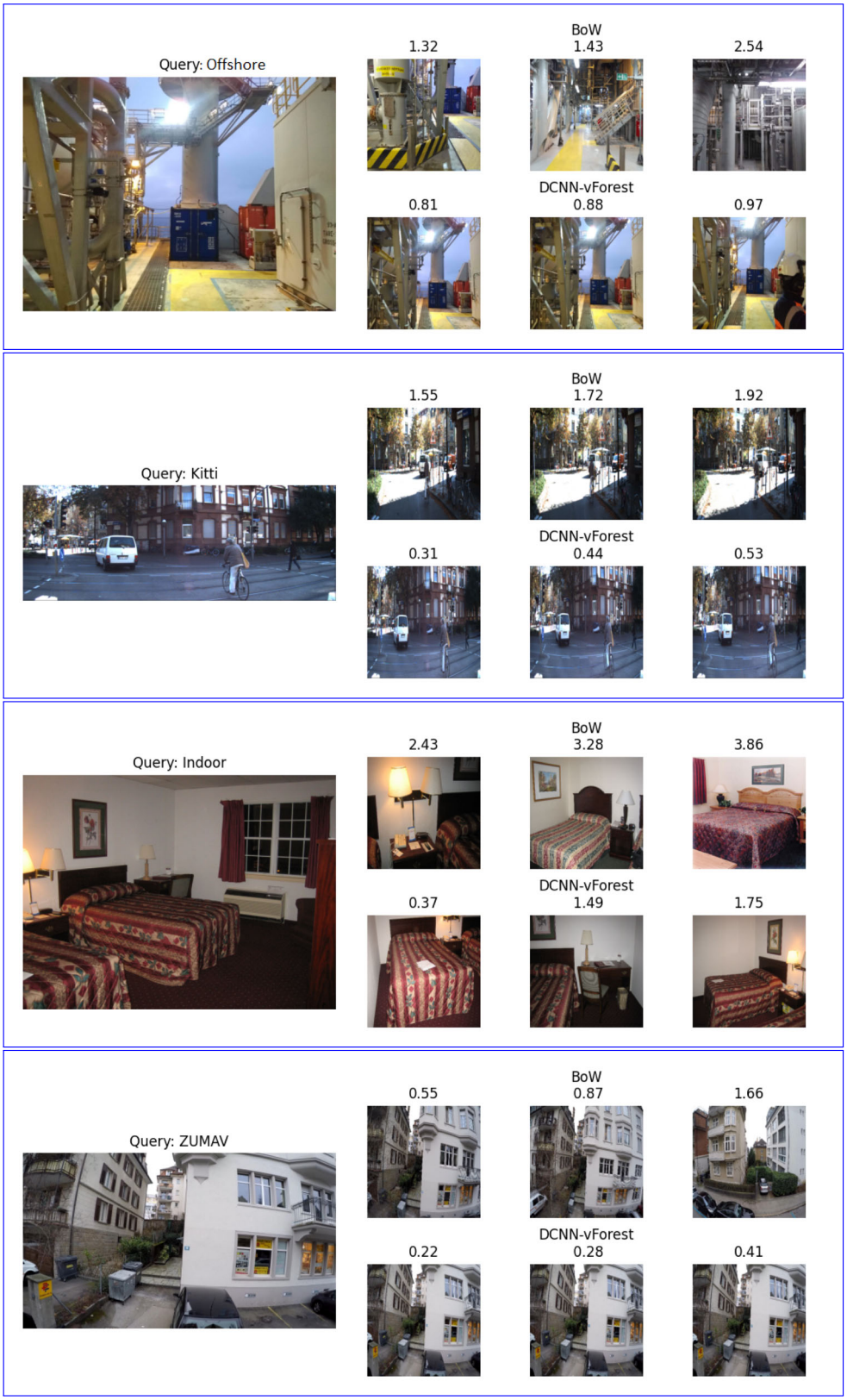

Fig. 10 DCNN-vForest Vs. BoW Illustration on Real Case Scenarios

created for each cluster to build a vocabulary forest. In the searching process, only the relevant clusters of images are explored instead of scanning the whole image database. Furthermore, since each cluster contains similar images, acceptable accuracy can be achieved with a small number of words compared to what is needed when using the BoW algorithm directly. Combined, these two factors allows the DCNN-vForest algorithm to be used in industrial settings where state-of-the-art BoW implementations are too time consuming to be used in real-time applications. The method is validated on well known and openly available image databases, two for place recognition in urban environments and one for classification in indoor environments. In each case, the performance is compared with a state-of-the-art BoW implementation. The same BoW implementation is used to create and evaluate each tree in the DCNN-vForest implementation. In all cases, both runtime performance and accuracy is significantly improved when all parameters are equal. The main goal of the work is to provide support for navigation in an industrial offshore site where GPS is not available, and BoW did not yield sufficient accuracy within the time budget. The initial results show good potential for using the DCNN-vForest also in this case.

Acknowledgements This paper is supported by the Norwegian Research Council funded project Advanced 3D visualization and AR for industrial operations. We would like to thank all project partners, including Aker BP, Lundin, Aker Solutions and Kværner for sharing ideas and data.

Funding Open access funding provided by SINTEF AS.

References

1. Anwar A, Raychowdhury A (2020) Autonomous navigation via deep reinforcement learning for resource constraint edge nodes using transfer learning. IEEE Access 8:26,549–26,560
2. Arandjelovic R, Gronat P, Torii A, Pajdla T, Sivic J (2016) Netvlad: Cnn architecture for weakly supervised place recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5297–5307
3. Bai Y, Yu W, Xiao T, Xu C, Yang K, Ma WY, Zhao T (2014) Bag-of-words based deep neural network for image retrieval. In: Proceedings of the 22nd ACM international conference on Multimedia, pp 229–232
4. Ban X, Lv X, Chen J (2009) Color image retrieval and classification using fuzzy similarity measure and fuzzy clustering method. In: Proceedings of the 48h IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference. IEEE, pp 7777–7782
5. Baumgartl H, Buettner R (2020) Development of a highly precise place recognition module for effective human-robot interactions in changing lighting and viewpoint conditions. In: Proceedings of the 53rd Hawaii International Conference on System Sciences
6. Belhadi A, Djenouri Y, Lin JCW, Zhang C, Cano A (2020) Exploring pattern mining algorithms for hashtag retrieval problem. IEEE Access 8:10,569–10,583
7. Bholowalia P, Kumar A (2014) Ebk-means: A clustering technique based on elbow method and k-means in wsn. Int J Comput Appl 105(9)
8. Cao F, Yan F, Wang S, Zhuang Y, Wang W (2020) Season-invariant and viewpoint-tolerant lidar place recognition in gps denied environments. IEEE Transactions on Industrial Electronics
9. Cao Y, Long M, Wang J, Liu S (2017) Deep visual-semantic quantization for efficient image retrieval. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1328–1337
10. Carrio A, Tordesillas J, Vemprala S, Saripalli S, Campoy P, How JP (2020) Onboard detection and localization of drones using depth maps. IEEE Access 8:30,480–30,490
11. Chancan M, Hernandez-Nunez L, Narendra A, Barron AB, Milford M (2020) A hybrid compact neural architecture for visual place recognition. IEEE Robot Autom Lett 5(2):993–1000
12. Choi J, Son MG, Lee YY, Lee KH, Park J, Yeo CH, Park J, Choi S, Kim WD, Kang TW et al (2020) Position-based augmented reality platform for aiding construction and inspection of offshore plants. Vis Comput 36(10):2039–2049
13. Cormack GV, Lynam TR (2006) Statistical precision of information retrieval evaluation. In: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pp 533–540
14. David A (2007) Vassilvitskii s.: K-means++: The advantages of careful seeding. In: 18Th annual ACM-SIAM symposium on discrete algorithms (SODA), New orleans, pp 1027–1035
15. Dinh VQ, Munir F, Azam S, Yow KC, Jeon M (2020) Transfer learning for vehicle detection using two cameras with different focal lengths. Inf Sci 514:71–87
16. Djenouri Y, Belhadi A, Fournier-Viger P, Lin JCW (2018) Fast and effective cluster-based information retrieval using frequent closed itemsets. Inf Sci 453:154–167
17. Djenouri Y, Comuzzi M (2017) Combining apriori heuristic and bio-inspired algorithms for solving the frequent itemsets mining problem. Inf Sci 420:1–15
18. Djenouri Y, Hjelmervik J (2021) Hybrid decomposition convolution neural network and vocabulary forest for image retrieval. In: 25th International Conference on Pattern Recognition, pp in press. IEEE
19. Doan AD, Latif Y, Chin TJ, Liu Y, Do TT, Reid I (2019) Scalable place recognition under appearance change for autonomous driving. In: Proceedings of the IEEE International Conference on Computer Vision, pp 9319–9328
20. Eder M, Reip M, Steinbauer G (2021) Creating a robot localization monitor using particle filter and machine learning approaches. Appl Intell:1–15
21. Erra U, Senatore S (2011) Hand-draw sketching for image retrieval through fuzzy clustering techniques. In: SEBD, pp 413–420
22. Ferrarini B, Waheed M, Waheed S, Ehsan S, Milford M, McDonald-Maier K (2020) Exploring performance bounds of

visual place recognition using extended precision. IEEE Robot Autom Lett 5(2):1688–1695
23. Geiger A, Lenz P, Stiller C, Urtasun R (2013) Vision meets robotics: The kitti dataset. Int J Robot Res 32(11):1231–1237
24. Giveki D (2021) Scale-space multi-view bag of words for scene categorization. Multimed Tools Appl 80(1):1223–1245
25. Hong Z, Petillot Y, Lane D, Miao Y, Wang S (2019) Textplace: Visual place recognition and topological localization through reading scene texts. In: Proceedings of the IEEE International Conference on Computer Vision, pp 2861–2870
26. Husbands P, Shim Y, Garvie M, Dewar A, Domcsek N, Graham P, Knight J, Nowotny T, Philippides A (2021) Recent advances in evolutionary and bio-inspired adaptive robotics: Exploiting embodied dynamics. Appl Intell:1–30
27. Khaliq A, Ehsan S, Chen Z, Milford M, McDonald-Maier K (2019) A holistic visual place recognition approach using lightweight cnns for significant viewpoint and appearance changes. IEEE Transactions on Robotics
28. Kim D, Seo D, Cho S, Kang P (2019) Multi-co-training for document classification using various document representations: Tf–idf, lda, and doc2vec. Inf Sci 477:15–29
29. Lee DH, Chen KL, Liou KH, Liu CL, Liu JL (2021) Deep learning and control algorithms of direct perception for autonomous driving. Appl Intell 51(1):237–247
30. Liu X, Zhang S, Huang T, Tian Q (2019) E2bows: An end-to-end bag-of-words model via deep convolutional neural network for image retrieval. Neurocomputing
31. Lowry S, Sünderhauf N, Newman P, Leonard JJ, Cox D, Corke P, Milford M (2015) Visual place recognition: a survey. IEEE Trans Robot 32(1):1–19
32. MacQueen J et al (1967) Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp 281–297
33. Majdik AL, Till C, Scaramuzza D (2017) The zurich urban micro aerial vehicle dataset. Int J Robot Res 36(3):269–273
34. Mansouri SS, Kanellakis C, Kominiak D, Nikolakopoulos G (2020) Deploying mavs for autonomous navigation in dark underground mine environments. Robot Auton Syst 126(103): 472
35. Mao QC, Sun HM, Zuo LQ, Jia RS (2020) Finding every car: a traffic surveillance multi-scale vehicle object detection method. Appl Intell 50(10):3125–3136
36. de Queiroz Mendes R, Ribeiro EG, dos Santos Rosa N, Grassi Jr V (2021) On deep learning techniques to boost monocular depth estimation for autonomous navigation. Robot Auton Syst 136(103):701
37. Sculley D (2010) Web-scale k-means clustering. In: Proceedings of the 19th international conference on World wide web, pp 1177–1178
38. Seong H, Hyun J, Kim E (2020) Fosnet: an end-to-end trainable deep neural network for scene recognition. IEEE Access 8:82,066–82,077
39. Vysotska O, Stachniss C (2019) Effective visual place recognition using multi-sequence maps. IEEE Robot Autom Lett 4(2):1730–1736
40. Yang X, Gao X, Song B, Han B (2020) Hierarchical deep embedding for aurora image retrieval. IEEE Transactions on Cybernetics
41. Yu J, Zhu C, Zhang J, Huang Q, Tao D (2019) Spatial pyramid-enhanced netvlad with weighted triplet loss for place recognition. IEEE transactions on neural networks and learning systems
42. Zaffar M, Ehsan S, Milford M, McDonald-Maier K (2020) Cohog: a light-weight, compute-efficient, and training-free visual place recognition technique for changing environments. IEEE Robot Autom Lett 5(2):1835–1842
43. Zhan Z, Zhou G, Yang X (2020) A method of hierarchical image retrieval for real-time photogrammetry based on multiple features. IEEE Access
44. Zhang J, Peng Y, Yuan M (2018) Unsupervised generative adversarial cross-modal hashing. In: Thirty-second AAAI conference on artificial intelligence

Affiliations

Youcef Djenouri[1] · Johan Hatleskog[2] · Jon Hjelmervik[1] · Elias Bjorne[3] · Trygve Utstumo[3] · Milad Mobarhan[3]

Johan Hatleskog
johan.hatleskog@cognite.com

Jon Hjelmervik
jon.m.hjelmervik@sintef.no

Elias Bjorne
elias.bjorne@cognite.com

Trygve Utstumo
trygve.utstumo@cognite.com

Milad Mobarhan
milad.mobarhan@cognite.com

1 Mathematics and Cybernetics Department, SINTEF Digital, Oslo, Norway

2 Cognite AS, Oslo, Norway & Engineering Cybernetics Department, The Norwegian University of Science and Technology, Trondheim, Norway

3 Cognite As, Oslo, Norway