Hindawi
Journal of Sensors
Volume 2020, Article ID 8861296, 13 pages
https://doi.org/10.1155/2020/8861296

Research Article

TCM: Temporal Consistency Model for Head Detection in Complex Videos

Sultan Daud Khan,[1] **Ahmed B. Altamimi**,[2] **Mohib Ullah**,[3] **Habib Ullah**,[2] **and Faouzi Alaya Cheikh**[3]

[1]*Department of Computer Science, National University of Technology, Pakistan*
[2]*Department of Computer Science and Software Engineering, University of Ha'il, Saudi Arabia*
[3]*Department of Computer Science, Norwegian University of Science and Technology, Norway*

Correspondence should be addressed to Mohib Ullah; mohib.ullah@ntnu.no

Received 18 June 2020; Revised 10 November 2020; Accepted 27 November 2020; Published 16 December 2020

Academic Editor: Abdellah Touhafi

Head detection in real-world videos is a classical research problem in computer vision. Head detection in videos is challenging than in a single image due to many nuisances that are commonly observed in natural videos, including arbitrary poses, appearances, and scales. Generally, head detection is treated as a particular case of object detection in a single image. However, the performance of object detectors deteriorates in unconstrained videos. In this paper, we propose a temporal consistency model (TCM) to enhance the performance of a generic object detector by integrating spatial-temporal information that exists among subsequent frames of a particular video. Generally, our model takes detection from a generic detector as input and improves mean average precision (mAP) by recovering missed detection and suppressing false positives. We compare and evaluate the proposed framework on four challenging datasets, i.e., HollywoodHeads, Casablanca, BOSS, and PAMELA. Experimental evaluation shows that the performance is improved by employing the proposed TCM model. We demonstrate both qualitatively and quantitatively that our proposed framework obtains significant improvements over other methods.

1. Introduction

Pedestrian detection is gaining much attention from the research community. Pedestrian detection has numerous applications in the surveillance domain, such as tracking [1, 2], anomaly detection [3, 4], congestion detection [5, 6], and behavior analysis [7, 8]. Most of the existing methods rely on face and pedestrian detection for tracking, counting, and behavior analysis. While pedestrian and face detection algorithms have gained much popularity, the task of detecting people in complex scenes is still a challenging task. Face detectors rely on extracting facial features that cannot be extracted when the pedestrian turns his back to the camera. On the other hand, pedestrian detection relies on detecting the whole pedestrian, which is not possible due to a number of problems in an unconstrained video environment. With these limitations, face and pedestrian detection methods cannot be employed in complex scenes. Therefore, to detect pedestrians in complex scenes, the head is the only visible and reliable clue.

Although several efforts have been made in this direction [9–11], head detection in an unconstrained environment is still an open issue and has enough room for improvement. The goal of a good head detector is to detect heads in an image with a high precision-recall rate. To achieve this, the head detector must be invariant to scale, pose, and appearance variations. Figure 1 shows the results of a generic head detector that we trained for this particular task. Despite low image quality, variations in poses, scales, and appearances, the generic head detector performed well. However, it misses many detections and accumulates a considerable number of false positives.

Inherently, large capacity convolutional neural networks (CNNs) have translation invariance property and can handle

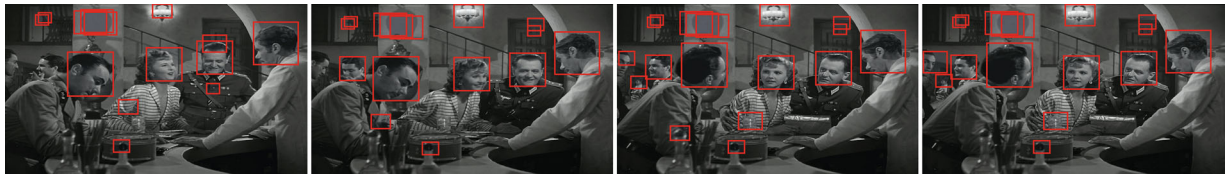

Figure 1: The results of a generic detector are shown using the Casablanca dataset [18]. Due to variations in human poses, scales, and appearances, a generic detector accumulates many false positives and missed detection in some frames.

pose and appearance variations in the image. This is due to the reason that CNNs have achieved tremendous success in object classification, detection, and segmentation tasks. Generally, most of the existing CNN-based methods deal head detection as a special case of object detection problem [12–16]. Vu et al. [16] proposed three models for person head detection, i.e., local model, global model, and joint model. Their local model is similar to the region-based convolutional neural network (R-CNN) [15] and uses the selective search (SS) [17] method for generating object proposals. The joint model exploits the contextual relationship among the pairs of detected heads. However, these models are computationally expensive. The region with CNN (R-CNN) method [12] generates 2000 region proposals using the SS method. Each proposal is then resized to fit the input of CNN and feed-forward to the network that extracts hierarchical features from the last convolution layer (5th layer of AlexNet). After extracting hierarchical features, SVM is trained to detect objects in the image. Faster R-CNN [15] proposed a two-stage network that uses a region proposal network (RPN) to generate high-quality object proposals. Faster R-CNN achieved superior results in various object detection tasks; however, the method suffers from computation complexity. You only look once (YOLO) [14] proposed a regression-based method that predicts the probabilities of classes by applying regression between pixels of an image and object bounding boxes. YOLO uses a limited number of candidate's regions for object detection. This strategy makes the YOLO faster than Faster R-CNN; however, the accuracy of YOLO is lower than Faster R-CNN. A single-shot detector (SSD) [13] produces a predefined number of object proposals by exploiting a fully convolutional neural network and predicts class probabilities of candidate bounding boxes. High confidence bounding boxes are retained, and low confidence bounding boxes are removed by applying the nonmaximum suppression method (NMS).

The above-mentioned object detection methods produce state-of-the-art results using static images; however, the performance of these detectors degrades when applied to videos.

This may attribute to the following two challenges:

(1) In videos, pedestrians pass through significant variations in scale, pose, texture, and illumination. These variations cause intraclass variability that degrades the performance of a detector. In most of the cases, the detector misses the detection or accumulates false positives that result in low mAP

(2) Object detectors based on CNN learn hierarchical features from raw images; however, they do not have the ability to leverage the temporal consistency among consecutive frames of a video

To address the above challenges, we proposed an approach that enhances the mAP of a generic detector. Figures 2 and 3 illustrate that the efficiency of the generic detector is improved by employing our proposed method that suppresses false positives and recovers missed detection. Generally, our approach has the following contributions:

(1) We propose the temporal consistency model (TCM) that leverages spatial and temporal information of video that exists between consecutive frames. An energy function is defined based on intensity and gradient consistency assumption that estimates the displacement vectors of all pixels of the image

(2) Using TCM, we enhance the performance of the generic detector by addressing two problems, i.e., suppressing false positives and recovering missed detection

(3) To suppress false positives, we propose an algorithm that uses the TCM model to leverage temporal information and to assign a confidence score to each detection

(4) In order to recover missed detection, a dual-mode tracking technique is adopted

(5) We use four challenging benchmark datasets, HollywoodHeads [16], Casablanca [18], BOSS [19], and PAMELA [20], to evaluate our approach. From experimental results, we observe that the mean average precision (mAP) of the generic detector is increased by 10% by considering the TCM model

The rest of the paper is organized as follows. We discuss related work in Section 2. Section 3 elaborates the details of the proposed method. Experiment results are discussed in Section 4. Section 5 discusses the conclusion and future work.

2. Related Work

Person head detection has numerous applications in video surveillance. Despite significant importance, little work has been reported in the literature regarding head detection. Most of the existing methods focus on face detection and pedestrian detection. These methods model the problem of

Figure 2: The results of our proposed Refinement Algorithm 1 are shown. The algorithm suppressed false positives in all video frames.

Figure 3: It shows improved results by recovering missed detections via the proposed unique tracking method.

head detection as a multiobject detection problem. Head detection can provide localization information that can be further utilized for face recognition in real-time applications. For the face recognition problem, some researchers focused on metric learning [21–23], and some researchers focused on feature representation [24–27]. In this section, we review some methods for face detection and head detection.

With the tremendous success of CNNs, most of the current methods are based on CNNs for detecting faces in images and videos [28, 29]. In recent years, many methods have been proposed in the literature that exploits contextual information for detecting human faces in complex scenes [30, 31]. In order to capture a wide range of scales, Hu and Ramanan [32] proposed a method that finds tiny faces by exploiting contextual information. Hao et al. [33] propose a face detector that detects human faces in a wide range of scales in an image. Face detection also plays an important role and serves as a preprocessing step in face recognition problems. Lu et al. [25] proposed a face recognition model by learning face descriptors using the feature mapping of pixel vectors. Duan et al. [34] exploit contextual information by using the local binary feature of adjacent pixels. The aim of metric learning is to measure similarity among features. Hu et al. [22] proposed a multimetric learning model by learning global distance metrics. We claim that our proposed model can also be used to improve the performance of face recognition systems by suppressing regions that do not contain humans' heads.

Generally, the task of head detection is similar to face detection, but compared to the structure of the face, the head has a limited number of features and can change drastically due to perspective distortions. Therefore, we cannot apply a face detector for the head detection task. Since face detectors rely on facial features, these features are impossible to extract when a person turns his back to the camera. The traditional head detection model uses statistical features to learn a nonlinear classifier. For instance, Viola and Jones [35] used Haar-like features and learn a Haar-cascade classifier for face classification. The method is then improved and extended by [18] using a conditional random field (CRF). The deformable part model (DPM) [36] is a popular model that used a histogram of oriented gradient (HOG) features. Ishii et al. train a linear classifier using hand-craft features. These traditional methods work well in a low-density environment; however, their performance suffers in complex scenes. Moreover, these models incur high computational costs due to the computation of complex features.

Recent head detection models are based on the convolutional neural network that extracts hierarchical features and learns a better representation of human heads. With the success of using contextual information in object detection tasks, Vu et al. [16] exploit contextual information by proposing a context-aware CNN model that leverages the relations between person-to-scene and person-to-person. Li et al. [37] proposed Faster R-CNN-based model that exploits regional context. Similarly, Wang et al. [38] fused multiscale features using SSD [37]. These methods improve the existing generic detection models, i.e., Faster R-CNN and SSD, by incorporating multiscale fusion strategy and contextual information. Furthermore, the most recent head detection method is proposed in [39] that employs the CNN model and learns the semantic connection between the human head and other body parts. Li et al. [9] proposed an end-to-end adaptive relational network that exploits contextual information to detect heads.

3. Proposed Methodology

In this section, we discuss the proposed methodology of detecting human heads. The pipeline of our framework has two sequential stages. In the first stage, we use an existing generic head detector to obtain initial head detection. The detection obtained during the first stage contains false positives. Moreover, the detector may suffer from miss detection at this stage. In the second stage, the obtained detections are refined by employing our proposed TCM model that suppresses false positives and recover missed detection, hence improving mean average precision (mAP) and recall rate.

3.1. Head Detection. Our head detection model follows the traditional pipeline of Faster R-CNN [15]. We use the model that is initially trained on ImageNet [40] and fine-tuned it on

the HollyWood dataset [16] for head detection. We use several backbone architectures, for example, VGG16 [41], VGGM [41], and ZF [42]. From empirical studies, we observed that VGG16 outperformed other architectures but caused computational complexity during training and testing. Faster R-CNN is a two-stage network. During the first stage, the region proposal network (RPN) generates proposals of various scales, while the classification of these object proposals is carried out in the second stage. We also fine-tuned the YOLO model [14] and single-shot detector (SSD) [13] on the HollywoodHeads dataset. These models tackle detection as a regression problem. To tackle the scale problem, YOLO divides the image into 13×13 cells of equal size. YOLO then treats each cell of a gird as an object proposal and predicts its confidence score. We observe that YOLO works faster as compared to its counterparts; however, it compromises the accuracy.

3.2. Temporal Consistency Model. Detection obtained by previous methods contains false positives. Moreover, the detector may suffer from miss detection due to occlusion and severe clutter in the scene. In order to address this problem, we leverage spatial-temporal information that naturally exists in videos. Generally, in object detection tasks, end-to-end learning is the most useful way of solving detection problems; however, in the case of videos, the end-to-end learning approach can cause significant computational and memory costs. Therefore, as a solution, we propose a TCM model based on intensity and gradient consistency assumption. We assume that objects detected in the first frame travel few pixels, thereby maintaining intensity and gradient consistency. To mathematically model this assumption, we define an energy function E. Energy function E has two main components, i.e., the intensity constancy model and the gradient constancy.

For calculating displacement vector, which represents the change in x and y directions, we assume that the intensity does not change [43] and is given by

$$\Psi(i, j, k) = \Psi(i + \hat{x}, j + \hat{y}, k + 1), \tag{1}$$

where $\Psi : \gamma \epsilon \mathbb{R}^3 \rightarrow \mathbb{R}$ is the bounding box represented as a 4-D vector $(i, j, \hat{x}, \hat{y})$, where $\tau \coloneqq (\hat{x}, \hat{y}, 1)$ is the displacement vector and (i, j) represents spatial coordinates of a pixel.

For calculating τ, we only consider (i, j) and assume the width and height of bounding box Ψ are the same. Therefore, we do not consider the size of the bounding box in the equation. In our calculations, we assume the gradient of intensity values. Since the intensity values of pixels are very sensitive to environmental changes, small changes in illumination may cause a huge change in the intensity values of pixels. In our calculation, we assume that the gradient of intensity values does not change due to the illumination and other environment disturbance [44] and is given by

$$\nabla\Psi(i, j, k) = \nabla\Psi(i + \hat{x}, j + \hat{y}, k + 1), \tag{2}$$

where ∇ is the gradient that captures the change in pixel's intensity value between current frame k and next frame $k + 1$. Equation (2) does not consider the influence of neighboring pixels. In this case, the model may encounter a problem of diminishing gradient. Moreover, this model may also catch outliers. In order to incorporate the influence of neighboring pixels, we employ spatial and temporal smoothing constraints in Equation (2).

After defining intensity and gradient consistency assumptions, we now formulate an energy function that computes the cost of deviations from the above-mentioned assumptions. The cost of deviation from the intensity and gradient consistency assumption [43] is computed as follows:

$$E_a = \int_{\gamma} \left(\left| \Psi(s + \tau) - \Psi(s) \right|^2 + \upsilon \left| \nabla\Psi(s + \tau) - \nabla\Psi(s) \right|^2 \right) ds, \tag{3}$$

where $s \coloneqq (i, j, k)$ and υ balances two terms in the equation. The smoothness equation which computes the cost of total variations in the flow field is computed as follows:

$$E_b = \int_{\gamma} \left(\left| \nabla \hat{x} \right|^2 + \left| \nabla \hat{y} \right|^2 \right) ds. \tag{4}$$

The final energy function E is the linear combination of Equations (3) and (4) and is given by

$$E(\hat{x}, \hat{y}) = E_a + \alpha E_b, \tag{5}$$

where α is a regularization parameter with $\alpha > 0$. For every pixel, we compute its displacement vector by minimizing the above energy function in Equation (5).

We now discuss our proposed temporal refinement algorithm that leverages temporal information to suppress false positives and recover missed detection.

3.2.1. Suppressing False Positives. Before recovering the missed detection, we first refine the detection by suppressing false positives. Let $D_t = \{d_1, d_2, \cdots, d_n\}$ represents n number of detections in frame t. Let $\Omega = \{D_1, D_2, \cdots, D_m\}$ is a container that contains detections of video sequence having m frames. To suppress false positives in frame k, we use similarity criteria φ between the detection d_t in the current frame and detection d_{t+1} in the next frame.

Algorithm 1 takes D_k for frame k as an input and gives a refined output R_k. For each detection, $d_i \epsilon D_k$ at frame k, we first define a temporal window W, and then by using Equation (5), we compute its location in the next frame $k + 1$. We then compute similarity σ, and distance τ between the centroids of the current detection d_i, and all detections belongs to D_{k+1}. We then select detection d_j that gives a maximum value of φ. We compute the final score φ for each detection d_i considering the predefined temporal window of size W. We define a threshold ϵ and delete detections whose confidence score is less than 0.5. We process whole container Ω in the same way.

3.2.2. Recovering Missed Detection. After refining detection, the next step is to recover detection that was missed by the generic detector. We utilize the method in [18] to recover

Input: Detections $D_{k.}$
Output: Refined detections R_k
1: **function** DETECTION REFINEMENT $D_{k.}$
2: $R_k \leftarrow 0$
3: $D_{k+1} = \{d_1, d_2, \cdots, d_{k+W}\}$
4: **for each** detection d_i in D_k **do**
5: **for each** d_j in D_{k+1} **do**
6: Compute similarity $\sigma(d_i, d_j) = |d_i - d_j/d_i + d_j|$
7: Compute distance $\tau(d_i, d_j) = \sqrt{(d_i - d_j)^2}$
8: $\varphi = \varphi + (1 - (\tau/\text{Max}\ (W, H))\sigma)$
9: **end for**
10: **if** $\varphi/W > \varepsilon$ **then**
11: Insert d_i in R_k
12: **end if**
13: **end for**
14: **return** R_k
15: **end function**

Algorithm 1. Refinement algorithm.

the missed detection via tracking. Generally, the tracker has two modes:

(1) Tracking via detection and

(2) Tracking via temporal correlation

For every detection, a tracker is initialized at frame k. If detection is found in frame $k+1$, the tracker follows the first mode. This mode is robust and invariant to the appearance, scale, and pose of human heads. In case the tracker could not find the detection in the next frame, it switches to the second mode.

In the second mode, for each detection d_i, we compute the next location of detection by using Equation (5). For detection d_i at frame k, let p_k and s_k represent its position and scale, respectively. Let $\bar{p}_k$ and $\bar{s}_k$ are the observations and $\widehat{p}_{k|k+1}$ and $\widehat{s}_{k|k+1}$ are the predictions. We keep head template patch H_{patch} search for the best match in the next frame around location $\widehat{p}_{k|k+1}$ and size $\widehat{s}_{k|k+1}$. The matching criteria are based on texture and appearance similarity between H_{patch} and patch in the next frame. We assign detection d_j to a current track if $\|\widehat{p}_{k|k+1} - p_{k+1}\| < \varsigma$ and $\|\widehat{s}_{k|k+1} - s_{k+1}\| < \beta$, where p_{k+1} is the position and s_{k+1} is the size of detection d_j at frame $k+1$. In case the tracker does not find the best match, the tracker update H_{patch} as follows:

$$H_{\text{patch}} = \left(1 - \psi H_{\text{patch}} + \psi I(p_{k+1}, s_{k+1})\right), \tag{6}$$

where ψ is the balancing parameter and $I(p_{k+1}, s_{k+1})$ is the patch in the next frame. In our experiments, we fix the value of ψ to 0.3. During tracking, we maintain N_1 and N_2, where N_1 is the number of frames that a tracker follows mode 1 and N_2 represents the number where the tracker follows mode 2.

4. Experimental Results

In this section, we discuss experimental results, and in order to qualitatively and quantitatively evaluate our proposed method, we use four publicly available benchmark datasets, i.e., HollywoodHeads, Casablanca, BOSS, and PAMELA datasets. We have shown sample frames from these datasets in Figure 4. We provide the details of each dataset as follows:

The BOSS dataset is originally proposed in [19]. This dataset contains 16 video sequences collected using 9-10 cameras on the moving train. The video sequences cover different anomalous behaviors, for example, theft, fight, and fainting as well as normal behaviors. The dataset is initially proposed to evaluate anomaly detection algorithms. However, the dataset also contains normal behaviors where people walk along the corridor of the train in different directions. One of the problems with the BOSS dataset is that it provides annotations for different anomalous and normal behaviors; however, annotations of heads are missing. Therefore, we annotate human heads and generate ground truth for all video sequences of the BOSS dataset. For annotations, we use the VIPER-GT [45] publicly available annotation tool and mark the position of each person by drawing a bounding box around the head. It is to be noted that we extend the size of the bounding box by 10% following the convention in [46]. After generating the ground truth, we then extract positive patches (belong to the head in the original image) and background for training the network.

PAMELA dataset is first proposed by [20]. This dataset is collected in 2008 to simulate the metro carriage at the London Underground station. The dataset consists of video sequences captured from different cameras with different viewpoints. For the human head detection problem, we use video sequences that were captured from the orthogonal views to avoid perspective distortions. The dataset covers two main situations at the train station:

(1) People alighting or getting off the train

(2) People waiting and then boarding

The alighting situation contains eight video sequences; waiting and boarding contain seven video sequences. The duration of each video sequence is about 1 to 2 minutes, with the 352×588 resolution and frame rate of 25 frames per second. For generating the ground truth, the authors used ViPER [47] to annotate the head of pedestrians in each frame. While annotating human heads, the authors extend the bounding box to cover also the shoulder of pedestrians in order to capture contextual information. We then generate positive and negative samples for training and testing the model. Our training set consists of a total of 109,376 samples, among which 43,751 are positive, and 65,625 are negative samples, while the testing set consists of 103,831 samples, among which 41,533 are positive and 62,298 are negative samples.

HollywoodHeads dataset is first proposed by Vu et al. [16] for evaluating head detection models. The dataset contains a total of 224,740 images that were collected from 21 Hollywood movies. The video sequences demonstrate huge

(a) BOSS dataset (b) Casablanca dataset

(c) PAMELA dataset (d) HollywoodHeads dataset

Figure 4: Samples from benchmark datasets.

variations in illumination, camera viewpoints, pose, and scales of human heads. In this dataset, 369,846 human heads are annotated in a way that can be easily deployed to train deep convolutional networks. The authors adopt the frame skip strategy by annotating the initial frame, and consecutive frames are annotated by linear interpolation of bounding boxes. For training and testing, we follow the same convention adopted by the authors. We use 216,719 frames for training (collected from 15 movies) and 1,302 frames (collected from the remaining 3 movies) for validation.

Casablanca dataset contains video sequences from the old movies and was first proposed by Ren [18] for evaluating head detection models. The dataset contains 147,600 frames of low resolution 464×640 and high variations in head scales and poses. The dataset is annotated in a way to cover the front face of people. To evaluate the effectiveness of the proposed TCM, we use different state-of-the-art generic detectors, for example, Faster R-CNN [15], YOLO [14], SSD [13], and R-CNN [12]. It is to be noted that the choice of these generic detectors is arbitrary. Instead of these detectors, one can use any good human detection model that is robust and performs detection in a wide range of scales. These detectors are used to provide initial detections that will be refined by employing the proposed TCM. It is to be noted that we train each of the above generic detectors from scratch. We first trained each detector using the ImageNet dataset and then fine-tuned the model on benchmark datasets used in this work. To evaluate the effectiveness of the proposed approach, we used mean average precision (mAP) that is calculated from the area under the precision-recall curve and has been used as a standard metric for evaluating object detectors.

4.1. Ablation Study. We follow the original implementation of generic detectors for ablation study. However, we introduce some changes during fine-tuning the network. For training Faster R-CNN, the frames used for testing are first rescaled to a shorter dimension of about 500 pixels. We keep the size of anchor boxes up to 10 scales, which has the potential for capturing scale variations in the image. In our experiments, we used different backbone networks, for example, VGG16 [41], VGGM [41], and ZF [42]. During the fine-tuning process, we allow 100k iterations and we analyze the network performance using mAP at every 10k interactions, as shown in Figure 5. From the figure, it is obvious that VGG16 performs well compare to other networks.

In the same way, we trained SSD and YOLO detectors, and the performances of these detectors are shown in Figure 6. For SSD, we use VGG16 as baseline architecture. From Figure 6, it is evident that YOLO performed comparatively lower than SSD. This is due to the fact that YOLO used a limited number of object proposals for detecting objects in the scene.

During the ablation study, we use different state-of-the-art generic detectors with different backbone CNN architectures. Comprehensive results on each dataset are reported in Tables 1–4. The third row of all tables shows the mean average precision (mAP) obtained by the detectors. The

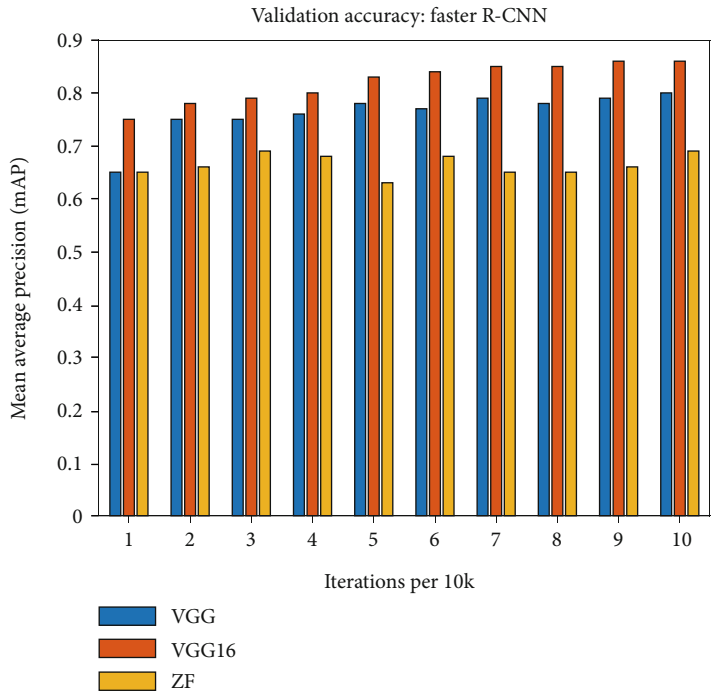

Figure 5: Performance of Faster R-CNN using different network architecture at every 10k iteration on HollywoodHeads dataset.

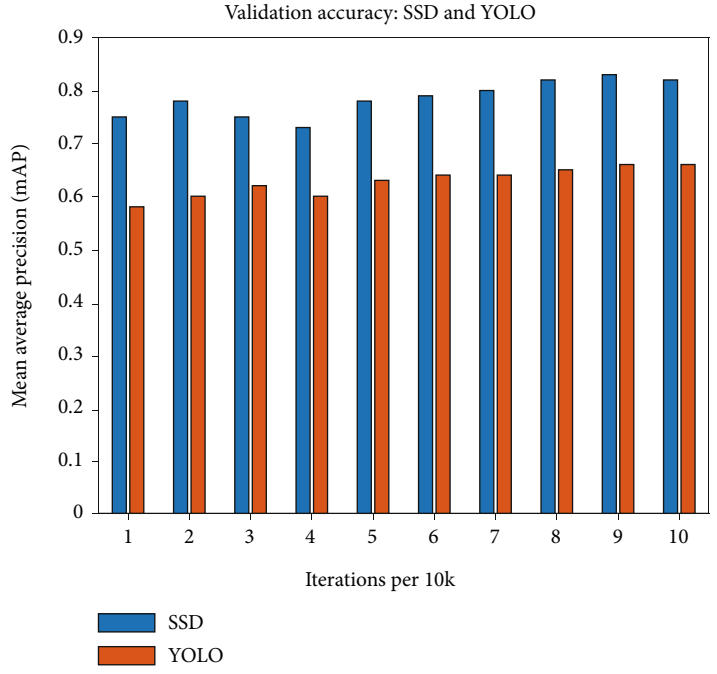

Figure 6: Summary of the performance of the network at different iterations using YOLO and SSD.

fourth column of all tables shows the mAP after employing the proposed TCM. It is obvious from the fifth column of all tables that mAP is improved for all detectors after employing the proposed TCM model.

From the third column of tables, it is obvious that the performances of generic detectors reach to 78% to 80% in some cases; however, there is still room for improvement. From empirical studies, we observe that generic detectors suffer from missed detection and accumulate false positives that lower the precision and recall rates. The proposed TCM model tackles this problem by employing spatiotemporal information that suppresses false positives and recovers missed detection.

Table 1 shows the performance of different generic detectors on the HollywoodHeads dataset. From the table, it is

Table 1: Performance evaluation on HollywoodHeads dataset.

Detectors	Baseline CNN	mAP	TCM (mAP)	Improvement (%age)
Faster R-CNN [15]	ZF [42]	0.76	0.83	9.20%
	VGG16 [41]	0.79	0.85	7.59%
	VGGM [41]	0.78	0.81	3.84%
R-CNN [12]	ZF [42]	0.71	0.79	11.26%
	VGG16 [41]	0.74	0.83	12.16%
	VGGM [41]	0.72	0.81	12.67%
YOLO [14]	13-layered architecture	0.39	0.46	18.42%
SSD [13]	VGG16 [41]	0.43	0.54	25.58%
			Average	12.59%

Table 2: Performance evaluation on Casablanca dataset.

Detectors	Baseline CNN	mAP	TCM (mAP)	Improvement (%age)
Faster R-CNN [15]	ZF [42]	0.48	0.54	12.50%
	VGG16 [41]	0.55	0.61	10.90%
	VGGM [41]	0.51	0.55	7.84%
R-CNN [12]	ZF [42]	0.43	0.45	4.65%
	VGG16 [41]	0.53	0.57	7.54%
	VGGM [41]	0.48	0.53	10.41%
YOLO [14]	13-layered architecture	0.31	0.40	20.30%
SSD [13]	VGG16 [41]	0.38	0.42	10.52%
			Average	8.46%

Table 3: Performance evaluation on BOSS dataset.

Detectors	Baseline CNN	mAP	TCM (mAP)	Improvement (%age)
Faster R-CNN [15]	ZF [42]	0.74	0.79	6.75%
	VGG16 [41]	0.80	0.87	8.75%
	VGGM [41]	0.78	0.84	7.69%
R-CNN [12]	ZF [42]	0.72	0.79	9.72%
	VGG16 [41]	0.75	0.83	10.95%
	VGGM [41]	0.73	0.80	9.58%
YOLO [14]	13-layered architecture	0.72	0.82	13.88%
SSD [13]	VGG16 [41]	0.75	0.81	8.00%
			Average	7.54%

obvious that generic detectors achieve good performance in detecting heads. This is due to the reason that the HollywoodHeads dataset contains heads of large size (_100 pixels), and scale variations are not significantly large. Therefore, it is a trivial job for a single-scale detector to detect heads in this dataset. However, we observe that these detectors missed many detections in different frames.

Furthermore, these detectors also accumulate false positives that reduce precision and recall rates. However, by employing the proposed TCM model, the performance of each detector is improved by 12.56% on average, as shown in the fourth column of the table. Table 2 shows the performance of detectors on the Casablanca dataset. From the table, it is obvious that generic detectors could not perform well as compared to other datasets. This is due to the reason that the dataset contains low-resolution frames, with significant variation in head scales. Furthermore, the size of heads in most of the cases was extremely small, and it was challenging for generic detectors to detect small heads. However, by employing the proposed TCM, the average performance of the generic detector is improved by 8.46%. Table 3 shows performance on the BOSS dataset. It is evident that generic

Table 4: Performance evaluation on PAMELA dataset.

Detectors	Baseline CNN	mAP	TCM (mAP)	Improvement (%age)
Faster R-CNN [15]	ZF [42]	0.63	0.67	6.34%
	VGG16 [41]	0.67	0.71	5.97%
	VGGM [41]	0.65	0.69	6.25%
R-CNN [12]	ZF [42]	0.52	0.55	5.76%
	VGG16 [41]	0.56	0.59	5.35%
	VGGM [41]	0.54	0.59	8.25%
YOLO [14]	13-layered architecture	0.59	0.63	6.77%
SSD [13]	VGG16 [41]	0.62	0.64	3.22%
			Average	4.79%

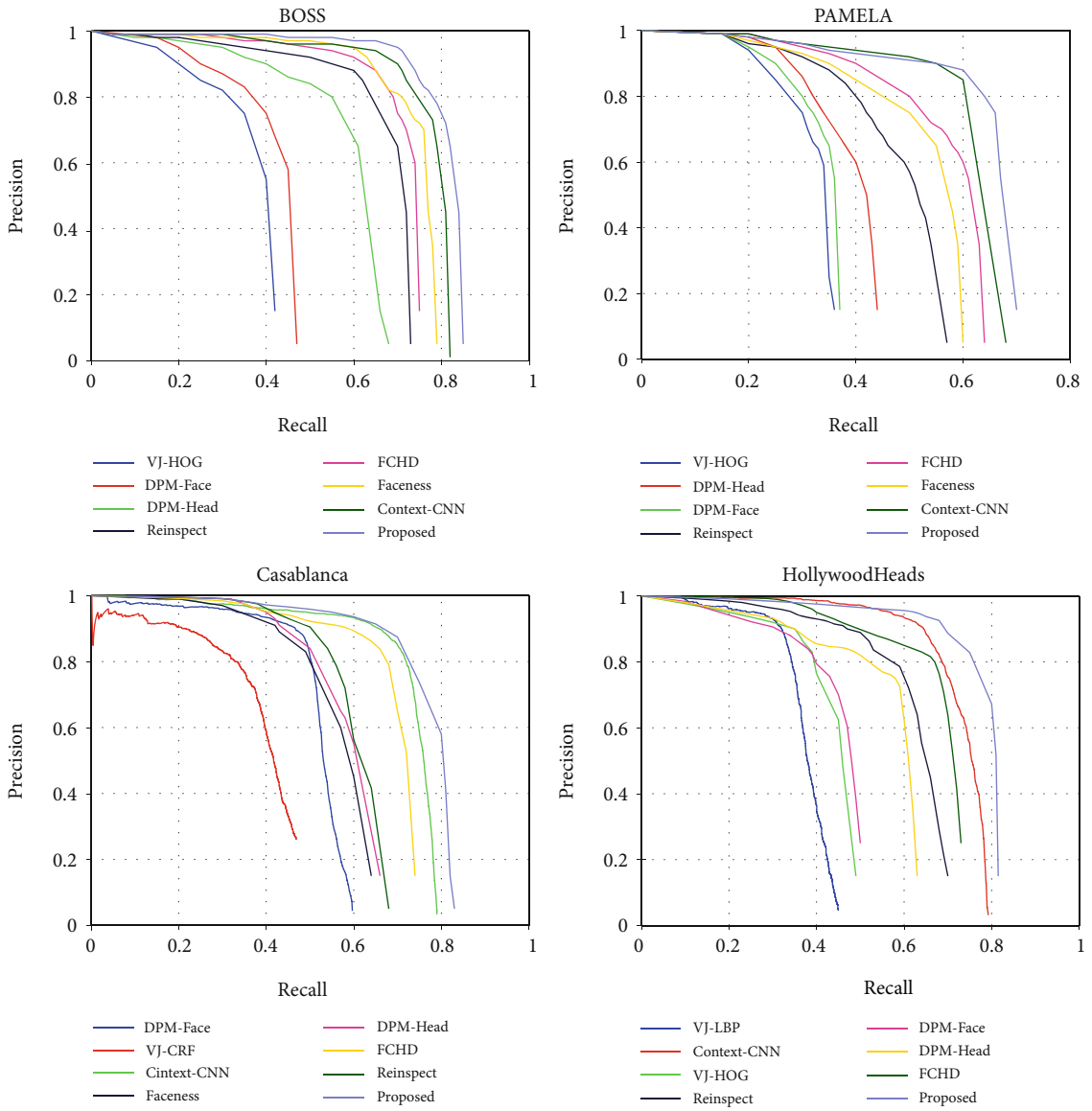

Figure 7: Precision-recall curves of different specific detectors on different datasets.

detectors perform well on this dataset as compared to other datasets. The dataset is relatively less dense and contains 2-5 persons per image. The heads are clearly visible with limited scales and perspective distortions that make head detection trivial in this dataset. As obvious from the fourth column of the table that by employing the proposed TCM, the

Figure 8: Qualitative comparison of the proposed method with reference methods using all benchmark datasets. The first row shows the results on the Casablanca dataset, the second row shows results on the BOSS dataset, the third row shows results on the HollywoodHeads dataset, and the last row shows results on the PAMELA dataset.

performance is increased to 7.53%. Table 4 shows performance on the PAMELA dataset. In this dataset, the proposed TCM model improves the performance of generic detectors; however, the average performance is 4.79 which is lower than achieved on other datasets. This is due to the reason that TCM could not recover detections. The tracker is lost after a few frames due to poor quality of image, cluttered background, and illumination changes.

From the tables, it is observed that the performance of generic detectors is relatively high on all datasets except PAMELA. The lower performance attributes to the complexity of the dataset. This dataset contains human heads of relatively small size, and it is challenging for the generic detector to detect such small human heads. The reason is that generic detectors generate a feature map from the last convolutional layer due to which features of the small objects become too small to be detected.

4.2. Comparison with a Head Detector. In this section, we compare the performance of different state-of-the-art head detectors with the proposed framework. These head detectors include VJ-LBP [35], VJ-HOG [35], Faceness [29], deformable part model (DPM-Head) [36], deformable part model (DPM-Face) [36], fully convolutional head detector (FCHD) [48], Reinspect [49], and context-aware convolutional neural network for head detection (Context-CNN) [16]. For comparisons, we select the best model among the models discussed in Tables 1–4. For fair comparisons, we use pretrained models of state-of-the-art head detectors and fine-tuned the models on analyzed datasets. From our empirical studies, we observe that the performance of state-of-the-art head detectors improves after finetuning. We use a precision-recall curve with different thresholds to rigorously evaluate the performance of head detectors.

The performance of different state-of-the-art detectors is shown in Figure 7. As evident from Figure 7, the proposed model outperforms state-of-the-art methods in all benchmark datasets.

We observed that Faceness [29] performs relatively low than other competing methods. This is due to the reason that the Faceness model is designed for face detection problems. The model extracts facial features from the image in order to detect human faces. However, we observe that the Faceness model could not effectively extract facial features from the image due to occlusions, variation in illumination, scale, and pose. For example, it is challenging for a face detector to detect the face of a person who turns his back to the camera. Furthermore, we observed that VJ-LBP and VJ-HOG performed significantly lower than other detectors. This is due to the fact that these detectors are based on the

Table 5: Speed versus accuracy of different detectors before and after employing the proposed TCM model.

	Detection models	Before TCM		After TCM	
		mAP	Speed (FPS)	mAP	Speed (FPS)
Generic detectors	Faster R-CNN [15]	0.80	17	0.87	10
	YOLO [14]	0.72	50	0.82	43
	SSD [13]	0.75	52	0.81	47
Specific detectors	Context-CNN [16]	0.81	10	0.88	6
	FCHD [48]	0.76	12	0.83	8
	Faceness [29]	0.79	19	0.82	13

traditional Viola-Jones algorithm and the sliding window approach makes the algorithm slow and unfeasible in real time. Furthermore, due to the sliding window approach, the algorithm generates many bounding boxes, and miss classification accumulates many false positives that lower precision and recall rates. We also observe that DPM faces problems in detecting small human heads as DPM detects human parts of size up to 23×23 pixels. However, Context-CNN [16] achieves comparable performance by utilizing both R-CNN as baseline model and exploits contextual information for human head detection in complex scenes. FCHD [39] on the other hand utilizes whole-body context to detect human heads; however, in complex scenes, whole body is not always visible.

The proposed framework, on the other hand, achieves state-of-the-art performance by adopting the temporal consistency model (TCM) that improves the performance of generic detectors. Through this work, we show that generic detectors compared to the specific detector (that are specially designed for head detection task) can perform well by integrating the proposed TCM model.

We also qualitatively evaluate and compare results with other reference methods in Figure 8. It is evident from the figure that by integrating the proposed model, a generic detector achieves better performance compared to head detection models specifically designed for head detection tasks.

4.3. Inference Speed. In this section, we discuss the inference speed of detectors before and after the integration of the proposed TCM model. For all experiments, we used a desktop computer equipped with Intel Core i7-8700K 8th generation CPU, 16 GB RAM, and NVIDIA Titan Xp GPU. It is also to be noted that the implementation of generic detectors, e.g., Faster R-CNN, SSD, and YOLO, is done in PyTorch library, while specific detectors, for example, FCHD, Context-CNN, and Faceness, is done in Keras and TensorFlow. We report the inference speed (in frames per second) versus accuracy (in terms of mean average precision (mAP)) for each detector with a batch size of 1 in Table 5. From the table, it is obvious that the performance of generic as well as specific detectors is improved after employing the proposed TCM model. However, integration of the proposed TCM model caused additional cost (in terms of speed). This is due to the reason that analyzed generic and specific detectors are specifically designed to detect heads in a single frame of video, while the proposed TCM model improves head detection by exploiting temporal consistency that exists among multiple frames. To model temporal consistency, the proposed TCM model causes additional cost by recovering missed detection and suppressing false positives in the selected temporal sequence.

5. Conclusion

In this work, we propose a novel model to recover missed detection and suppress false positives by leveraging temporal consistency that exists among subsequent frames of videos. The main objective of this work is to improve the performance of generic object detectors by integrating spatial-temporal information. We evaluate our proposed model using challenging benchmark datasets that contain severe clutter and occlusions. From experiments, we observed that the mAP is improved by employing the proposed temporal consistency model. We believe that the proposed model can also be applied to any other object detector. Our model can also be useful in identifying and localizing human behaviors and emotion recognition and is part of our future work.

Data Availability

We used publicly available datasets in our research. The source and URL of the datasets are given in the following. The sources are also cited in the paper: HollywoodHeads dataset: https://www.robots.ox.ac.uk/~vgg/software/headmview/;Casablanca dataset: https://www.di.ens.fr/willow/research/headdetection/;Boss dataset: http://velastin.dynu.com/videodatasets/BOSSdata/;PAMELA dataset: https://www.di.ens.fr/willow/research/headdetection/.

Conflicts of Interest

The authors declare no conflict of interest.

Acknowledgments

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU for this research.

References

[1] Y. Tian, A. Dehghan, and M. Shah, "On detection, data association and segmentation for multi-target tracking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 9, pp. 2146–2160, 2018.

[2] M. Ullah and F. Alaya Cheikh, "A directed sparse graphical model for multi-target tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 1816–1823, Salt Lake City, Utah, June 2018.

[3] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6479–6488, Salt Lake City, Utah, June 2018.

[4] H. Ullah, M. Ullah, and N. Conci, "Real-time anomaly detection in dense crowded scenes," in *Proceedings Volume 9026, Video Surveillance and Transportation Imaging Applications 2014*, p. 902608, San Francisco, California, United States, 2014, International Society for Optics and Photonics.

[5] S. D. Khan, "Congestion detection in pedestrian crowds using oscillation in motion trajectories," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 429–443, 2019.

[6] Y. Li, M. Sarvi, K. Khoshelham, and M. Haghani, "Multi-view crowd congestion monitoring system based on an ensemble of convolutional neural network classifiers," *Journal of Intelligent Transportation Systems*, vol. 24, no. 5, pp. 1–12, 2020.

[7] S. D. Khan, S. Bandini, S. Basalamah, and G. Vizzari, "Analyzing crowd behavior in naturalistic conditions: identifying sources and sinks and characterizing main flows," *Neurocomputing*, vol. 177, pp. 543–563, 2016.

[8] M. Ullah, H. Ullah, N. Conci, and F. G. De Natale, "Crowd behavior identification," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, pp. 1195–1199, Phoenix, AZ, USA, Sept. 2016.

[9] W. Li, H. Li, Q. Wu, F. Meng, L. Xu, and K. N. Ngan, "Headnet: an end-to-end adaptive relational network for head detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 2, pp. 482–494, 2019.

[10] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, Boston, MA, USA, June 2015.

[11] Z. Sun, D. Peng, Z. Cai, Z. Chen, and L. Jin, "Scale mapping and dynamic re-detecting in dense head detection," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, pp. 1902–1906, Athens, Greece, Oct. 2018.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, Columbus, OH, USA, June 2014.

[13] W. Liu, D. Anguelov, D. Erhan et al., "Ssd: single shot multibox detector," in *European conference on computer vision*, pp. 21–37, Cham, 2016.

[14] J. Redmon and A. Farhadi, "YOlO9000: better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7263–7271, Honolulu, HI, USA, July 2017.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, pp. 91–99, Montreal, Quebec, Canada, December 2015.

[16] T.-H. Vu, A. Osokin, and I. Laptev, "Context-aware cnns for person head detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2893–2901, Santiago, Chile, December 2015.

[17] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[18] X. Ren, "Finding people in archive films through tracking," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE conference on*, pp. 1–8, Anchorage, AK, USA, June 2008.

[19] http://multitel.be/projects/boss/.

[20] "UCL Transport Institute," 2017, http://www.ucl.ac.uk/transportinstitute/Researchsnapshots/PAMELA.

[21] Y. Duan, J. Lu, J. Feng, and J. Zhou, "Deep localized metric learning," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 2644–2656, 2017.

[22] J. Hu, J. Lu, Y.-P. Tan, J. Yuan, and J. Zhou, "Local large-margin multimetric learning for face and kinship verification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 8, pp. 1875–1891, 2017.

[23] J. Lu, G. Wang, and P. Moulin, "Localized multi-feature metric learning for image-set-based face recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 3, pp. 529–540, 2015.

[24] Y. Duan, J. Lu, J. Feng, and J. Zhou, "Learning rotation-invariant local binary descriptor," *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 3636–3651, 2017.

[25] J. Lu, V. E. Liong, X. Zhou, and J. Zhou, "Learning compact binary face descriptor for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 10, pp. 2041–2056, 2015.

[26] J. Lu, G. Wang, and J. Zhou, "Simultaneous feature and dictionary learning for image set based face recognition," *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 4042–4054, 2017.

[27] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1891–1898, Columbus, OH, USA, June 2014.

[28] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, "A convolutional neural network cascade for face detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5325–5334, Boston, MA, USA, June 2015.

[29] S. Yang, P. Luo, C.-C. Loy, and X. Tang, "From facial parts responses to face detection: a deep learning approach," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3676–3684, Santiago, Chile, December 2015.

[30] K. Zhang, Z. Zhang, H. Wang, Z. Li, Y. Qiao, and W. Liu, "Detecting faces using inside cascaded contextual cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3171–3179, Venice, Italy, October 2017.

[31] C. Zhu, Y. Zheng, K. Luu, and M. Savvides, "Cms-rcnn: contextual multi-scale region-based cnn for unconstrained face detection," in *Deep Learning for Biometrics*, pp. 57–79, Springer, 2017.

[32] P. Hu and D. Ramanan, "Finding tiny faces," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 951–959, Honolulu, HI, USA, July 2017.

[33] Z. Hao, Y. Liu, H. Qin, J. Yan, X. Li, and X. Hu, "Scale-aware face detection," in *Proceedings of the IEEE Conference on*

Computer Vision and Pattern Recognition, pp. 6186–6195, Honolulu, Hawaii, USA, July 2017.

[34] Y. Duan, J. Lu, J. Feng, and J. Zhou, "Context-aware local binary feature learning for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 5, pp. 1139–1153, 2018.

[35] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[36] J. Yan, Z. Lei, L. Wen, and S. Z. Li, "The fastest deformable part model for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2497–2504, Columbus, OH, USA, June 2014.

[37] Y. Li, Y. Dou, X. Liu, and T. Li, "Localized region context and object feature fusion for people head detection," in *2016 IEEE International Conference on Image Processing (ICIP)*, pp. 594–598, Phoenix, AZ, USA, September 2016.

[38] Y. Wang, Y. Yin, W. Wu, S. Sun, and X. Wang, "Robust person head detection based on multi-scale representation fusion of deep convolution neural network," in *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 296–301, Macau, China, December 2017.

[39] G. Chen, X. Cai, H. Han, S. Shan, and X. Chen, "Headnet: pedestrian head detection utilizing body in context," in *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 556–563, Xi'an, China, May 2018.

[40] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: a large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Miami, FL, USA, 2009.

[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, http://arxiv.org/abs/1409.1556.

[42] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, pp. 818–833, Cham, 2014.

[43] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[44] S. Uras, F. Girosi, A. Verri, and V. Torre, "A computational approach to motion perception," *Biological Cybernetics*, vol. 60, no. 2, pp. 79–87, 1988.

[45] G. Chen and R. Hou, "A new machine double-layer learning method and its application in non-linear time series forecasting," in *2007 International Conference on Mechatronics and Automation*, pp. 795–799, Harbin, China, August 2007.

[46] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, pp. 886–893, San Diego, CA, USA, June 2005.

[47] D. Doermann and D. Mihalcik, "Tools and techniques for video performance evaluation," in *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, pp. 167–170, Barcelona, Spain, September 2000.

[48] A. Vora and V. Chilaka, "Fchd: fast and accurate head detection in crowded scenes," 2018, http://arxiv.org/abs/1809.08766.

[49] R. Stewart, M. Andriluka, and A. Y. Ng, "End-to-end people detection in crowded scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2325–2333, Las Vegas, NV, USA, June 2016.