Scale-Aware Multi-Path Deep Neural Networks for Unconstrained Face Detection

Yuguang Liu

Department of Electrical and Computer Engineering
McGill University
Montreal, Canada

May 2017

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of Engineering.

Abstract

Unconstrained face detection is the task of robustly finding and locating faces in an image subject to possible variations in facial scale, blur, pose, illumination, occlusion, and facial expression. It is a critical first step towards a host of modern surveillance applications, including but not limited to face verification, face recognition, face tracking, and human-computer interaction. Though much progress has been made in unconstrained face detection during the past decade, the majority of work focuses on improving the detection robustness on variations caused by blur, pose, illumination, occlusion and facial expression. Facial scale, despite its immense influence on face detection accuracy, has received much less attention than have the above factors. This is partially due to the fact that most traditional face detection benchmark datasets tend to collect faces of relatively large size and with modest scale variation. Nonetheless, in real-world applications, such as surveillance systems, it is imperative to possess an equal ability to detect both big faces (close to camera) and tiny ones (far away from the camera) at the same time. To the best of our knowledge, no published face detection algorithm can detect a face as large as 1000×1000 pixels while simultaneously detecting another one as small as 10×10 pixels within a single image with similarly high accuracy.

We introduce a Multi-Path Face Detection Network (MP-FDN) to filter an image for simultaneously proposing and verifying different sized faces in parallel paths. This is the first time that faces across a large span of scales are detected by a single network with forked detection paths. More importantly, the division of the paths are not handcrafted, but totally based on the scale sensitivity inherent in the convolutional networks that was also discovered in this thesis for the first time. MP-FDN consists of two stages. The first stage is a Multi-Path Face Proposal Network (MP-FPN) that suggests faces at three different scale ranges. This design is based on our observation that the hierarchical multi-scale layers of deep convolutional networks (ConvNet) can inherently represent face patterns at multiple scales. In particular, low-level ConvNet layers are more sensitive to tiny faces, while high-level ConvNet layers are more discriminative to big faces. To this end, MP-FPN utilizes three parallel outputs of the convolutional feature maps to simultaneously predict small, medium and large candidate face regions, respectively. The second stage is a Multi-Path Face Verification Network (MP-FVN) that further eliminates false positives while including false negatives. MP-FVN utilizes the same three parallel paths as MP-FPN.

For each detection path, it pools features from both a face candidate region (provided by MP-FPN) and a larger contextual region (surrounding the face candidate region).These facial and contextual features are then concatenated to provide a more accurate "faceness" probability to the face candidate. Note that the network structure and hyper-parameters of MP-FPN and MP-FVN are completely based on controlled experiments, rather than being "handcrafted".

To testify to the performance of MP-FDN on the basis of its ability to perform face detection, we conducted comprehensive experiments on two challenging public face detection benchmark datasets: WIDER FACE and FDDB datasets. MP-FDN consistently achieves better than the state-of-the-art performance on both of them. Specifically, on the most challenging so-called "hard partition" of WIDER FACE test set that contains faces as small as about 9 pixels and as large as more than 1000 pixels in height, MP-FDN outperforms the former best result by 9.8% for the Average Precision. This demonstrates that MP-FDN is a viable and accurate face detector for unconstrained face detection, especially in the case of large scale variations.

Résumé

La détection de visage sans contrainte est l'action de trouver et localiser de façon précise des visages dans une image, avec de possibles variations dans la taille, la netteté, la pose, l'éclairage, l'occlusion ou l'expression du visage. C'est une première étape critique permettant la créationd'un grand nombred'applications de surveillance modernes, qui comprendla vérification faciale, la reconnaissance faciale, le pistage des visages et l'interaction homme-machine. Bien que beaucoup de progrès aient été réalisés dans la détection de visage sans contrainte au cours de la dernière décennie, la majorité du progrès a consisté à améliorer la détection en présence de variations causées par la netteté, la pose, l'éclairage, l'occlusion et l'expression du visage. La taille du visage, malgré son influence sur la précision de détection de visage, a reçu beaucoup moins d'attention que les facteurs ci-dessus. Ceci est en partie dû au fait que, traditionnellement, la plupart des jeux de données utilisés dans la détection des visagestendent à rassembler des visages de taille relativement grande et sans grande variation. Néanmoins, dans de réelles applications, comme les systèmes de surveillance, il est impératif de pouvoir détecter à la fois les gros (près de la caméra) et les petits visages (loin de la caméra). À notre connaissance, aucun algorithme de détection de visage publié ne peut simultanément détecter un visage d'une taille de 1000×1000 pixels et un autre aussi petit que 10×10 pixels dans une seule image avec une précision similaire.

Nous présentons un Réseau de Détection de Visages Multi-Voies (RDV-MV) qui filtre une image en proposant et vérifiant simultanément plusieurs visages de taillesdifférentes dans des voies parallèles. C'est la première fois qu'un seul réseau est capable de détecter plusieurs visages de tailles variables en utilisant des voies de détection fourchues. Plus important encore, la division des chemins n'est pas décidée par l'auteur, mais plutôt basée sur la sensibilité à la taille inhérente aux réseaux convolutifs qui a également été découverte dans cette thèse. RDV-MV se divise en deux étapes. La première étape est un Réseau de Proposition de Visages Multi-Voies (RPV-MV) qui suggère des visages appartenant à trois catégories de taille. Cechoix est basé sur l'observation suivante: les couches àéchelles multiples hiérarchisées des réseaux convolutifsprofonds (ConvNet) peuvent représenter de façon intrinsèque des types de visage de tailles différentes. En particulier, les couches ConvNet de bas niveau sont plus sensibles aux petits visages, tandis que les couches ConvNet de haut niveau sont plus discriminatoires pour les grands visages. À cette fin, RPV-MV utilise trois sorties parallèles des couches convolutives pour prédire simultanément les pe-

tites, moyennes et grandes régions de visages candidats. La deuxième étape est un Réseau de Vérification de Visages Multi-Voies (RVV-MV) qui élimine les faux positifs tout en incluant les faux négatifs. RVV-MV utilise les mêmes trois chemins parallèles que RPV-MV. Pour chaque chemin de détection, le réseau regroupe à la fois les particularités de la région candidate (fournie par RPV-MV) et d'une région contextuelle plus grande (entourant la région candidate). Ces caractéristiques faciales et contextuelles sont ensuite concaténées pour fournir une probabilité plus précise de la validité du visage proposé. Notez que la structure du réseau et les hyper paramètres des RPV-MV et RVV-MV sont entièrement basés sur des expériences contrôlées plutôt que d'être choisis manuellement.

Pour témoigner de la capacité du RDV-MV à effectuer une détection de visage, nous avons mené des expériences approfondies sur deux ensembles de données de référence de détection de visage difficiles: les jeux de données WIDER FACE et FDDB. Le RDV-MV réalise constamment de meilleurs résultatsque l'état de l'art. Plus précisément, sur la partie la plus difficile, du jeu de données WIDER FACE qui contient des visages aussi petits que 9 pixels et aussi grands que 1000 pixels de hauteur, RDV-MV surpasse le précédant meilleur résultat par 9.8% pour la précision moyenne. Cela démontre que RDV-MV est un détecteur de visages viable et précis pour la détection de visages sans contrainte, en particulier dans les visages avec une grande variation de tailles.

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my supervisor, Professor Martin D. Levine, for his persistent support, encouragement and inspiration to my research and thesis. During the past three years, Prof. Levine has always been inspiring me to keep up with the top technology in computer vision, to pay close attention to every detail in research, and to maintain a reasonable balance between work and life. These invaluable instructions and advice have benefited me throughout the preparation of this thesis, and will continue benefitting me for a whole life.

I would also like to acknowledge Prof. Martin D. Levine, Prof. Tal Arbel, Prof. Doina Precup, Prof. Joelle Pineau, and Prof. Hannah Michalska, for offering me courses in the field of computer vision, machine learning and optimization. These excellent courses laid a solid foundation for me to complete this thesis.

Meanwhile, I would like to thank my friends and lab colleagues, in particular, Fanxiang Zeng, Qing Tian, Vikram Vedala and Andrew Hebb. My interactions with them always led me to new ideas. In addition, I greatly appreciate Auguste Lalande and Chuan He for their help with my French abstract.

Finally, and most importantly, I dedicate this thesis to my loving parents for their consistent understanding and support during my studying in Canada.

- To my parents.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Although face detection has been extensively studied during the past two decades, detecting unconstrained faces in images and videos has not yet been convincingly solved: most classic learning methods and recent deep learning methods tend to detect faces where fine-grained facial parts (e.g., eyes, nose and mouth) are clearly visible. This property negatively affects their detection performance in the case of faces at low-resolution or out-of-focus blur, which are common issues in surveillance camera data. The lack of progress in this regard is largely due to the fact that current face detection benchmark datasets (e.g., FDDB [1], PACAL FACE [2] and AFW [3]) are biased towards high-resolution face images with limited variations in scale. Recently, a new dataset, WIDER FACE [4], has been published as a potential face detection benchmark. This database consists of 32,203 images with 393,703 labeled faces. It is the largest publicly available database to date for face detection research. Images in WIDER FACE also have the highest degree of variation in scale, pose, occlusion, lighting conditions, and image blur, as shown in Figure 1.1.

As indicated in the WIDER FACE report [4], of all the factors that affect face detection performance, scale is the most significant one. Specifically, when using EdgeBox [5], a state-of-the-art object proposal approach, for discovering potential faces in WIDER FACE dataset, the detection rates consistently stay below 30% for the small scale faces (between 10-50 pixels in height), even if 10,000 proposals are used for each image. The WIDER FACE report [4] further presented the detection performance of four well-known face detection algorithms on small scale faces: The Viola-Jones (VJ) face detector [6] achieved only 4%

Fig. 1.1 An example of face detection results on the WIDER FACE dataset [4] using the MP-FDN method proposed in this thesis. We observe that it can robustly detection unconstrained "hard faces" with large variations in scale, pose, occlusion, lighting conditions, and image blur.

Average Precision (AP), the Deformable Part Model (DPM) face detector [7], 5.5% AP, the Aggregated Channel Features (ACF) face detector [8], 11.5% AP, and Faceness-net [9], 12% AP.

With the above result, we can arguably say that the so-called handcrafted features[1] used in first three methods, that is, VJ [6], DPM [7] and ACF [8] face detectors are not sufficiently informative to represent small-scale facial patterns. However, it is surprising that Faceness-net, a recently proposed deep convolutional network model, also cannot achieve promising performance on small-scale faces. This result contradicts the commonly held assumption in the machine learning community that deep convolutional networks (ConvNets) possess an ability to automatically learn informative features and achieve high classification accuracy based on these features. Therefore we must ask the following:

Question 1: What is the reason behind the phenomenon that "tiny faces" cannot be accurately detected by ConvNets?

Question 2: Is there any way that we can adapt the deep learning framework so as

[1] VJ employs Haar-like features, DPM utilizes HOG features, and ACF uses a combination of multiple simple channel features, such as HSV color and gradient magnitude channels.

to detect tiny facial patterns with high accuracy?

This thesis aims to investigate these two questions. Our investigation suggests a detailed answer to Question 1, and a "YES" answer to Question 2. To further verify the "YES" answer, we have proposed a Multi-Path Face Detection Network (MP-FDN) to detect both tiny and big faces with high accuracy. At the same time, it is noteworthy that by virtue of the abundant feature representational power of deep neural networks and the employment of contextual information, our method also possesses a high level of robustness to variations in pose, occlusion, illumination, out-of-focus blur and background clutter.

MP-FDN is composed of two stages: face proposal and face verification, as shown in Figure 1.2.

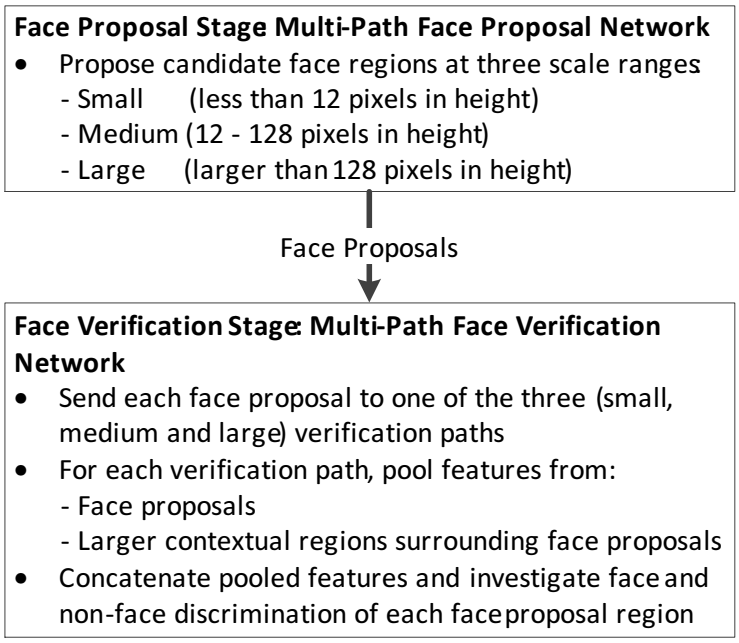


Fig. 1.2 Two stages of the proposed Multi-Path Face Detection Network (MP-FDN).

In the face proposal stage, a Multi-Path Face Proposal Network (MP-FPN) proposes faces at three different scales[2]: small (less than 12 pixels in height[3]), medium (12-128 pixels in height) and large (larger than 128 pixels in height[4]). These scales cover the majority of

[2]By scale we refer to the size of a square box surrounding a face.
[3]In practice, we found the proposed MP-FDN can detect faces as small as 6 pixels in height.
[4]In practice, we found the proposed MP-FDN can detect faces as large as about 610 pixels in height.

faces available in all public face detection databases, e.g., WIDER FACE [4], FDDB [1], PASCAL FACE [2] and AFW [3]. For each input image, MP-FPN outputs a set of bounding boxes containing candidate face regions and a set of corresponding scores indicating the so-called "faceness" probabilities of each face candidate. As will be seen in Chapter 3, MP-FPN can serve as a strong stand-alone face detector. However, in order to further remove difficult false positives while including difficult false negatives, we add a second face verification stage.

In the face verification stage, a Multi-Path Face Verification Network (MP-FVN) first sends each proposal to one of the three verification paths according to the same scale partitions as in MP-FDN. Then the corresponding verification path pools features from both the face proposal region (provided by MP-FPN) and a larger contextual regions surrounding the face proposal. These two features are then concatenated as the final feature to discriminate the face proposal as face or non-face.

1.2 Thesis Contributions

This thesis makes two contributions.

First, this thesis investigates the reason behind the phenomenon that tiny facial patterns cannot be accurately detected by deep convolutional networks (ConvNets). With a series of controlled experiments, we have been able to establish a rule for the sensitivity of scale for individual layers in the ConvNets. That is, *lower-level* convolutional layers with higher-resolution feature maps are most sensitive to small-scale facial patterns, but almost agnostic to large-scale facial patterns. This is due to the limited size of the receptive field. Conversely, *higher-level* convolutional layers with lower-resolution feature maps respond strongly to large-scale facial patterns while ignoring the small-scale patterns. Consider the well-known VGG16 ConvNet model [10] as an example. For a face of a height within 5-15 pixels, the conv3 layer has the highest detection accuracy among all convolutional layers. Even if their feature maps are extrapolated to the same spatial resolution as conv3s feature maps, the conv4 and conv5 layers cannot achieve as high detection accuracy as the conv3 layer. Notwithstanding this phenomenon, most previous VGG16-based object detection and face detection methods make predictions solely based on feature maps of the conv5 layer, which is not an optimal approach for detecting tiny object/face patterns. We believe

For an image containing even larger faces, we can down-sample it to detect these faces.

that the results of our investigation can benefit the future research on small-scale object detection, such as face and pedestrian detection.

Second, this thesis proposes a new deep convolutional neural network model, Multi-Path Face Detection Network (MP-FDN) for unconstrained face detection. MP-FDN achieves state-of-the-art detection performance on both the WIDER FACE [4] and FDDB [1] datasets. In particular, on the most challenging so-called "hard partition" of the WIDER FACE test set that contains mostly small faces, we outperform the former best result by 9.8% for the Average Precision.

1.3 Thesis Organization

The rest of the thesis is organized as follows. Chapter 2 is a literature review of the background related to face detection research, including a review of general object detection, face detection, and a summary of available benchmarks for face detection. Chapter 3 conducts a series of controlled experiments to investigate the layer-wise scale sensibility of deep convolutional networks. Chapter 4 illustrates the details of the proposed Multi-Path Face Detection Network (MP-FDN). Chapter 5 presents experiments that compare the proposed MP-FDN with other state-of-the-art face detection algorithms on the WIDER FACE [4] and FDDB [1] datasets. We present a detailed analysis of MP-FDN's robustness to various factors: scale, blur, occlusion, illumination, head pose, and facial expression. Chapter 6 concludes the thesis and proposes future work.

Chapter 2

Literature Review

Face detection is the task of finding and locating faces in an image, while general object detection is to find and locate all the instances of one or more categories of objects in an image. It is obvious that face detection can be regarded as a special case of general object detection with a single object category – human face – to be detected. In reality, many face detection algorithms were inspired by the methodologies employed in general object detection. Recently, with deep learning methods being successfully applied in the field of object detection, they have provided new insights into designing high performance face detection algorithms.

In view of the strong affinity between general object detection and face detection, in the first section, we review the recent developments regarding general object detection. We concentrate on how to detect objects at different scales which may give insight to the research topic of this thesis face detection across a large span of scales. In the second section, we summarize the development of face detection methods over the past decade, and when necessary, link them with related general object detection methodologies. In the third section, we discuss commonly used face detection benchmark datasets and their properties. Lastly, we present an overall analysis of the related work. Based on this analysis, we propose the research plan of this thesis, which will be discussed in Chapter 3 and Chapter 4.

2.1 General object detection

Object detection systems can be viewed as image classifiers that are repurposed to perform detection tasks. To be specific, an object detection system uses a classifier for a certain specific object and evaluates it at different scales and locations in a given image. In terms of how these scales and locations are visited, general object detection methods can be broadly classified into three categories:

1. *Sliding window approaches* re-sample an input image into an image pyramid and then densely scan the image pyramid at evenly spaced locations with the sliding window. The characteristic of Objectness is evaluated within each window by an object classifier.

2. *Region proposal approaches* first employ a region proposal method in an image to generate candidate bounding boxes that potentially enclose objects, and then run object classifiers only on these proposed boxes. Clearly, there is a risk factor in this case since not all pixels in the image are visited by this method.

3. *Proposal-free approaches* utilize a set of prior boxes (often some large non-overlapping grids that cover the whole image) to replace region proposals. These boxes are regressed to more precise bounding boxes that tightly enclose potential objects, and at the same time, classified as object/non-object by virtue of deep convolutional neural networks. Table 2.1 summarizes the three categories by indicating representative approaches for each category.

Table 2.1 Categorization of general object detection methods

Category	**Representative Approaches**
Sliding window	DPM [11], Aggregated Channel features [12], Overfeat [13]
Region Proposal	
- Classical region proposal	R-CNN [14], SPP-net [15], Fast R-CNN [16], ION [17], MultiPath network [18]
- CNN-based single-path region proposal	DeepMultiBox [19], Faster R-CNN [20], PVANet [21], R-FCN [22]
- CNN-based multi-path region proposal	MS-CNN [23], FPN [24]
Proposal-free	YOLO [25], SSD [26]

2.1.1 Sliding window approaches

A representative of using a sliding window based object detection is the well-known Deformable Parts Models (DPM) [11]. In DPM, each object category consists of a mixture of star-structured models. Each star-structured model has a root filter that approximately covers an entire object, as well as several higher resolution part filters that cover smaller parts of the object. The object detection process procedes as follows. Given an input image, a standard image pyramid is obtained via repeated smoothing and sub-sampling. Then a HOG feature map [12] is computed from each level of the image pyramid, thus forming a feature pyramid. Finally, objects are detected by computing the appearance and deformation scores of the root and part filters at each position and scale of the HOG feature pyramid. This mixture of deformable part models methodology can usually well account for various views of an object as well as the appearance variations within each view, thus achieving high detection performance. However, the calculation of a feature pyramid is quite time-consuming. For example, to construct a feature pyramid of three scales (e.g., 1, $\frac{1}{2}$ and $\frac{1}{4}$ of the original image size, respectively) that engages two octaves[1], DPM needs to compute 21 feature maps[2].

Dollar et al. [27] went a step further to accelerate the feature pyramid computation. They only perform an exact computation for one feature map at each octave. All other feature maps are approximated via extrapolation from the neighboring feature maps of nearby scales. Then a so-called aggregated channel feature (ACF) pyramid is computed from this approximated image pyramid. ACF is a combination of multiple visual cues, including HOG features, normalized gradient magnitude, and LUV color channels. Finally, a scan window slides over each position of this ACF pyramid, in this case, used for object/pedestrian detection. Aggregated channel features can be efficiently computed and are more informative than the single HOG features used in the DPM detector. However, both ACF and HOG features are handcrafted[3] features which may unintentionally filter out some informative visual cues.

Overfeat [13] is one of the early works that utilize a deep learning framework for object

[1] An octave is the interval between one scale and another with half or double its spatial resolution.

[2] There are 10 levels in each octave of a DPM. This means that we need to go down 10 levels in the pyramid from a certain feature map to obtain a feature map computed at twice the resolution.

[3] This is a term that is used in the literature to distinguish features computed by a Deep Neural Network from features that are arbitrarily selected by the person writing the computer program. It has nothing to do with either hands or crafts!

detection. The Overfeat network consists of two stages, feature extraction and classification. In the feature extraction portion, a set of convolution and subsampling filters extract features at each location and scale of an image pyramid to generate feature maps. Classification is accomplished by a sub-network of regressors and classifiers. These are simultaneously computed across all positions in the feature maps to obtain bounding boxes and their corresponding confidence scores. Nearby bounding boxes are then merged via a greedy merging strategy to obtain final object predictions. Unlike DPM [11] and ACF [27], these filters are not assigned to any particular semantic meaning beforehand, but automatically acquire informative cues during the training process. Furthermore, at each image scale, the filters are convolved across the entire image in one pass in order to generate feature maps. This avoids an explicit scan process as in DPM [11] and ACF [27]. However, to detect an object, the regressor and classifier will still need to densely sample all locations of all the feature maps.

Sliding window approaches share a common drawback that an object classifier needs to evenly and densely sample feature maps. This is normally quite a time-consuming process.

2.1.2 Region proposal approaches

Region proposal approaches provide a way to circumvent the above-mentioned drawback of sliding window approaches. Based on the fact that in most situations objects are sparsely located in natural images, region proposal methods first eliminate most obvious background regions using a simple binary-classification algorithm. It then applies a set of strong category-specific detectors to the remaining regions to detect instances of each object category. Region proposal approaches can be divided into three categories based on how they are generated: independent region proposal, CNN-based single-path region proposal and CNN-based multi-path region proposal. The first category employs a third-party algorithm (mostly classical learning methods) to propose object candidate regions. These regions are then classified by a detection network as object/non-object by a detection network. The region proposal algorithm and the following detection network do not share any computation, so we name it “independent” region proposal. The last two both utilize Convolutional Neural Networks (CNN) to generate region proposals. The region proposal network always shares convolution computation with the following detection network to improve computation efficiency. Single-path region proposal methods postulate object candidates by means

of feature maps of a single convolutional layer (hereafter abbr. for conv-layer), which is always the last conv-layer of a CNN framework. By comparison, multi-path region proposal methods feature maps use several intermediate conv-layers that are at different spatial resolutions. A detailed review of each category of region proposal approaches is as follows.

Independent region proposal approaches

Girshick et al. [14] proposed a Region-based Convolution Neural Networks (R-CNN), which is a seamless combination of classic learning and deep learning methods for object detection. R-CNN starts by generating a few thousands of category-agnostic[4] region proposals with the selective search algorithm [28] for each image. These region proposals, with arbitrary shapes, are warped to rectangles of a fixed size and then used by AlexNet [29] as a training set (See Figure 2.1). The resulting Deep Neural Network is employed extract deep features. A class-specific SVM classifier is then applied to classify these deep features as object or background. Finally, these same deep features are used by a class-specific bounding box regressor to provide more accurate localization for each object instance. R-CNN improves object detection accuracy by a large margin compared to previous classical learning methods. However, because R-CNN needs a forward pass through the convolutional network for each region proposal, it is very time-consuming.

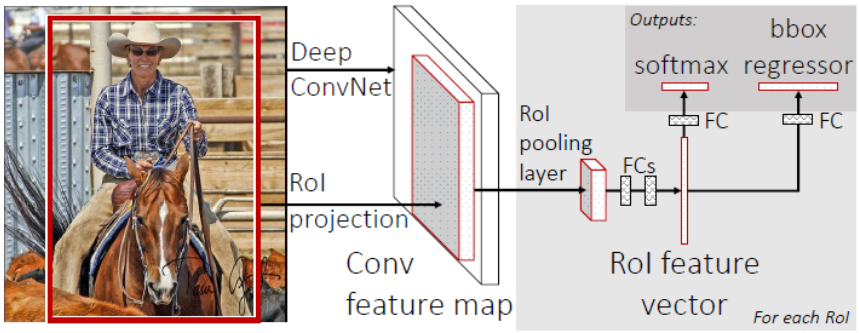


Fig. 2.1 An overview of R-CNN object detection system. Reprinted from Rich feature hierarchies for accurate object detection and semantic segmentation [14], by Girshirk et al., 2014, retrieved from `http://ieeexplore.ieee.org/` Copyright 2014 by IEEE.

Spatial pyramid pooling networks (SPP-net) [15] were proposed to speedup R-CNN by passing the entire image through CNN only once. The SPP-net first computes the

[4]By class-agnostic we mean the region proposals are selected as potential object.

convolutional feature maps for the whole image. Then spatial pyramid pooling is applied on each candidate window in the feature maps to pool a fixed-length deep feature vector for this proposal. These candidate windows are the original region proposals that are projected on feature maps (See Figure 2.2). Finally, as in R-CNN, the deep features are passed through a SVM classifier for object/background classification, followed by a bounding box regressor for localization regression. Compared to R-CNN, SPP-net has two significant advantages: (1) SPP-net computes convolutional feature maps for the entire image only once, thus avoiding a repeated computation of CNN features for each region proposal and leading to a 10 to 100× speedup when compared to R-CNN; (2) R-CNN needs to warp an region proposal to a fixed size which is then submitted to a CNN to obtain a fixed-length deep feature vector. In contrast, SPP-net can accept region proposals of arbitrary size and use spatial pyramid pooling to generate a fixed-length deep representation. However, we note that the conv-layers before spatial pyramid pooling deal with the whole image while the fully connected layers after spatial pyramid pooling deals only with region proposals (See Figure 2.2). Thus the network fine-tuning process cannot update the conv-layers that precede spatial pyramid pooling.

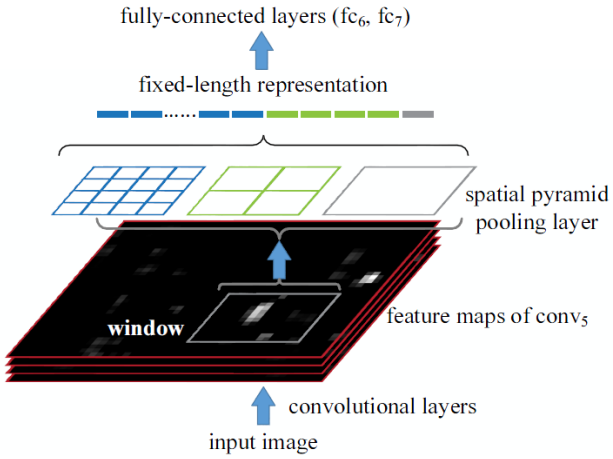


Fig. 2.2 An overview of SPP-net [15]. The feature maps are computed from the entire image. The spatial pyramid pooling is performed in candidate windows to obtain a fixed-length feature vectors. The candidate windows are the result of projecting original region proposals to feature maps. Reprinted from Spatial pyramid pooling in deep convolutional networks for visual recognition, by He et al., 2014, retrieved from `https://link.springer.com/chapter/10.1007/978-3-319-10578-9_23` Copyright 2014 by Springer.

Fast R-CNN [16] was proposed to address the above issue of SPP-net. An input image of arbitrary size is input into a fully convolutional network to obtain convolutional feature maps. At the same time, region proposals are projected to the feature maps to form a set of candidate windows. A regions-of-interest (ROI) pooling layer is then applied to pool each candidate window into a fixed-size feature map, which is later mapped to a feature vector by fully connected layers. The feature vector then passes through two sibling layers, one for object/background classification and the other for bounding box regression (See Figure 2.3). Fast R-CNN gets rid of the multi-stage pipeline (including three stages: feature extraction with CNN, object classification with SVM, bounding box regression with a linear regressor) in R-CNN [14] and SPP-net [15] . Instead, it combines feature extraction and object detection into a single network via ROI pooling layer. Also, by virtue of multi-task learning, object classification and bounding box regression can be done simultaneously within the network, and their respective losses can be back-propagated through the whole network to update all fully connected layers and conv-layers. These technical improvements enable fast R-CNN to gain higher detection quality than [14, 15] on the PASCAL VOC dataset [30]. However, fast R-CNN only uses information near an object's region of interest that was pooled from feature maps of the last conv-layer. As a result, its detection performance drops considerably on the MS COCO dataset [31], which contains a larger proportion of small-scale objects than the PASCAL VOC dataset [30].

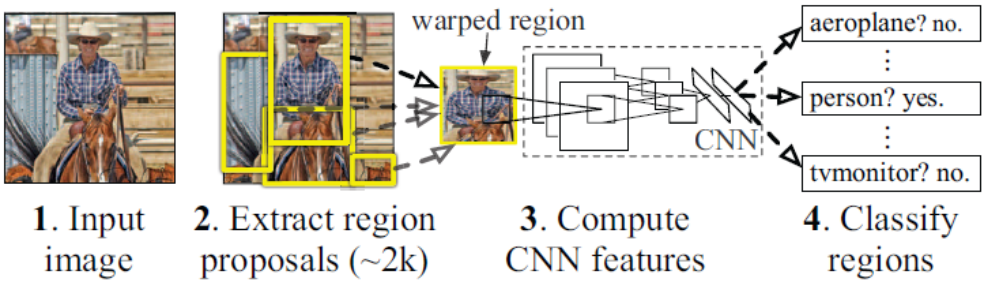


Fig. 2.3 An overview of Fast R-CNN architecture [16]. Reprinted from Fast r-cnn, by Girshick, 2015, retrieved from `http://ieeexplore.ieee.org/` Copyright 2015 by IEEE.

Several approaches have adapted fast R-CNN [16] to detect objects at a broad range of scales. For example, Inside-Outside Net (ION) [17] introduced multi-scale analysis and contextual representations to the fast R-CNN system. The authors first use ROI pooling

to extract features not only from the last conv-layer, but also several intermediate conv-layers that contain high-resolution visual information. Then two spatial Recurrent Neural Networks (RNNs) are applied after the last conv-layer to provide horizontal and vertical contextual information for the whole image, thus forming a contextual feature map. This is followed by the use of an ROI pooling layer that provides for contextual features in the image. Both the multi-scale and contextual features are L2-normalized and concatenated to produce a single feature vector for object classification and bounding box regression. The MultiPath Network [18] pools features from multiple conv-layers (same as [17]) for four region crops with fields-of-view of $1\times$, $1.5\times$, $2\times$ and $4\times$ of the original proposal box. The $1\times$ field of view is the only information used in fast R-CNN, while other three are newly added to explicitly include contextual information. This is an alternative to the spatial RNNs used in [17] for context representation. Due to the inclusion of multi-scale and contextual information, ION [17] and MultiPath Network [18] improve the detection quality of small-scale objects in the MS COCO dataset [31] by a large margin. However, they share a significant disadvantage, namely, that both training and testing are not end-to-end processes. This is because region proposals are mostly obtained by classical learning methods (e.g. selective search [28], EdgeBox [5]) that are independent from the following convolutional network. Moreover, as indicated in [32], region proposal generation dominates the processing time of the entire pipeline, thus posing a computational bottleneck.

The algorithms presented in the next two sub-sections manage to solve this problem by using CNNs for both region proposal and object detection.

CNN-based single-path region proposal approaches

Erhan et al. [19] proposed "DeepMultiBox", a deep neural network that generates a small number of object candidates in a class agnostic manner. During training, the authors first cluster the ground truth locations of training data and obtain K clusters to be used as priors for predicted locations. An optimal assignment problem is then solved for each training image so that each true object box in this image is assigned to its nearest prior predication. These assigned prior predications are used as positive samples and the rest as negative. Finally, the network is trained on these data with a multi-task loss function. This function consists of an L2 distance loss between coordinates of each positive sample and its ground truth, and a cross-entropy loss for each positive/negative prediction. During testing, an image plus the K prior predictions (same as those used in training) are input

to the trained network, which outputs an updated location as well as a confidence score for each prior prediction. The top 10 highest scoring detections are kept and classified by a class-specific deep neural network. DeepMultiBox is one of the earliest works that applied a CNN for generating object proposals. However, a separate CNN that does not share any computation with DeepMultiBox is required to classify these proposals. Also, it may miss some potential objects by only using a predefined number of prior predictions.

Ren et al. [20] proposed Faster R-CNN that shares convolutional features between a class-agnostic Region Proposal Network (RPN) and the fast R-CNN [16] object detector mentioned earlier. An RPN replaces the usage of sparse prior predictions in [19] with a dense prediction of all positions in a feature map. At each feature map location, k region proposals of different scales and aspect ratios are predicted. They are parameterized relative to k reference boxes, called anchors. Each anchor is centered at a given position and associated with a scale and aspect ratio. The anchors are necessary because they refer to both the scale and position information so that objects of different sizes located in any position of an image can be detected by an RPN. Moreover, the RPN shares all convolutional layers with fast R-CNN. This design help reduces the region proposal time by a considerable margin when compared to the selective search algorithm [28] used in [16].

Recently, Kim et al. introduced PVANet [21], which enhanced the faster R-CNN system with three modifications. First, a C.ReLU unit [33] was added to after each conv-layer in the first three stages in order to reduce the computational cost. Second, Inception structures [34] were applied to replace the normal 3×3 conv-layers of the last three stages. An Inception unit combines conv-layers of different sizes, e.g. 1x1, 3x3 and 5x5, which correspond to different receptive fields that can better capture visual patterns of various scales than a single 3×3 conv-layer. Third, the feature maps of both the last conv-layer and two intermediate conv-layers are rescaled to the same spatial dimension and then concatenated for both region proposal and object classification. These skip connections are similar to those used in [17, 24]. The philosophy of this design is that a combination of fine-grained details (intermediate conv-layers) with highly abstracted information (the last conv-layer) can improve both region proposal and classification networks when detecting objects of different scales. With these modifications, PVANet is able to achieves superior detection performance than faster-RCNN [20] on PASCAL VOC dataset [30].

Dai et al. [22] have proposed Region-based Fully Convolutional Network (R-FCN) that replaces the ROI pooling in the classification sub-network of [20] with a position-sensitive

ROI pooling layer. This layer pools features from position-sensitive score maps produced by feature maps of the last conv-layer. The pooled features are directly used by a softmax layer for object classification. The position-sensitive ROI pooling improves computation efficiency of the classification sub-network by eliminating time-consuming fully connected layers from it. Moreover, the so-called atrous convolution trick [35] is used on last stage of conv-layers to increase the spatial resolution of their features maps so as to increase the detection accuracy of small objects.

In summary, Faster R-CNN and its descendants have achieved state-of-the-art object detection performance on the PASCAL VOC [30] and MS COCO [31] datasets. However, both RPN and fast R-CNN sub-networks are based on feature maps at a single spatial resolution. Furthermore, skip connections and atrous convolution [35] are used in [21] and [22], respectively, in order to maintain fine-grained details. Nevertheless, the stride of the single-resolution feature maps still makes it difficult to detect small objects[5].

CNN-based multi-path region proposal approaches

Cai et al. [23] proposed a multi-scale CNN (MS-CNN) that created multiple forked processing. Hence candidates are simultaneously passed through three parallel region proposal branches. These branches emanate from different conv-layers of a CNN trunk. On the one hand, small objects can be proposed from the branch that emanates after an intermediate conv-layer with a small stride and fine-grained details. On the other hand, large objects can be proposed from the branch that emanates after the final conv-layer with a large stride and highly abstracted information. Finally, a detection sub-network is employed to process and classify region proposals into different categories. As might be expected, MS-CNN shows better detection performance than faster RCNN [20] when detecting small objects, such as pedestrians and cars at a distance.

Lin et al. [24] have noticed that a multi-scale, pyramidal hierarchy of deep convolutional networks is a natural source for constructing a deep feature pyramid for object detection. They created such a feature pyramid in a top-down manner by utilizing feature maps of the last conv-layer of each convolutional stage[6]. For a given convolutional stage, the feature

[5]Faster R-CNN, PVANet and R-FCN all have a stride of 16 pixels w.r.t the input image, meaning that a 16x16 object in the original image is represented by only one node in the feature maps of last conv-layer. Therefore, an object smaller than 16x16 can hardly be detected.

[6]A convolutional stage is a stack of several conv-layers that are of the same spatial resolution. Two nearby convolutional stages are connected by a sub-sampling layer (e.g. a max pooling or average pooling

maps of its last conv-layer are first convolved with a 1×1 conv-layer to reduce the channel dimension to 256. The convolved feature maps are then added to the de-convolved ($\times 2$) feature maps of nearby higher-level convolutional stage in an element-wise manner. The resulting feature maps serve as a layer in the final feature pyramid. RPN and the fast R-CNN sub-network in [20] are consecutively applied to each layer of the generated feature pyramid to propose object candidates and classify them. Small objects are proposed and classified by low-level high-resolution layers, while large objects by high-level low-resolution layers. This feature pyramid network (FPN) achieve higher detection accuracy than [20] when detecting small and medium-size objects in the MS COCO dataset [31].

Multi-path region proposal approaches provide a natural way for utilizing the multi-scale feature maps of a CNN to detect objects at multiple scales. However, we note that neither [23] nor [24] discussed how to choose the scale range for each proposal branch.

2.1.3 Proposal-free approaches

Even if faster R-CNN [20] and its variations [21, 22, 23, 24] are able to reduce the region proposal overhead by sharing convolutional computations between region proposal and classification sub-networks, both training and testing still contain multiple steps. Proposal-free approaches eliminate the region proposal sub-network and thus make training and testing of the object detection network a single-step process.

Redmon et al. [25] proposed You Only Look Once (YOLO) a single CNN network that models object detection as a regression problem. The approach first resizes an input image to a fixed size (448×448), and then runs a convolutional network[7] over the image to obtain a $7 \times 7 \times 30$ tensor of predictions (See Figure 2.4). This tensor corresponds to a 7×7 evenly divided grid of the input image. Each grid cell predicts two bounding boxes and their class probabilities. Thus the complete tensor predicts 98 bounding boxes per image and class probabilities for each box. Because of the removal of the separate region proposal step, YOLO enjoys a speedup in detection efficiency. However, as the authors have reported, it suffers from more localization errors and fails to detect small objects[8]. The high miss

layer).

[7]This network consists of six stage of conv-layers and two fully connected (fc) layers. The $7 \times 7 \times 30$ tensor is the output of the second fc layer.

[8]Note that it may be possible to tweak the YOLO framework to detect smaller objects. For example, this could be achieved by enlarging the default image size (448×448) or dividing the input image into a grid denser than 7×7. However, this is beyond the scope of the original paper.

rate for small objects is caused largely by the 7×7 coarse division of the input image, which cannot account for small objects less than the size of a single cell (64×64). The large localization error is mostly due to the usage of a fully connected (fc) layer to generate the tensor. The fc layer combines global information of the whole image which obviously cannot regress the position of a local grid cell with high accuracy.

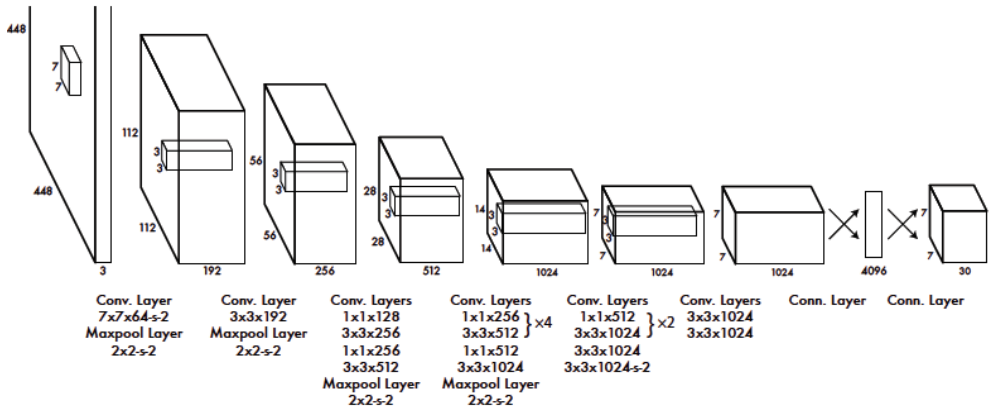


Fig. 2.4 An overview of the YOLO architecture [25]. YOLO consists of six stages of convolutional layers (short for Conv. Layers in figure) followed by two stages of fully connected layers (short for "Conn. Layer" in figure). Reprinted from You only look once: Unified, real-time object detection, by Redmon et al., 2016, retrieved from `http://ieeexplore.ieee.org/` Copyright 2016 by IEEE.

The Single Shot Multibox Detector (SSD) [26] provides a solution to the above two issues. First, SSD evaluates default boxes of different aspect ratios at each location of several feature maps. These feature maps are of a different spatial resolution that is generated by conv-layers at multiple stages (See Figure 2.5). This is very similar to the MS-CNN [23] mentioned above. However, these default boxes are not used for proposing region candidates, but directly for object classification and bounding box regression. Since they originate from feature maps at different resolutions, they can better predict both small and large object than a fixed 7×7 grid used in [25]. Second, the SSD network is fully convolutional, so that each default box is classified and regressed by the local information surrounding the box. This leads to more precise object localization than [25].

Proposal-free methods provide simpler and more efficient ways for object detection. Nevertheless, the default grid partition [25] or default boxes [26] are, after all, very simple

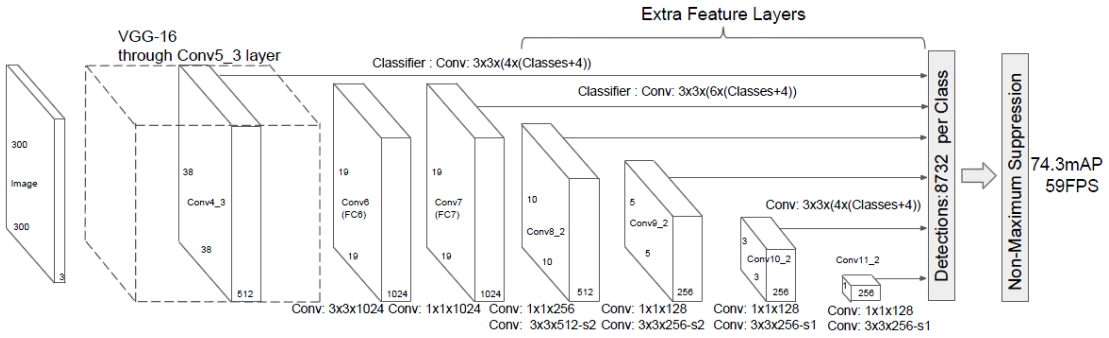


Fig. 2.5 An overview of the SSD architecture [26]. Reprinted from SSD: Single shot multibox detector, by Liu et al., 2016, retrieved from `https://link.springer.com/chapter/10.1007/978-3-319-46448-0_2` Copyright 2016 by Springer.

object priors. They generally have large position offsets and shape inconsistency with respect to ground-truth object boxes. This leads to heavy location regression burden to the object detector. In contrast, the object proposals provided by a RPN [20] are more complex object priors that match ground-truth object boxes well and lead to more accurate object localization. As a result, both YOLO [25] and SSD [26] are inferior to region proposal-based approaches [20, 21] in terms of average precision on the PASCAL VOC dataset [30].

In summary to this section, the current state-of-the-art of general object detection indicates that region proposal approaches demonstrate the best performance with an efficient end-to-end network structure. This is particularly the case for multi-path region proposals, which have showed their potential to detect small-scale objects. However, these are still equal or larger than the size of 32×32. Moreover, authors of these publications have not provided any details of how to actually select the appropriate scale range for each proposal branch. Consequently, in order to deal with these issues, the research presented in this thesis will propose the method Multi-Path Face Detection Network (MP-FDN). This approach belongs to the family of multi-path region proposals but includes two major improvements. First, we provide a detailed and systematic way to select the optimal scale range for each proposal branch. Second, we extend the lower bound of the object size from 32×32 to 8×8, so that faces as small as 8×8 and as large as 800×800 can be detected simultaneously with high accuracy.

2.2 Face detection

Face detection is a binary classification problem that classifies each image patch of a static image[9] as face or non-face. It has been extensively studied over the past two decades: when searching the key words "face detection" in the Google Scholar academic search engine, more than three million results are produced[10]. We review only the most representative works. Chronologically, face detection methods can be classified into two major categories:

1. *Classical learning approaches* densely sample an input image with a sliding window. At each sampling position, so-called handcrafted features are extracted and evaluated by a classic learning method. According to how the facial features are modeled, this category can be further divided into two sub-categories: deformable part and rigid template approaches. Classical learning methods have dominated the face detection literature from the late 1990s to 2014.

2. *Deep learning approaches* use either a sliding window or a region proposer to sample an input image. The sampled image patches are then fed to a deep convolutional neural network (CNN) for feature extraction and face/non-face classification in an end-to-end manner. Deep learning approaches have dominated the face detection literature since 2015. Table 2.2 summarizes the two categories by indicating their representative approaches.

2.2.1 Classical learning approaches

There are two established sets of methods for this category of approaches, one based on deformable parts models and the other on rigid templates. The former models facial organs (e.g. eyes, nose and mouth) as a set of deformable parts, which add robustness to a face detector with respect to partial occlusion and facial expression changes. The latter models the whole face by a set of rigid templates. Though it is less flexible than a deformable parts model, we will shortly see that good feature selection and training strategies can achieve equally high performance.

Deformable parts models

Inspired by the successful application of Deformable Parts Model (DPM) in generic ob-

[9] Face detection can also be used in videos, but this thesis focuses solely on its application in static images.

[10] As of March 1, 2017.

Table 2.2 Categorization of face detection methods

Face Detection Category	Representative Approaches
Classical learning method	
- Deformable parts models	Zhu and Ramanan [3], structural model [2]
- Rigid template	Viola-Jones detector [6], NPD [36], Joint cascade [37], Headhunter [7]
Deep learning method	
- Sliding window	DDFD [38], CNN cascade [39], Multi-task cascade [40]
- Region proposal	Hyperface [41], STN [42], Faceness-Net [9], CMS-RCNN [43]

ject detection [11], Zhu and Ramanan [3] proposed a mixture of tree-structured deformable models to jointly perform multi-view face detection, pose estimation and facial landmark detection. In their framework, each part represents a facial landmark, and the part appearances and spatial relationship between a pair of parts is integrated into the cost function to infer whether or not a region contains a face.

Yan et al. [2] went a step further by introducing two complementary structural deformable models. The first captured both appearance and shape variations in facial regions, while the second was designed to capture the co-occurrence between the face and other body parts. The latter was said to add robustness to the face detector in case of heavy occlusion.

Face detectors based on deformable parts enjoy high detection accuracy and require less training data. However, these data require a laborious annotation of the facial landmarks in different poses. Moreover, because of the usage of fine-grained facial landmark information, these methods are not suitable for tiny or blurred face detection.

Rigid templates models

Rigid templates for face detection appeared in the literature much earlier than DPM, and provoked the publication of many more variants. One of the most influential works in this direction was the Viola-Jones (VJ) face detector [6]. This paper applied a cascade of boosting decision stumps on simple and efficient Haar-like features to achieve real-time

frontal and near-frontal face detection. The idea of efficient extraction of simple features and boosting cascades inspired a number of papers [36, 37, 44].

Li et al. [44] utilized SURF instead of Haar-like features to reduce feature pool size and replaced the decision stumps with logistic regression classifiers so as to directly output a so-called "faceness" probability. Liao et al. [36] proposed a novel Normalized Pixel Difference (NPD) feature to capture facial appearance variations in unconstrained scenarios and employed a quadratic tree with a depth of eight instead of the decision stump to enhance the learning ability. Unsurprisingly, Chen et al. [37] experimentally determined that face alignment boosts face detection performance. Based on this finding, they learnt a mixed face detection-landmark regression decision tree with shape-indexed features within a boosted cascade framework. This approach showed promising results for both face detection and alignment.

Inspired by the favorable application of channel features in the domain of pedestrian detection and general object detection [45, 27], Yang et al. [8] introduced channel features within the VJ boosted cascade framework. The ensuing rich representation capacity of the multi-mode channel features helped achieve high accuracy for multi-view face detection. Mathias et al. [7] studied both rigid template and DPM face detectors. They showed that, given carefully chosen hyper-parameters, both "vanilla" DPM [11] and VJ detectors [6] armed with channel features [45] could achieve state-of-the-art performance.

Although the performance of face detectors has increasingly improved by the use of these classical methods, using handcrafted features and classic classifiers has stymied the development of seamlessly connecting these two steps in the computational process. This is because they require heuristically setting many hyper-parameters. For example, both [8] and [7] need to divide the training data into several partitions according to face pose and train a separate model for each partition.

2.2.2 Deep learning approaches

Inspired by the successful application of deep convolutional neural networks (CNNs) in image classification [29, 10] and object detection [14, 15, 16, 20], researchers began to adapt CNNs to the face detection task. Early works inherited the sliding window approach commonly used in the classical learning methods mentioned in the above sub-section. The only difference was that they replaced handcrafted features and classical learning algorithms

by a CNN structure to represent and classify each image patch in an end-to-end manner. More recently, even the sliding windows have been replaced by region proposals to increase computational efficiency. This is a similar developmental trend to general object detection.

Sliding window methods

Farfade et al. [38] proposed a single CNN model based on the AlexNet [29] to deal with multi-view face detection. Li et al. [39] used a cascade of six CNNs for alternative face detection and face bounding box calibration. Zhang et al. [40] went a step further by utilizing multi-task learning in a CNN cascade. They designed a cascade of three CNN stages, referred to as P-net, R-net and O-net, respectively. P-net is a simple CNN that scans through an image pyramid to find potential facial regions and eliminate most background regions. Then R-net and O-net, possessing an increasing complexity, are used for further eliminating hard negatives. All three sub-networks are learnt in a multi-task manner, where face classification, bounding box regression and facial landmark localization losses are simultaneously back-propagated to update the network parameters. Not only does the multi-task cascade help reduce the number of CNN stages (3 here compared to 6 used in [39]), but it also boosts face detection performance.

Nevertheless, in general, sliding window methods need to crop facial regions and re-scale them to specific sizes. This increases the complexity of the training and testing. Thus they are not suitable for efficient unconstrained face detection where faces of different scales coexist in the same image.

Region proposal based methods

Yang et al. [9] have suggested applying five parallel CNNs to propose five different facial parts, and then evaluate the degree of face likeliness by analyzing the spatial arrangement of responses of the facial parts. The use of facial parts makes the face detector more robust to partial occlusions, but like the DPM face detectors, as discussed earlier, this method can only deal with faces of relatively large size.

Ranjan et al. [41] proposed HyperFace, a deep multi-task learning framework that performs face detection, landmark localization, pose estimation and gender recognition at the same time. Similar to R-CNN [14], HyperFace first employs the Selective Search algorithm [28] to generate region proposals for faces in an image, and then uses AlexNet [29] to extract deep features for each region proposal. But unlike [14], HyperFace extracts “hy-

perfeatures" that combine feature maps from both two intermediate conv-layers and the final conv-layer to generate a fully connected feature vector. Five fully connected layers are then added in parallel to this feature vector for predicting individual task labels. As reported by the authors, the deep multi-task framework helps the features achieve a better understanding of faces, and thus leads to improvements in the individual tasks. However, similar to [9], the inclusion of the landmark localization task compromises the systems ability to detect small-scale faces. Also, feature maps of the two intermediate conv-layers need to be down-sampled to the same size as the feature maps of last conv-layer so that all of them can be concatenated. This may partially compromise their representation power for fine-grained details.

Two more recent face detectors [42, 43] are based on Faster R-CNN [20], which has been discussed in detail in the previous section. Chen et al. [42] have suggested a Supervised Transformer Network based on [20]. This so-called Transformer uses RPN of [20] to simultaneously propose faces and their associated facial landmarks. Following this, the candidate face regions are warped by mapping the detected facial landmarks into a set of canonical facial landmarks. Finally the warped face regions are verified by a RCNN. Naturally, the use of facial warping reduces face pose variations, thereby producing more accurate face detection results. However, it is questionable that this is effective for very small faces.

Zhu et al. [43] proposed a Contextual Multi-Scale Region-based CNN (CMS-RCNN), which extended Faster RCNN [20] in regard to two aspects. First, RPN was replaced by a Multi-Scale Region Proposal Network (MS-RPN) to propose face regions based on combined information from multiple convolutional layers. Secondly, a Contextual Multi-Scale Convolution Neural Network (CMS-CNN) was proposed for pooling features to replace RCNN. This is not restricted to the last convolutional layer, as in fast R-CNN, but may also originate from several lower level convolutional layers. In addition, contextual information was also pooled to promote robustness. Thus CMS-RCNN [43] has indeed improved RPN by combining feature maps from multiple convolutional layers in order to make a proposal. However, similar to HyperFace [41], in order to concatenate the feature maps of the last convolutional layer it is necessary to down-sample the lower-level feature maps. This design diminishes the network's discriminative power for small-scale face patterns.

In summary, deep learning approaches, due to their seamless concatenation of features and pattern classification, have outperformed all kinds of classical learning methods and

become the current trend for performing face detection. In particular, CMS-RCNN [43], a deep learning framework built on the Faster R-CNN [20], has achieved state-of-the-art performance in both WIDER FACE [4] and FDDB [1] datasets. However, in the most difficult hard partition of the WIDER FACE test set that contains mostly small-scale faces, CMS-RCNN only achieves an average precision (AP) of 64.3%. The research in this thesis will improves this situation by proposing a Multi-Path Face Network (MP-FDN). This is achieved by dividing the overall range of facial scales into separated and different partitions (e.g., small, medium and large) and choosing the optimal convolutional feature maps in each partition to propose and detect faces. We show that this novel methodology in the field of face detection can further improve the detection accuracy of small-scale faces while maintaining the high detection accuracy of the medium- and large-scale faces that has been achieved by previous deep learning methods.

2.3 Face detection benchmark datasets

This section summarizes nine benchmark datasets for face detection that have been commonly used in the past decade. We compare them in terms of the following five aspects:

(1) The numbers of images contained in a dataset

(2) The number of faces contained in the images of a dataset

(3) The proportion of different facial scales (in terms of face height according to [4], small: between 10-50 pixels, medium: between 50-300 pixels and large: over 300 pixels)

(4) Facial properties[11]

(5) Annotation style, as shown in Table 2.3.

Early face detection datasets, such as CMU-MIT [46] and CMU profile [47] collected gray-scale images with frontal and profile faces. Some of the faces had a large degree of in-plane rotation in order to test the rotation-invariance of a face detection algorithm. But these faces have relatively large size and little occlusion. The face height partitions are listed as "N/A" in Table 2.3 because these two datasets only provide facial landmarks that cannot be directly converted to face height. Nowadays, since color images have prevailed in most digital cameras and surveillance devices, these datasets are not used very often for

[11]The property "in the wild" in Table 2.3 means that faces have large variations in scale, pose, illumination, occlusion and background clutter. In other works, the face images are collected from real world situations. It is now a phrase frequently used in recent face detection literature.

benchmarking face detection algorithms.

GENKI-SZSL [48] is one of the early face detection datasets that contain mostly color images (there is still a small percentage of gray-scale images). Each image contains only one face, and the face is often centered and salient in an image. In contrast, the Annotated Faces in-the-Wild (AFW) dataset [3] is a collection of real world face images from Flickr[12] images. Faces in this dataset have large variations in scale, pose and illumination. But since AFW is also used for facial landmark localization, the size of the faces is relatively large. PASCAL Face [2] is another "in-the-wild" face detection benchmark collected from the PASCAL person layout dataset, a subset of PASCAL VOC dataset [30]. It contains more small-scale faces. However, both AFW and PASCAL FACE datasets only contain a few hundred images.

The FDDB dataset [1] contains 2,845 images with 5,171 faces collected from the Yahoo! News website[13]. These faces portray a wide range of difficulties including occlusions, difficult poses, low resolution and out-of-focus blur. The face annotation is provided by bounding ellipses rather than the commonly used bounding boxes. The authors claimed that an ellipse provides a more accurate specification than a bounding box without introducing additional parameters. FDDB is a much larger dataset for the evaluation face detection in the wild. However, like AFW [3] and PASCAL FACE [2], it only reports result for the whole dataset.

In order to support a fine-grained analysis of detection results, the Multi-Attribute Labeled Faces (MALF) dataset [49] was proposed. It not only contains more images and faces than FDDB, but also contains more annotated facial attributes, such as gender (male, female, unknown), face scale level (easy, medium, hard), pose deformation level (small, medium and large for each of yaw, pitch and roll), wearing glasses (true, false) and exaggerated expression (true, false). These attributes can assist a quantitative exploration of causes and correlation between different types of errors. The IJB-A dataset [50] was almost simultaneously proposed for benchmarking both face detection and face recognition. It contains 24,327 images with 49,759 faces. However, the faces are generally of large size in order to permit consideration of the face recognition task.

Recently, WIDER FACE [4] has been proposed to combine the merits of both MALF [49] and IJB-A [50]. First, it contains a very large number (32,203) of images with 393,703 faces.

[12] `https://www.flickr.com/`
[13] `https://www.yahoo.com/news/`

These images were collected from the Internet based on 61 public event categories, such as festivals, picnics, parades, etc. Currently this is the largest publicly available face detection dataset. Unlike IJB-A [50], WIDER FACE is solely intended for evaluating face detection algorithms. Therefore, a large proportion of small-scale faces are included, making it an excellent source for studying face detection across a large span of scales. Second, similar to MALF [49], WIDER FACE [4] has annotated additional facial attributes, including overall level of difficulty (hard, medium, easy), scale (small, medium, large), occlusion (heavily, partially, none) and pose (typical, atypical), in order to facilitate a fine-grained analysis of face detection algorithms. For each category of events in WIDER FACE, 40%, 10%, 50% data are randomly selected as training, validation, and testing sets, respectively. Therefore, WIDER FACE provides an effective source for both training and benchmarking a face detection algorithm.

2.4 Conclusion

In this chapter, we first presented a literature review of three categories of general object detection methods. The sliding window approaches can cover each position and scale of an image pyramid thus not missing any important visual cues. However, it is a time-consuming process. Region proposal approaches have improved the computational efficiency by using region proposal algorithms to eliminate most background regions and generate pertinent object proposals. Then powerful object classifiers need only to be applied to these proposals. Completely proposal-free approaches utilize default grids or boxes to replace region proposals in order to further boost computational efficiency. However, not surprisingly, they have shown larger object localization errors and inferior performance at detecting small objects. The current state-of-the-art of general object detection indicates that CNN-based region proposal approaches demonstrate the best performance with an efficient end-to-end network structure. This is particularly the case for multi-path CNN region proposals, which have shown their potential to detect small-scale objects.

We then reviewed two categories specifically designed face detection algorithms. Classic learning approaches, with various types of features and learning algorithms, have increasingly improved face detection performance over the past decade. However, they fail to seamlessly connect handcrafted features and the succeeding classifiers in the computational process and many hyper-parameters need to be set heuristically. During the past

three years, deep learning approaches have been developed to solve the above issues. They have boosted the face detection performance by a large margin by employing automatic feature extraction and a seamless concatenation of features and classifiers. In particular, similar to general object detection, CNN-based region proposal approaches have also achieved state-of-the-art performance in unconstrained face detection. However, detecting tiny faces with high accuracy is still an open issue. As a consequence, this thesis proposes a Multi-Path Face Detection Network (MP-FDN) to detect faces across a large span of scales with high accuracy. The main principle behind MP-FDN is that it is not necessary to employ only a single pathway through a Deep Neural Network (DNN) so that, essentially, the exact same algorithm analyzes each face detection category. Part of the analysis in the DNN may be shared by different pathways to deal with various face detection categories. Such a "forked" strategy permits giving special attention to certain classes. In the case of face detection, the latter is facial size.

MP-FDN is inspired by multi-path region proposal approaches used in general object detection, but includes two major improvements. First, we provide a detailed and systematic way to select the optimal scale range for each proposal branch of the fork. Second, we extend the lower bound of the object size from 32×32 to 8×8, so that faces as small as 8×8 and as large as 800×800 can be detected simultaneously with high accuracy. We will also show that, by virtue of the abundant feature representational power of deep neural networks and the employment of contextual information, our method also possesses a high level of robustness to variations in pose, occlusion, illumination, out-of-focus blur and background clutter.

According to the review of face detection datasets in Section 2.3, we choose the WIDER FACE [4] training set to train our MP-FDN face detector for the following three reasons. First, it contains 159,424 annotated faces collected from 12,880 images. This is currently the largest publicly available face detection training set. Second, faces in the dataset span a large number of scales. In particular, 50% of the faces have a height within 10-50 pixels and 43% within 50-300 pixels. This makes it a relatively balanced source of training data for creating a DNN detector capable of detecting both small- and large-scale faces. Third, images in this dataset are collected from 61 human activity events that possess various types of cluttered backgrounds, which can naturally be used as informative negative examples. As for testing, we have benchmarked our method on two large datasets, WIDER FACE [4] test set and FDDB [1] dataset. WIDER FACE test set consist of 16,097 images. As

previously mentioned, it has multiple facial attribute annotations, and thus supports a fine-grained analysis of face detection performance in terms of different factors, such as face size, poses, occlusion, etc. Although MALF [49] also supports fine-grained analysis, we do not use it because it only contains 5,250 images, much less than WIDER FACE test set. FDDB contains 2,845 images. Although it is relatively small and does not support fine-grained analysis, it was released much earlier (in 2010) and has been most frequently used as a face detection benchmark dataset during the past eight years. Therefore we will also benchmark the proposed MP-FDN on FDDB so as to make a fair comparison with most of state-of-the-art algorithms.

Table 2.3 Comparison of face detection benchmark datasets.

Dataset	#Image	#Face	Face Height (pixels)			Property	Annotation
			10-50	50-300	300-		
CMU-MIT [46]	130	511	N/A	N/A	N/A	Gray-scale, frontal	6 landmarks
CMU profile [47]	208	441	N/A	N/A	N/A	Gray-scale, frontal and profile	6 landmarks for frontal face, 9 landmarks for profile face
GENKI-SZSL [48]	3,500	3,500	31%	69%	0	Color & gray-scale, with salient face	Square box
AFW [3]	205	473	12%	70%	18%	Color, in the wild	Rectangle box, 6 landmarks, discretized viewpoint for pitch, yaw and roll direction
PASCAL FACE [2]	851	1,341	41%	57%	2%	Color, in the wild	Rectangle box
FDDB [1]	2,845	5,171	8%	86%	6%	Color & gray-scale, in the wild	Bounding ellipse
MALF [49]	5,250	11,931	N/A	N/A	N/A	Color, in the wild	Square box, detection difficulty level, gender, pose deformation level of pitch, yaw and roll, occlusion, wearing glasses, exaggerated expression
IJB-A [50]	24,327	49,759	13%	69%	18%	Color, in the wild with salient face	Rectangle box
WIDER FACE [4]	32,203	393,703	50%	43%	7%	Color, in the wild	Rectangle box, detection difficulty level, event class, occlusion, scale, pose

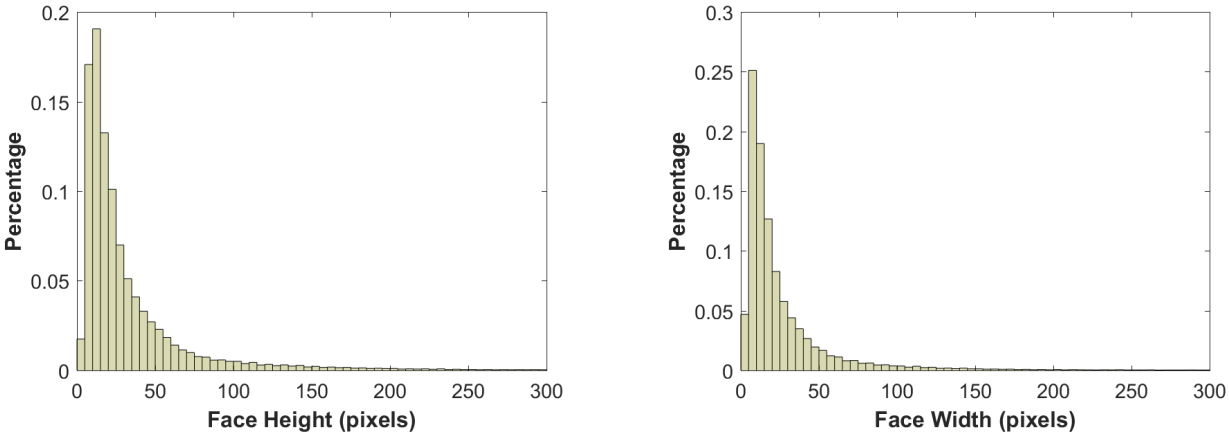
Chapter 3

Sensitivity to Scale According to Layer Level in a CNN

Multi-path region proposal methods for CNNs (e.g. FPN [24] and MS-CNN [23]) have achieved state-of-the-art performance in the field of general object detection, especially for detecting small objects. To our knowledge, such a "multi-path" methodology has never been applied specifically to face detection so that faces with large-scale variations could be detected in parallel. However, when directly transferring the available general multi-path region proposal methods from object detection to face detection, two problems arise. On the one hand, these methods have a lower bound of detection scale of around 32×32 pixels. Nevertheless, in the WIDER FACE dataset [4], 47% of the faces are within the range of heights 10-30 pixels and the face width has a similar distribution (see Figure 3.1a and Figure 3.1b). In other words, nearly half of faces can hardly be detected by these off-the-shelf multi-path object detection methods. On the other hand, both FPN [24] and MS-CNN [23] have not provided any details of how to select the appropriate scale range for each proposal branch. To avoid an intuitive but potentially suboptimal partition of scale range, as well as extend the lower bound of the object size from 32×32 to 8×8[1], we provide a detailed and systematic methodology for selecting an optimal scale range for each proposal branch. We begin this chapter with an introduction to the Faster R-CNN [20] framework,

[1]In this thesis, we set 8 x 8 as the smallest face scale that needs to be detected. Although there are faces in the WIDER FACE dataset that are less than 8 x 8, we find that most of them can hardly be discriminated even by human. Also, these faces are mostly annotated as "ignore", meaning they are not required to be detected.

followed by a series of controlled experiments to verify the scale-sensitivity variation of feature maps produced by different conv-layers. Next, a multi-path face proposal network is deduced from these experiments. Lastly, we present an overall conclusion.



(a) (b)

Fig. 3.1 Histograms of face height and width in the WIDER FACE dataset. In addition, about 1.17% faces have a height larger than 300, and about 0.64% faces have a width larger than 300. For viewing convenience, these are not included in the above histograms.

3.1 Overview of Faster R-CNN

The so-called Faster R-CNN [20] is a well-known object detection framework, which has been extended to multi-path object proposal approaches [23, 24] for general object detection. It has also had a major influence on the state-of-the-art of face detection approaches. Examples are the supervised transformer network [42] and CMS-RCNN [43]. Therefore, we employ Faster R-CNN as the basic architecture for a series of controlled experiments. Faster R-CNN is composed of two modules. The first is a Region Proposal Network (RPN), a fully convolutional network for generating object proposals. The second is a Fast R-CNN [16][2] object detector that consists of both convolutional layers and fully-connected layers. Fast R-CNN receives an image as well as its object proposals (provided by RPN) for classifying these proposals and regressing their positions. In essence, Faster R-CNN transfers the supervised pre-trained image representation for image classification to the object detection

[2]Distinguished from Faster R-CNN.

task. As a consequence, various pre-trained deep models for image classification can be employed in Faster R-CNN using transfer learning. Among them, AlexNet [29], ZF-net [51], VGG16 [10] and the recently proposed deep residual nets [52] are most commonly used. Table 3.1 provides a comparison of these models in terms of the number of convolutional layers and parameters.

Table 3.1 A comparison of common deep CNN models for image classification

Model Name	**#conv-layer**[3]	**#parameter**[4]
AlexNet [29]	5	4M
ZF-net [51]	5	4M
VGG16 [10]	13	15M
Res-50 [52]	53	24M
Res-101 [52]	104	38M

VGG16 is more complex than AlexNet and ZF-net, and achieved better classification accuracy in the ImageNet Challenge [53]. Although the recently proposed deep residual networks [52] have reported better image classification performance than VGG16, the commonly used residual networks, such as ResNet-50 and ResNet-101, possess a much higher network complexity and more parameters. Therefore, in view of the training and testing efficiency, we finally select VGG16 as the backbone of Faster R-CNN. The architecture of VGG16 is shown in Figure 3.2. It consists of five stages of convolutional layers (conv-layer), followed by three fully-connected (fc) layers and one softmax layer. Each conv-layer stage contains 2 or 3 conv-layers, and these conv-layers produce feature maps of the same spatial dimension. Two consecutive conv-layer stages are connected by a max-pooling layer for down-sampling feature maps. So the feature maps produced by the conv-layer stages 1-5 are 1, $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$ of the size of input image, respectively. The fc layers transform the convolutional feature maps into a 1000-dimensional vector, which is then passed through the softmax layer to form a probability distribution corresponding to 1000 object categories.

[3]Here "conv-layer" includes only convolutional layers but not fully connected (fc) layers. Although an fc layer can be viewed as a special convolutional layer with a spatial dimension of 1×1, it is eliminated from region proposal network (RPN). Since the experiments in this chapter are based mainly on RPN, we exclude fc layers when comparing different models.

[4]These are the number of parameters in the convolutional layers. All numbers are approximated to one million bit (M).

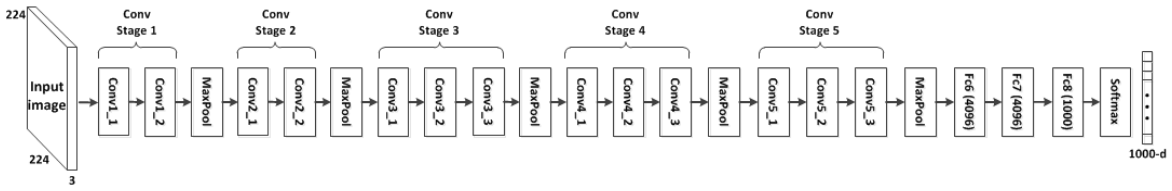


Fig. 3.2 The architecture of VGG16.

The two major modules of Faster R-CNN, RPN and Fast R-CNN are depicted in Figures 3.3 and 3.4, respectively. Both of them are adapted from VGG16. In RPN, the last max-pooling layer, three fc layers and the softmax layer are removed. Instead, a new 3×3 conv-layer ("Conv-proposal" in Figure 3.3) is directly added to "Conv5_3". This newly added conv-layer can be seen as sliding a 3×3 convolutional window over the feature map of "Conv5_3". The sliding window is fully connected to each 3×3 spatial position of the "Conv5_3" feature map to form a 512-dimensional vector. "Conv-proposal" is followed by two sibling 1×1 conv-layers: "Conv_cls" for generating object/non-object probabilities, and "Conv_reg" for predicting bounding boxes. At each sliding window location, k region proposals of different scales and aspect ratios are predicted at the same time. The k proposals are parameterized relative to k reference boxes, called anchors [20]. Each anchor is centered at the sliding window and associated with a scale and an aspect ratio. The anchors are necessary because they refer to both the scale, shape and position information. This ensures that objects of different sizes located at any position in an image can be detected by the convolutional network. The feature map of "Conv5_3" has a stride of 16 pixels with respect to the input image. Thus, for an input image of $H \times W \times 3$, "Conv_cls" and "Conv_reg" will together produce a feature map of dimension $\frac{H}{16} \times \frac{W}{16} \times (2k + 4k)$.

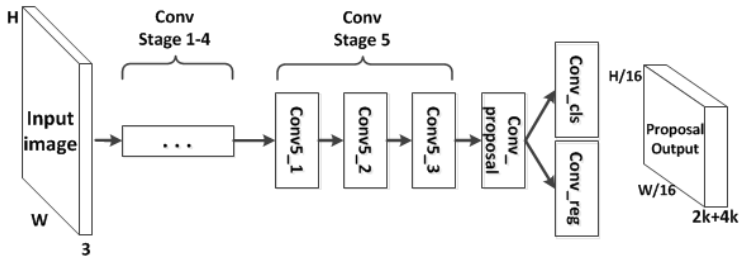


Fig. 3.3 The architecture of RPN. The conv-layers and max-pooling layers prior to the convolutional stage 5 have been omitted for the sake of brevity.

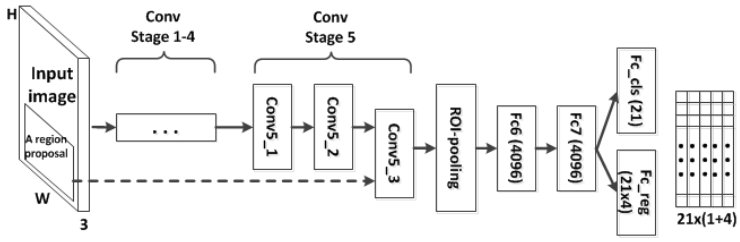


Fig. 3.4 The architecture of Fast R-CNN. The conv-layers and max-pooling layers prior to the convolutional stage 5 have been omitted for the sake of brevity.

The fast R-CNN module receives object proposals from RPN and maps them to ROI regions on the Conv5_3 feature map. For example, an h x w object proposal in an input image is mapped to an $\frac{H}{16} \times \frac{W}{16}$ ROI region in the corresponding position. An ROI-pooling layer is then used to pool features from each ROI region on the Conv5_3 feature map into a fixed-length vector. This vector goes through two fully connected layers and finally passes two sibling fully connected layers that generate class scores and bounding box coordinates[5].

3.2 Experiments on layer-level scale sensitivity

We point out that in Figure 3.3, RPN generates object proposals solely based on the last convolutional layer (Conv5_3), which poses two limitations. First, the feature map of Conv5_3 has a stride of 16 pixels w.r.t the input image, so that objects less than 16×16 are likely to be ignored. Second, Conv5_3 has a relatively large receptive field[6] (See Table 3.2), which is less sensitive to small-scale object patterns, as illustrated in [23].

Partially inspired by the multi-path object detection methods in [23, 24], we postulate that *conv-layers with large receptive fields (e.g. Conv4_3 and Conv5_3) are more sensitive to large object patterns, while those with small receptive fields (e.g. Conv2_2 and Conv3_3) are more sensitive to small-scale patterns.* In this section, we will verify this postulate and

[5]Faster R-CNN was originally evaluated on PASCAL VOC 2007/2012 detection benchmarks, which have 20 object categories. So there are totally 21 classes (20 object categories and 1 background category). For each class there are 4 bounding box parameters: two coordinates of the center of a box, and the width and height of the box.

[6]The "receptive field" of a conv-layer is the total number of pixels in the input image that contributes to the calculation of a node in the feature map of this conv-layer. Note that we only compare the last conv-layer of each convolutional stage since the deepest layer of each stage always has the largest receptive field in this stage.

Table 3.2 A comparison of the receptive field of conv-layers in VGG16

Conv-layer	**Conv1_2**	**Conv2_2**	**Conv3_3**	**Conv4_3**	**Conv5_3**
Receptive Field	5	14	40	92	196

also identify the optimal scale range for each conv-layer.

It is noteworthy that the receptive fields listed in Table 3.2 are not necessarily equivalent to the optimal scales, because they are merely the "theoretical" receptive field values calculated from the geometrical structure of the network. As argued in [54], a "theoretical" receptive field is generally much larger than an actual "effective" receptive field for a certain conv-layer. Therefore, we verify the aforesaid postulate as well as identify the optimal scale ranges through two groups of controlled experiments. The first group studies the scale sensitivity of individual conv-layers. This lays the foundation for the next group of experiments, where feature maps of nearby conv-layers are combined to extend the optimal scale range and increase the recall rate within the scale range.

3.2.1 Scale sensitivity of individual conv-layers

In this subsection, we investigate the scale sensitivity of different conv-layers. Initially, inspired by the so-called "network head" defined in [24], we define an *RPN head* as the structure of *a "Conv_proposal" followed by "Conv_cls" and "Conv_reg" for classification and regression.* A RPN head is actually the last three layers in RPN (see Figure 3.3).

Network architecture

We attach an RPN head to the end of each convolutional stage, i.e., Conv2_2, Conv3_3, Conv4_3 and Conv5_3, for proposing object regions (see Figure 3.5, 3.6, 3.7 and 3.8).

Besides the above four convolutional stages inherent in the VGG16 [10] framework, we additionally create a new conv-layer stage, the convolutional stage (conv-stage) 6. This stage consists of two conv-layers, Conv6_1 and Conv6_2. Conv6_1 is a $3 \times 3 \times 512 \times 1024$ convolutional layer, and Conv6_2 is a $1 \times 1 \times 1024 \times 256$ convolutional layer for feature map dimension reduction (see Figure 3.9). This stage is created to presumably improve the recall rate of large faces[7]. Similarly, we will use Conv6_2 to propose object regions.

[7]We noticed a small drop of recall rate between the last two scale ranges in Figure 3.11 (on Page 43), indicating that Conv5_3 may not be the best choice for detection large-size faces. So we created the

Note that using the last conv-layer of each convolutional stage is a natural choice because it is the deepest and thus the most informative layer of each convolutional stage, as argued in [24].

In addition, we avoid using Conv1_2 in this experiment due to its high resolution and large memory footprint. Moreover, since Conv1_2 is the first convolutional stage of the network, it is too "shallow" to learn representative features. Another notable issue is that feature maps of different conv-layers have different numerical ranges. For example, as reported in [55], the norm of Conv4_3 feature map is much larger than that of Conv5_3. Consequently, we have added an L2 normalization layer [55] in between the last conv-layer and the RPN head such that feature maps of different conv-layers are normalized to the same numerical range before proceeding to the region proposal process.

The criterion for comparing the above conv-layers is quite simple: given objects of a certain scale range, *the conv-layer (equipped with a RPN head) that achieves the top recall rate for a fixed number of proposals should be the most sensitive to this scale range.*

The resolution of output feature map ("Proposal Output") is another important issue. From the four figures (Figure 3.5, 3.6, 3.7, 3.8 and 3.9), we notice that the RPN head attached to different convolutional stages generates output feature maps of different strides and resolutions[8]. The stride of an output feature map is the same as the stride of the last conv-layer to which an RPN head is attached. In other words, the stride of the last conv-layer decides the stride and resolution of the output feature map, and thus decides the number of proposals. For example, given an input image of $H \times W \times 3$ and k anchors, RPN_conv4 in Figure 3.7 generates $\frac{H}{8} \times \frac{W}{8} \times k$ region proposals, which are four times as many as the $\frac{H}{16} \times \frac{W}{16} \times k$ region proposals generated by RPN_conv5 in Figure 3-8. In this situation, it is *unfair* to directly compare the recall rate of RPN_conv4 and RPN_conv5, because RPN_conv4 has more choices than RPN_conv5 for selecting good proposals. We therefore employs a deconvolutional layer between the last conv-layer and the L2 normalization layer. This permits us to up-sample the feature map (of last conv-layer). In addition, we employ a max-pooling layer for down-sampling the feature map (of the last conv-layer) so that RPNs with different conv-layers can generate output feature maps of the same stride and

convolutional stage 6, where the Conv6_2 layer has a receptive field of 276, larger than 196 of Conv5_3 and so it may better detect large facial patterns.

[8]The stride of a feature map is inversely proportional to the resolution of this feature map. For example, given an input image of the spatial dimension of H x W, if the output feature map of a CNN has a stride of N, then the feature map resolution is $(\frac{H}{N}) \times (\frac{W}{N})$.

thus the same number of object proposals for a fair comparison. Consider RPN_conv4 as an example. We can up-sample its feature map by employing a deconvolutional layer (see Figure 3.10a) so that the RPN_conv4 can be fairly compared with RPN_conv3. Also, it can be down-sampled by a max-pooling layer (see Figure 3.10b) in order to compare it with RPN_conv5.

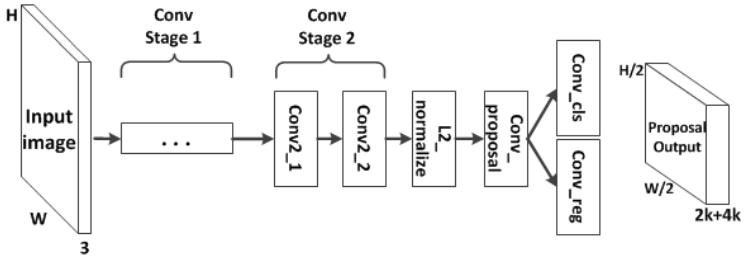


Fig. 3.5 The architecture of an RPN with a Conv2_2 (RPN_conv2). The conv-layers and max-pooling layers prior to the convolutional stage 2 have been omitted for the sake of brevity.

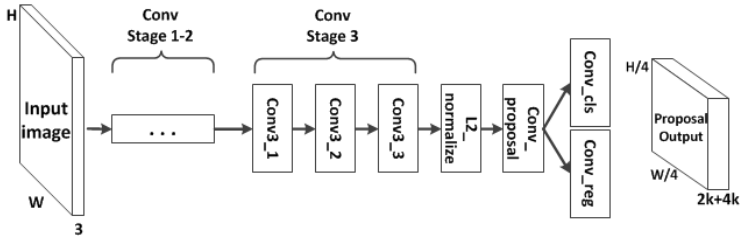


Fig. 3.6 The architecture of an RPN with a Conv3_3 (RPN_conv3) The conv-layers and max-pooling layers prior to the convolutional stage 3 have been omitted for the sake of brevity.

Consequently, we create 15 versions of RPN, as shown in Table 3.3. An RPN name in the format of *conv(M)_s(N)* indicates that the PRN head is attached to the last conv-layer of the M^{th} convolutional stage, and it generates an output feature map with a stride of N. Note that we only use three stride output sizes, 4, 8 and 16, which corresponds to the stride size of Conv3_3, Conv4_3 and Conv5_3, respectively. Although Conv2_2 has a stride of 2, we do not use this stride because of its large memory footprint. Similarly, we do not use the stride 32 inherent in Conv6_2 because small faces will probably be neglected using such a large stride. Instead, we employ deconvolutional layers to enlarge the stride of Conv6_2,

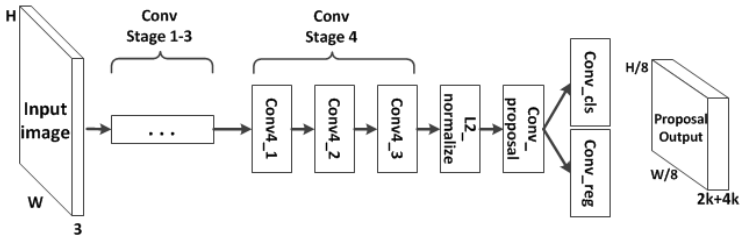


Fig. 3.7 The architecture of an RPN with a Conv4_3 (RPN_conv4). The conv-layers and max-pooling layers prior to the convolutional stage 4 have been omitted for the sake of brevity.

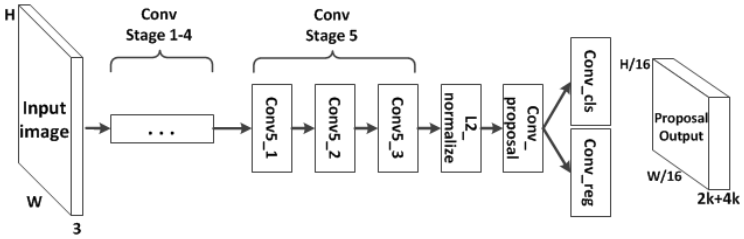


Fig. 3.8 The architecture of an RPN with a Conv5_3 (RPN_conv5). The conv-layers and max-pooling layers prior to the convolutional stage 5 have been omitted for the sake of brevity.

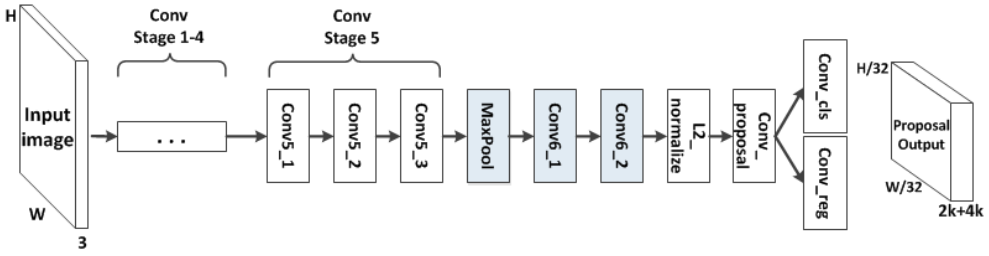
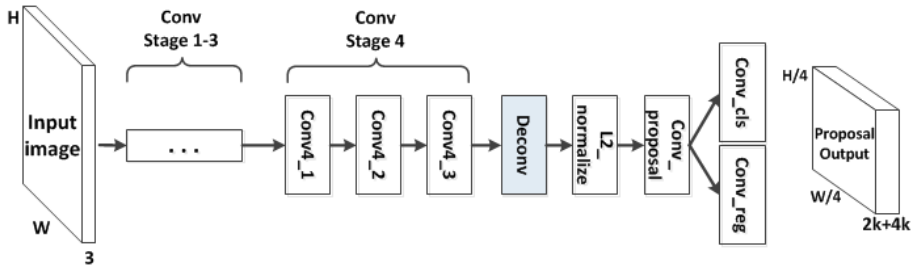
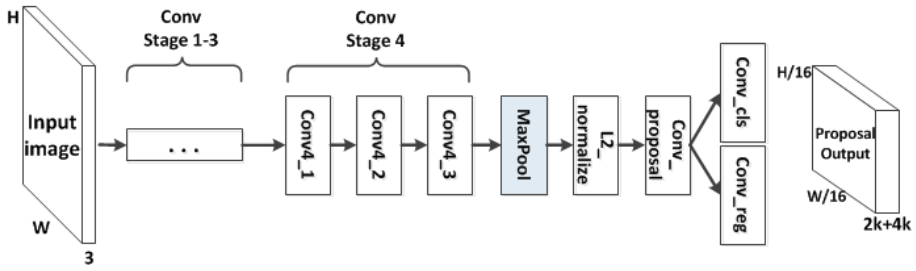


Fig. 3.9 The architecture of an RPN with Conv6_2 (RPN_conv6). The conv-layers and max-pooling layers prior to the convolutional stage 5 have been omitted for the sake of brevity.



(a) Feature map up-sampling



(b) Feature map down-sampling

Fig. 3.10 Feature map upsampling using a deconvolutional layer ("Deconv" in figure) and down-sampling using a max-pooling layer ("MaxPool" in figure).

and thus create conv6_s4, conv6_s8 and conv6_s16, respectively, for a fair comparison with other convolutional stages.

Data Preparation

We use the WIDER FACE [4] dataset for training and testing, as has been explained earlier in Section 2.3 of Chapter 2. Since there are nine networks to be trained and tested in this group of controlled experiments, in order to save training time, we selected 15 out of 61 events in the WIDER FACE training set for training. The selection process was as follows. As introduced in Chapter 2, images in the WIDER FACE dataset are organized based on 60 event categories. These categories are evenly divided into three levels of difficulty, "Easy", "Medium" and "Hard" according to the detection results of EdgeBox [5]. We first randomly selected 5 out of the 20 events at each difficulty level. Specifically, "Press Conference", "Swimming", "Family Group", "Couple" and "Tennis" were selected for the "Easy" level, "Greeting", "People Driving Car", "Group", "Interview" and "Rescue" were selected for the "Medium" level, and "Parade", "People Marching", "Concerts", "Award Ceremony" and "Car Racing" were selected for the "Hard" level. The 15 events that were chosen contain a total of 3,963 images, which make up about 31% of the whole training

Table 3.3 Different RPN's used in the controlled experiments

Conv-layer Used	RPN name	Output feature map stride	Deconvolution?	Max-pooling?
Conv2_2	conv2_s4	4	×	✓
	conv2_s8	8	×	✓
	conv2_s16	16	×	✓
Conv3_3	conv3_s4	4	×	×
	conv3_s8	8	×	✓
	conv3_s16	16	×	✓
Conv4_3	conv4_s4	4	✓	×
	conv4_s8	8	×	×
	conv4_s16	16	×	✓
Conv5_3	conv5_s4	4	✓	×
	conv5_s8	8	✓	×
	conv5_s16	16	×	×
Conv6_2	conv6_s4	4	✓	×
	conv6_s8	8	✓	×
	conv6_s16	16	✓	×

set.

Given a selected image I in the WIDER FACE dataset, we down-sample it to a half to obtain image $I_{\times 0.5}$ and up-sample it twice to obtain $I_{\times 2}$. Both down-sampling and up-sampling were done using bicubic interpolation. Then we randomly cropped a 512×512 image patch[9] from I, $I_{\times 0.5}$ and $I_{\times 2}$, respectively. Therefore, three 512×512 images were generated to replace the original image I and employed for training. This choice has three advantages. First, the original images in WIDER FACE have a relatively large and unfixed size, which varies around 900×1024. Since the image sizes are not fixed, only one image could have been trained each time (mini-batch size=1). However, after cropping to a fixed size (512×512), multiple images could be trained simultaneously (mini-batch size¿1), which improves the training efficiency. Second, as shown in Figure 3.1a and fig3-1-b, small faces

[9]If the original image had one or both sides containing less than 512 pixels, we padded the cropped image patch with zeros.

(between 10-50 pixels in height) are dominating the dataset. Up-sampling images can help enlarge the size of these faces, thus increasing the number of medium (between 50-300 pixels in height) and large (over 300 pixels in height) ones. Third, a small number of extra large faces are over 500 pixels in height, so we down-sample these images to reduce the size of these faces and maintain 512×512 image patches throughout the experiments. Note that some small faces could be made even smaller by down-sampling. However, as discussed in the next paragraph, extremely small faces (less than 5 pixels in height) were ignored during the training and testing process.

We use all 3,226 images of 60 event categories in the WIDER FACE validation for testing. These images were resized and cropped following the same procedure as stated above, thus creating a total of 9678 test images of the size of 512×512.

Training and testing Settings

a) Training settings

The computer code of the above 15 versions of RPN was built using Caffe [56]. The backbone architecture, VGG16, was pretrained on the ImageNet dataset [53]. The weights of all newly added convolutional layers were randomly initialized from a zero-mean Gaussian distribution with a standard deviation of 0.01. We use the following $k = 7$ anchor[10] scales for all RPN's: 8, 16, 32, 64, 128, 256 and 480, respectively. The aspect ratio was set to 1 for all anchors. An anchor was assigned as a positive sample if it had an intersection-over-union (IOU) ratio greater than 0.5 with any ground truth box, and as a negative sample if it had an IOU ratio less than 0.3 with any ground truth box. Each mini-batch contains 6 images of the size of 512×512. Each image had 40 sampled anchors. The ratio of positive and negative samples was set to 1:3 for all detection branches. All RPN's were trained by back-propagation and stochastic gradient descent (SGD) [57], using a learning rate of 0.0005 for 16k mini-batches, and 0.00005 for another 4k mini-batches. A momentum of 0.9 and a weight decay of 0.0005 were used.

b) Testing settings

After obtaining face proposals from a test image using the trained model, we first eliminated all proposals with a confidence score less than 0.1. Then non-maximum suppression (NMS) with a threshold of 0.7 was adopted to filter the remaining proposals based on their confidence scores. Finally, the rest of the proposals were ranked by their scores and a

[10]See section 3.1 of an explanation of the anchors.

certain number of the top-ranked ones were selected to calculate the face detection recall rate. In the past, methods based on region proposals retained a fixed number of proposals (e.g., 300 proposals used in [20]) for each image in order to compute the recall rate. In contrast, we have adopted a stricter adaptive selection strategy. Given a test image with N ground truth faces, we only selected the $2N$ top-ranked face proposals and computed their recall rates. Since images in the test set have an average of 7.5 faces, we used an average of 15 proposals per image instead of the commonly used 300. This strict setting put more responsibility on the RPN to provide high-quality face proposals. We computed the recall rate of these proposals according to 31 scale ranges of ground truth face height: [5, 15), [15, 25), , [295, 305), [305, -). The final scale range included all faces with a height equal or larger than 305 pixels. For a face in a certain scale range, if it has an IOU larger than 0.5 with any proposal, it is counted as a true positive; otherwise, it is counted as a false negative.

Results

The recall rates of the fifteen RPNs are shown in Figure 3.11. We see that conv3_s4 achieves the highest recall rate for the range of 5-15, and conv4_s8 has the best performance for the range of 15-35. For the scale larger than 35 pixels, conv5_s16 and conv6_s16 perform the best[11]. This result verifies our postulate that *conv-layers with large receptive fields (e.g. Conv5_3 and Conv6_2) are more sensitive to large object patterns, while those with small receptive fields (e.g. Conv3_3) are more sensitive to small scale patterns.* Another interesting observation from Figure 3.11 is that *the output stride (i.e., s4, s8 and s16) influences the recall rate.* For an RPN at any conv-layer (e.g. conv2, conv3, conv4 or conv5), a small stride (e.g., 4) can help detect small-scale faces with a higher recall than a large stride (e.g., 16), while a large stride can facilitate a better detection of large-scale faces[12].

[11]Please visit the website `https://plot.ly/~yumkong/38/`, where a detailed view of Figure 3.11 is provided for a fine-grained comparison of different curves within each facial height range. This website is long-term effective.

[12]Note that there is an exception to this rule at the conv6 stage: conv6_s4 performs worse than conv6_s16 in the detection of small faces within the range of 5-15 pixels in height. This is possibly because conv6_s4 undergoes an 8× interpolation in the original feature map (by a deconvolutional layer), and this large distortion severely affects its performance in detecting small facial patterns.

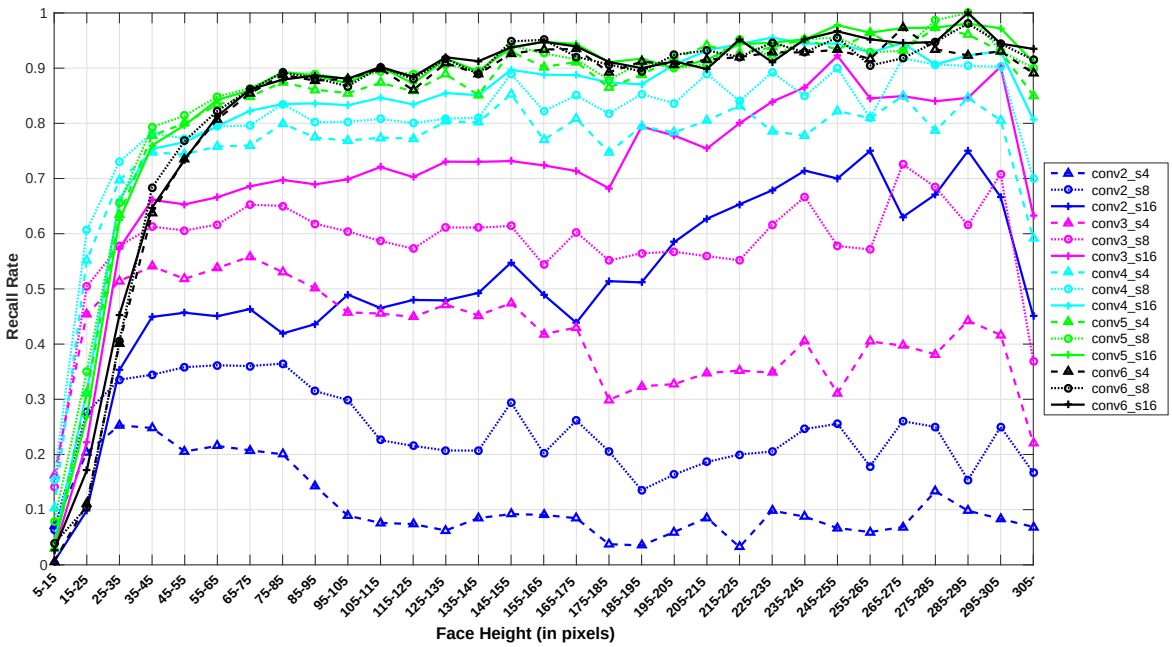


Fig. 3.11 Recall rates of different Region Proposal Networks.

3.2.2 Scale sensitivity of combined conv-layers

From the previous sub-section, we know conv-layers differ in scale sensitivity and there is no single conv-layer that is proficient in detecting faces at all scales. A possible workaround is to combine different conv-layers in order to extend the range of scale sensitivity. In this sub-section, we investigate different ways of combining multiple conv-layers to see if they can extend scale sensitivity as well as improve the recall rate in each scale range.

Network architecture

a) A new convolutional stage

Before investigating the combination of conv-layers, we first create a new conv-layer stage, the convolutional stage (conv-stage) 6. This stage consists of two conv-layers, Conv6_1 and Conv6_2. Conv6_1 is a $3 \times 3 \times 512 \times 1024$ convolutional layer, and Conv6_2 is a $1 \times 1 \times 1024 \times 256$ convolutional layer for feature map dimension reduction. This stage is created to presumably improve the recall rate of large faces[13]. Like other convo-

[13]We noticed a small drop of recall rate between the last two scale ranges in Figure 3.11, indicating that Conv5_3 may not be the best choice for detection large-size faces. So we created the convolutional stage

lutional stages, conv-stage 6 can add a deconvolutional or max-pooling layer and a RPN head to form a new version of RPN of different strides. Here we have created two versions: conv6_s16 and conv6_s32 (see Figure 3.9). We do not use conv6_s4 or conv6_s8, because conv-stage 6 is created mainly for detecting large facial patterns, while small strides can compromise this goal, as discussed in the previous section.

b) Combining multiple conv-layers

We investigate two methods of combining multiple conv-layers. The first is to apply a concatenation layer to concatenate multiple conv-layers along the channel dimension. This is especially useful when the conv-layers have different channel dimensions. For example, when combining conv2_2, conv3_3 and conv4_3_reduce[14], which have feature maps of dimensions $\frac{H}{4} \times \frac{W}{4} \times 128$, $\frac{H}{4} \times \frac{W}{4} \times 256$ and $\frac{H}{4} \times \frac{W}{4} \times 256$ respectively, a concatenation layer can concatenate them into a single feature map of $\frac{H}{4} \times \frac{W}{4} \times 640$ (see Figure 3.12). The second is to use an element-wise addition layer to sum feature maps of multiple conv-layers, on the premise that all these feature maps are of the same channel dimension. For example, in Figure 3.13, we add the feature maps of conv3_3, conv4_3 and conv5_3 since they have or are reduced to the same dimension of $\frac{H}{8} \times \frac{W}{8} \times 256$. The resulting feature map is still of the same dimension of $\frac{H}{8} \times \frac{W}{8} \times 256$. We experimentally found that a concatenation layer and an element-wise addition layer have almost the same performance. Since the element-wise addition layer takes up less memory, we mostly use it in our experiments. But when combining Conv2_2 with other conv-layers, we use the concatenation layer since Conv2_2 has a different spatial dimension.

All the new RPNs with combined layers are listed in Table 3.4. Note that conv2 is used only with a stride of 4, and conv6 only with a stride of 16 and 32. This corresponds to the conclusions drawn in sub-section 3.2.1. That is, that conv-layers with small receptive field should use a small stride to better detect small facial patterns, while those with a large receptive field should be equipped with a large stride to increase the recall rate of large facial patterns. By contrast, the conv-layers with medium-size receptive fields, such as conv3, conv4 and conv5, are flexible and already equipped with different possible strides.

6, where the Conv6_2 layer has a receptive field of 276, larger than 196 of Conv5_3 and so it may better detect large facial patterns.

[14]Conv4_3_reduce is a $1 \times 1 \times 512 \times 256$ convolutional layer added after conv4_3 in order to reduce the channel dimension of the conv4_3 feature map from 512 to 256. It can reduce memory footprint of the RPN network without compromising performance. A similar conv-layer, conv5_3_reduce has been used after conv5_3.

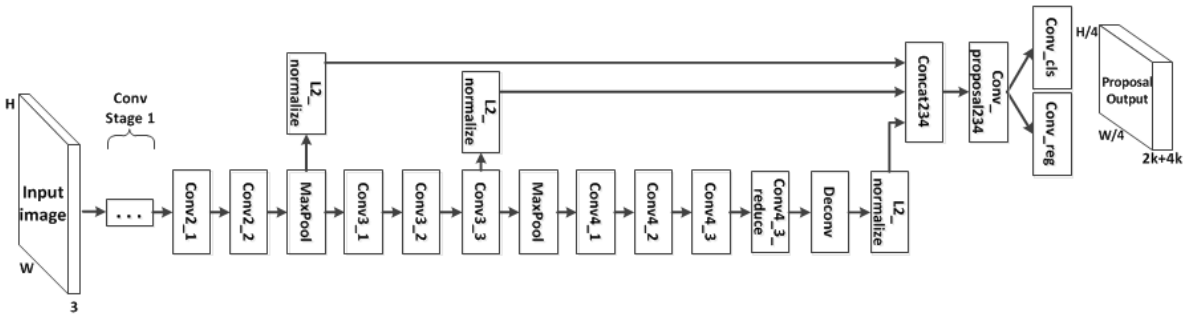


Fig. 3.12 The architecture of conv234_s4. A concatenation layer is used to combine different conv-layers (Conv2_2, Conv3_3 and Conv4_3).

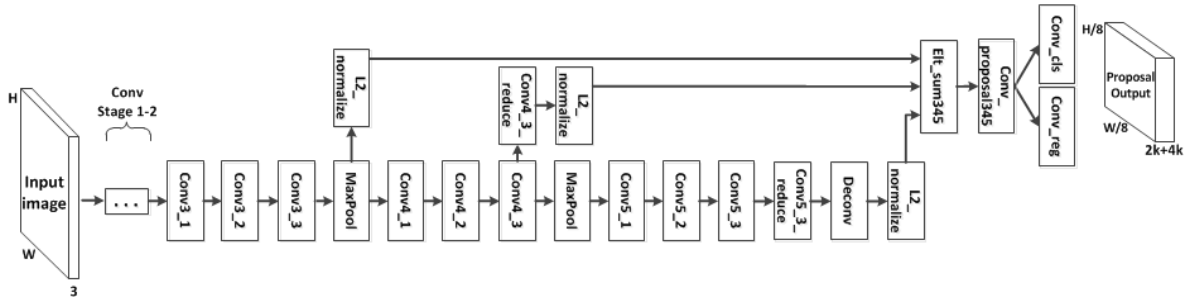


Fig. 3.13 The architecture of conv345_s8. An element-wise addition layer is used to combine different conv-layers (Conv3_3, Conv4_3 and Conv5_3).

Table 3.4 RPNs with combined conv-layers

Stride	RPN name	Conv-layer Used	Combination Method	
			Concatenation	Element-wise addition
4	conv23_s4	conv2_2, conv3_3	✓	
	conv34_s4	conv3_3, conv4_3		✓
	conv45_s4	conv4_3, conv5_3		✓
	conv234_s4	conv2_2, conv3_3, conv4_3	✓	
	conv345_s4	conv3_3, conv4_3, conv5_3		✓
8	conv34_s8	conv3_3, conv4_3		✓
	conv45_s8	conv4_3, conv5_3		✓
	conv345_s8	conv3_3, conv4_3, conv5_3		✓
16	conv34_s16	conv3_3, conv4_3		✓
	conv45_s16	conv4_3, conv5_3		✓
	conv56_s16	conv5_3, conv6_2		✓
	conv345_s16	conv3_3, conv4_3, conv5_3		✓
	conv456_s16	conv4_3, conv5_3, conv6_2		✓
32	conv56_s32	conv5_3, conv6_2		✓

Results

Putting the RPNs with both single layers and the combination of multiple conv-layers together, there are 28 different versions of RPNs. To enable a fair and detailed comparison, we compute the recall rate of a even more fine-grained partitions of scale ranges: [5, 8), [8, 11), ..., [497, 500), [500, -). This leads to a 28×167 table, which is not included in this thesis due to its large size. However, we found an interesting phenomenon that can help simplify the original table: in some consecutive scale ranges, a certain RPN always achieves the highest recall rate. For example, conv23_s4 achieves the top recall rate in the ranges of [5, 8) and [8, 11), and conv345_s8 performs the best in the ranges from [11, 14) up to [44, 47). Therefore, we can combine [5, 8) and [8, 11) into [5, 11), and [11, 14) up to [44, 47) into [11, 47). As a result, we partition the whole spectrum of scale into eight ranges: [5, 11), [11, 47), [47, 65), [65, 86), [86, 128), [128, 239), [239, 371), [371, -). The

recall rates of all RPN's on these scale ranges are shown in Table 3.5 (next page).

We can see that in each scale range, the best recall rate is always achieved by a combination of conv-layers. Also, the newly added conv6_2 contributes to conv56_s16 and conv456_s16, which achieve the highest recall rate for large-scale faces. However, it is noticeable that even the combination of multiple conv-layers can only achieve top performance *within a certain scale range*, not *the whole scale spectrum*.

3.3 Multi-Path Face Proposal Network

We have shown in the previous section that RPN with either a single conv-layer or a combination of several conv-layers cannot achieve top performance for all scale ranges. This necessitates a multi-path region proposal network, where each path only deals only with a scale range in which it is most proficient. Together, the parallel paths achieve the top performance for each scale range. According to Table 3-5, we can clearly divide the whole scale spectrum into three big ranges: [5,11) where conv23_s4 achieves the best recall, [11, 128) where conv345_s8 performs the best, and [128,-) where conv56_s16 performs the best. According to this partition, we can create a forked three-path face proposal network to simultaneously examine the three ranges, as shown in Figure 3.14.

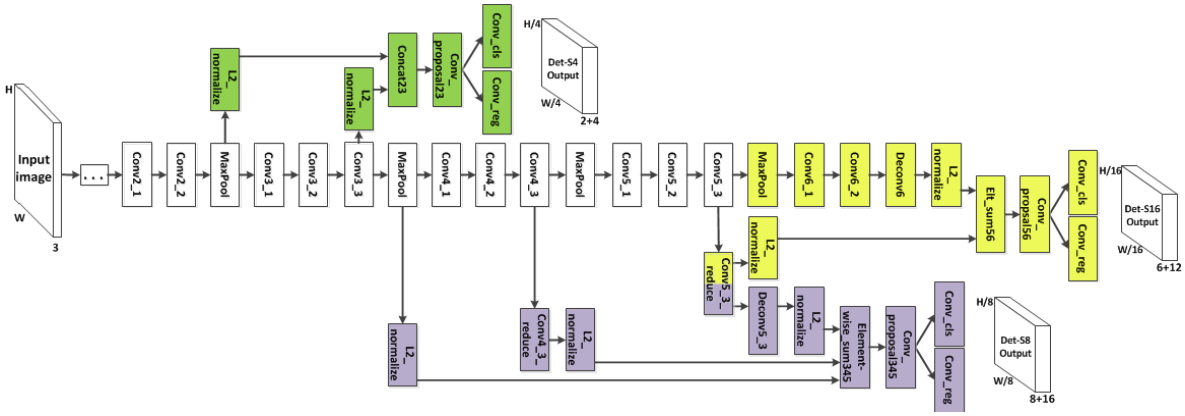


Fig. 3.14 The architecture of a Multi-Path Face Proposal Network proposed in this thesis. The conv-layers and max-pooling layers prior to the convolutional stage 2 have been omitted for the sake of brevity. The three parallel paths are colored in green, purple and yellow, respectively.

Table 3.6 below shows the anchor scales (in pixels) allocated to each path. These are decided automatically by the appropriate scale range.

Data Preparation and Settings

We trained the proposed MP-FPN with exactly the same dataset used in section 3.2. The training and testing settings also follow section 3.2, except for the following one difference. In section 3.2, all the RPN's in the experiments have only one output path. Thus the 40 sampled anchors were assigned to this single output path. However, there are three parallel paths in MP-FPN, so that the sampled anchors should accordingly be split into three paths. We assigned 16 sampled anchors for Det-s4 path, 24 anchors for Det-s8 path and 8 anchors for Det-s16 path. This allocational ratio is based on the number of ground truth faces within the scale range of each detection path.

Results

The detection recall rate of MP-FPN is shown in Table 3.7. It is compared with three strong baselines: conv23_s4, conv345_s8 and conv56_s16, which are the building blocks of MP-FPN. We see that MP-FPN outperforms the three strong baselines in all scale ranges except for the range of 86-128, where it is slightly inferior to conv56_s16. We attribute the overall superior performance of MP-FPN to the multi-path partition of the network, where each detection path is able to concentrate on the most appropriate facial range of scales.

3.4 Conclusion

This chapter investigates the scale sensitivity of convolutional layers by a series of controlled experiments. These experiments are based on the Region Proposal Network (RPN) of Faster R-CNN framework [20]. We first introduced Faster R-CNN and the traditional RPN, and then converted the latter into twelve versions of new RPNs. By training and testing these RPNs with exactly the same data and parameters, we obtained the first interesting result: conv-layers of different stages are sensitive to different scale ranges. In particular, the result reflects the common sense that conv-layers with large receptive fields are more sensitive to large object patterns, while those with small receptive fields are more sensitive to patterns of small scales. This result well answers ***Question 1*** proposed in Chapter 1: What is the reason behind the phenomenon that “tiny faces” cannot be accurately

detected by ConvNets? Our answer is that *a common convolutional neural network simply employs the feature map of its last conv-layer to predict faces. This last conv-layer, with a large receptive field, is not sensitive to small facial patterns, thus leading to a low detection accuracy of these tiny patterns.*

According to this experimental result, we raise another interesting question: can we combine different stages of conv-layers to create a single output feature map that *simultaneously* provides an optimal recall rate for each input facial scale range? The second group of controlled experiments gave us a partial NO answer. These experiments showed that a combination of conv-layers from nearby convolutional stages can actually increase the sensitivity, and thus the recall rate of a specific scale range. However, NO combination can achieve an overall optimal performance across the whole scale range. Worse still, combining too many stages of conv-layers may reduce the recall rate. For example, in Table 3-5, conv234_s4 performs worse than conv23_s4 when detecting faces of the scale range of 8-11.

The above results inspired the creation of a Multi-Path Face Proposal Network (MP-FPN) presented in this thesis. MP-FPN uses three parallel paths for proposing faces of different scales. Each detection path is a combination of two or three conv-layers from nearby convolutional stages that already achieve the best performance in a certain scale range. We experimentally found that MP-RPN achieves an overall optimal performance within the whole scale range. We attribute this result to two merits of MP-RPN. First, the partition of multiple paths values the scale sensitivity of individual conv-layers. In each detection path, the conv-layers are allowed to deal solely with the scale ranges that they are proficient in. Second, the appropriate combination of conv-layers of nearby scales enhances their sensitivity within a certain scale range.

We observe that MP-FPN is the first CNN framework that achieves a parallel detection of faces in multiple scales. It eliminates the onerous common practice of extracting features from an image pyramid. Instead, MP-FPN simply requires a single-scale image as input, and the detection of faces at multiple scales can be efficiently achieved by the natural hierarchical structure of convolutional layers.

So far, we have already provided an answer to ***Question 2*** proposed in Chapter 1: Is there any way that we can adapt the deep learning framework so as to detect tiny facial patterns with high accuracy? Our answer is: *Yes, MP-FPN is a viable choice.*

Table 3.5 Recall rate of all RPNs based on scale range

Output feature map stride	RPN name	Scale range							
		5-11	11-47	47-65	65-86	86-128	128-239	239-371	371-
4	conv2_s4	0.027	0.173	0.117	0.096	0.07	0.086	0.088	0.017
	conv3_s4	0.101	0.449	0.567	0.586	0.509	0.445	0.412	0.061
	conv4_s4	0.036	0.539	0.773	0.809	0.81	0.834	0.81	0.378
	conv5_s4	0.009	0.388	0.821	0.86	0.867	0.902	0.939	0.753
	conv23_s4	**0.105**	0.444	0.573	0.579	0.511	0.427	0.314	0.064
	conv34_s4	0.091	0.571	0.769	0.809	0.81	0.827	0.805	0.382
	conv45_s4	0.035	0.563	0.82	0.868	0.865	0.903	0.941	0.743
	conv234_s4	0.098	0.582	0.774	0.81	0.812	0.823	0.769	0.412
	conv345_s4	0.092	0.61	0.82	0.865	0.866	0.904	0.93	0.764
8	conv2_s8	0.025	0.269	0.382	0.375	0.292	0.224	0.266	0.047
	conv3_s8	0.065	0.489	0.64	0.68	0.626	0.614	0.634	0.155
	conv4_s8	0.071	0.588	0.793	0.828	0.825	0.861	0.879	0.527
	conv5_s8	0.057	0.421	0.833	0.877	0.888	0.914	0.957	0.818
	conv34_s8	0.085	0.611	0.787	0.83	0.837	0.868	0.875	0.493
	conv45_s8	0.087	0.619	0.831	0.874	0.884	0.92	0.944	0.828
	conv345_s8	0.099	**0.635**	**0.838**	0.882	**0.892**	0.914	0.959	0.821
16	conv2_s16	0	0.182	0.456	0.447	0.473	0.542	0.686	0.213
	conv3_s16	0.009	0.318	0.661	0.692	0.708	0.75	0.849	0.405
	conv4_s16	0.012	0.393	0.782	0.831	0.837	0.889	0.923	0.669
	conv5_s16	0.015	0.365	0.82	0.873	0.882	0.922	0.96	0.845
	conv6_s16	0.012	0.27	0.779	0.869	0.89	0.925	0.959	0.899
	conv34_s16	0.014	0.434	0.796	0.844	0.841	0.889	0.933	0.709
	conv45_s16	0.015	0.452	0.831	0.882	0.888	0.921	0.958	0.858
	conv56_s16	0.015	0.371	0.82	**0.884**	0.885	**0.926**	**0.962**	0.899
	conv345_s16	0.017	0.45	0.832	0.877	0.891	0.925	0.958	0.858
	conv456_s16	0.017	0.416	0.825	0.883	0.885	0.925	0.956	**0.902**
32	conv6_s32	0	0.103	0.65	0.822	0.86	0.911	0.951	0.875
	conv56_s32	0	0.114	0.651	0.823	0.659	0.919	0.957	0.882

Table 3.6 Anchor allocation for MP-FPN

Detection Path	**Det-s4**	**Det-s8**	**Det-s16**
Anchor scale	8×8	16×16, 32×32, 64×64, 128×128	128×128, 256×256, 480×480

Table 3.7 Rate recall of MP-FPN and other baselines

RPN name	**Scale range**							
	5-11	11-47	47-65	65-86	86-128	128-239	239-371	371-
conv23_s4	0.105	0.444	0.573	0.579	0.511	0.427	0.314	0.064
conv345_s8	0.099	0.635	0.838	0.882	**0.892**	0.914	0.959	0.821
conv56_s16	0.015	0.371	0.82	0.884	0.885	0.926	0.962	0.899
MP-FPN	**0.284**	**0.655**	**0.847**	**0.891**	0.888	**0.93**	**0.966**	**0.959**

Chapter 4

Overall Architecture of Multi-Path Face Detection Network

This chapter deals with the overall face detection architecture.

In the previous chapter, we proposed a Multi-Path Face Proposal Network (MP-FPN) that generates face proposals of different sizes via three parallel paths. Each face proposal has a corresponding confidence score, indicating its probability of being a face. As has been seen in the previous chapter, by sorting face proposals according to their confidence score and selecting the high-scoring proposals as predicted faces, MP-FPN can already serve as a strong stand-alone face detector.

Nevertheless, when visualizing the detection results of MP-FPN, we found that some high-scoring face proposals are actually false positives. For example, human hands, ears, and textured walls are mistakenly detected[1] as faces as shown in Figure 4.1a. In contrast, some low-scoring face proposals are true faces (false negatives). For instance, some faces that are blurred, partially-occluded, of low-resolution or with large yaw/roll angles are overlooked as shown in Figure 4.1b

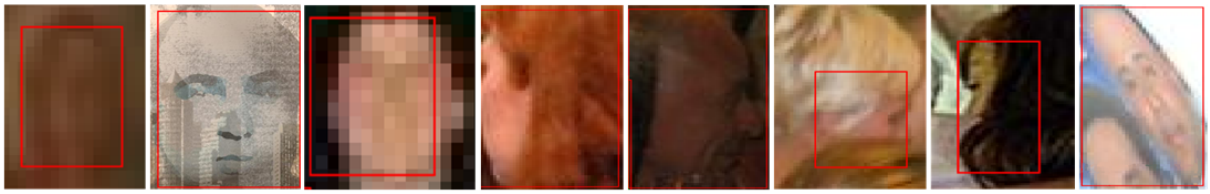
In view of the above situation, we propose making two separate decisions rather than one in order to detect a face:

1. Is an image patch of small, medium or large-size a potential face?
2. Is the potential face actually a true face?

The MP-FPN proposed in the previous chapter has already made the first decision.

[1]We use the value 0.7 to threshold face/non-faces. A face proposal with a confidence score larger than 0.7 is counted as a face. Otherwise, it is counted as a non-face.

(a) False positive examples generated by MP-FPN (green boxes are predicted face bounding boxes)



(b) False negative examples missed by MP-FPN (red boxes are ground-truth face bounding boxes)

Fig. 4.1 Examples of false positives and false negatives of MP-FPN on WIDER FACE validation set.

This chapter first makes the second decision by removing difficult false positives while still including difficult false negatives. This is achieved by a newly proposed Multi-Path Face Verification Network (MP-FVN). Next, a new overall architecture of the proposed Multi-Path Face Detection Network is introduced, with MP-FPN and MP-FVN as the building blocks. This is the primary contribution of this thesis. Finally, we present a chapter conclusion.

4.1 Multi-Path Face Verification Network

We attribute the occurrence of difficult false positives and false negatives to two major reasons. First, in MP-FPN, each face proposal is represented by either a 256-dimensional (Det-s8 and Det-s16 path) or a 384-dimensional (Det-s4 path) vector. This turns out not to be sufficiently discriminative of the difficult face/non-face patterns as shown in Figure 4-1. Second, we can observe from Figure 4-1 that most misclassifications happen when the image patches are of low resolution. Clearly, these low-resolution patches are insufficient for extracting representative features. A natural way to solve this problem would be to introduce contextual information. Torralba and Sinhas human vision experiments [58] have shown that the inclusion of contextual information increases a humans ability to detect

faces. Later, contextual information was also used in face detection algorithms [43, 2] to improve detection accuracy. The objective of this section is to design a Face Verification Network (FVN) to further improve the detection recall rates for faces at various scales. To this end, the FVN should be able to easily combine features from both a facial region and a corresponding contextual region. Moreover, the FVN should also leverage higher dimensional vectors to better represent the facial and contextual features than MP-FPN does.

We first chose to use Fast R-CNN [16] as a preliminary framework of the Face Verification Network (FVN) for the following two reasons. First, it leverages an ROI-pooling layer and two consecutive fully-connected (fc) layers to represent each candidate object region. Thus employing both of these fc layers makes it possible to represent the region by a 4096-dimensional vector. This is more representative of facial features than the 256-d or 384-d vectors used in MP-FPN. Second, the ROI-pooling layer can easily be extended to pool features from both an object region and a larger contextual region[2].

However, the preliminary result showed that Fast R-CNN can only improve the recall rate for large-size faces compared to MP-FPN. The recall rate of Fast R-CNN for small- and medium-size faces even decreases compared to MP-FPN[3]. We postulate that this result is largely due to the fact that Fast R-CNN only pools features from the feature map of the last conv-layer (conv5_3). Nevertheless, as has been shown in Chapter 3, a single conv-layer cannot generate feature maps that are sensitive to all facial scales. To verify this postulate, as well as design an optimal Face Verification Network that can improve the recall rate of all facial scales, we adapted the original Fast R-CNN to the following structure as shown in Figure 4.2.

We use exactly the same three paths (s4, s8 and s16) as in MP-FPN to represent small, medium and large facial patterns. The only difference is that an ROI-pooling layer is attached to the end of each detection branch to replace an RPN head in MP-FPN. These ROI-pooling layers pool features from different detection paths according to the size of the face proposals. When a face proposal has a height less than 12 pixels, its features are pooled from s4 path. When a face proposal has a height larger than 128 pixels, its features are pooled from s16 path. Lastly, face proposals with a height between 12 and 128 pixels are processed by the s8 path. This partition follows the optimal scale range of each

[2]The ROI-pooling layer in original Fast R-CNN framework only pools features from an object region.
[3]See Table 4.1 for detailed comparisons. "FVN-conv5" in Table 4.1 is actually Fast R-CNN.

detection path, as indicated in Chapter 3. Since the values of the features pooled from these different paths have different scales and norms, an L2-normalization layer is added after ROI-pooling layer to keep the value from each path within approximately the same scale. Then, because the features pooled from each detection path are of the same dimension, they can be concatenated along in a fourth channel (the so-called "num" channel in Caffe [56]), and then passed through two fc layers for face/non-face classification[4]. Note that the original Fast R-CNN does a simultaneous classification and regression because the prior RPN stage cannot provide a precise bounding box for each object category. In contrast, since MP-RPN stage has already provided precise face bounding boxes, we eliminate the regression task in this face verification stage. We will refer to the face verification stage as a Multi-Path Face Verification Network (MP-FVN) since face proposals are verified via three parallel paths according to their size.

In the following sub-sections, a series of controlled experiments are discussed to verify the effectiveness of the proposed MP-FVN. We also investigate the effect of adding different contextual information, different ROI-pooling sizes, as well as an online hard example mining (OHEM) layer.

4.1.1 Training and test settings

We use exactly the same data and settings for all of the controlled experiments in this chapter, as follows.

Data Preparation We directly use the same training and testing data as in Chapter 3. Specifically, the training data contain 11889 images of size 512×512. These images come from 15 event categories of the WIDER FACE training set. The testing data contain 9,678 images of size 512×512. These images come from all 61 event categories of the WIDER FACE validation set. See Section 3.2 for the details of the data preparation.

Training settings The proposed MP-FVN and other baseline networks were all built using Caffe [56]. The trained MP-FPN model in Chapter 3 was used to initialize MP-FVN and all other baselines. For example, since MP-FVN and MP-FPN share convolutional layers in conv-stage1-6, as well as "Conv4_3_reduce" and "Conv5_3_reduce", the parameters of these conv-layers in MP-FVN were initialized by the parameters of corresponding conv-

[4]This face/non-face classification is actually a re-scoring of the face proposals provided by MP-FPN.

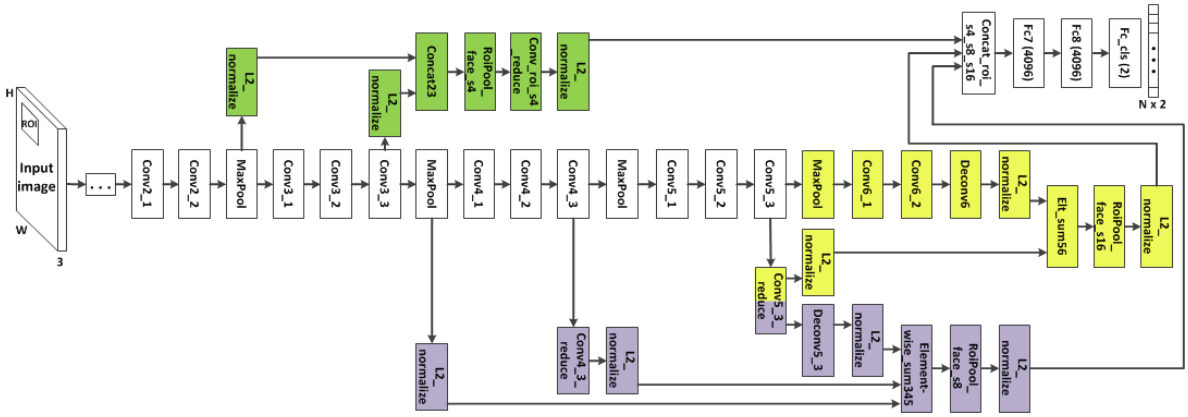


Fig. 4.2 The architecture of the proposed Multi-Path Face Verification Network (MP-FVN) without contextual information. Given an image and N face proposals as input data, MP-FVN outputs an $N \times 2$ vector, indicating the face/non-face score of each proposal. Note that a "Conv_roi_s4_reduce" convolutional layer is added after "RoiPool_face_s4" to reduce the pooled feature block in s4 path from 384-d to 256-d. S8 and s16 paths do not have such a conv-layer because they naturally generate 256-d feature blocks.

layers in MP-FPN. The weights of all newly added fully-connected layers were randomly initialized from a zero-mean Gaussian distribution with a standard deviation of 0.01. A face proposal was assigned as a positive sample if it had an intersection-over-union (IOU) ratio greater than 0.5 with any ground truth box, and as a negative sample if it had an IOU ratio less than 0.3 with any ground truth box. Each mini-batch contained 2 images of the size of 512×512. Each image had 48 sampled face proposals in each baseline network. The ratio of positive and negative samples was set to 1:3. For MP-FVN, we assigned 16 sampled face proposals for the s4 path, 32 for the s8 path and 8 for the s16 path. This allocational ratio was based on the number of ground truth faces within the scale range of each detection path. The ratio of positive and negative samples was set to 1:3 for all detection paths. All networks were trained by back-propagation and stochastic gradient descent (SGD) [57], using a learning rate of 0.0005 for 12k mini-batches, and 0.00005 for another 12k mini-batches. A momentum of 0.9 and a weight decay of 0.0005 were used.

Testing settings Given a test image, we first applied MP-FPN discussed in Chapter 3 to obtain face proposals of this image. Then all proposals with a confidence score less than 0.1 were eliminated. All remaining face proposals and the test image were fed to

a MP-FVN to obtain final classification scores. Next, non-maximum suppression (NMS) with a threshold of 0.5 was adopted to filter the face proposals based on their classification scores. Finally, the rest of the proposals were ranked by their classification scores. The same adaptive selection strategy that was introduced in Chapter 3 was adopted to obtain a specific number of the top-ranked face proposals in order to calculate the face detection recall rate. Specifically, given a test image with N ground truth faces, we only selected the 2N top-ranked face proposals and computed their recall rates. To make a fine-grained comparison of the recall rate in different facial scale ranges, we computed the recall rate of these proposals according to the same eight scale ranges of ground truth face heights used in Chapter 3: [5, 11), [11, 47), [47, 65), [65, 86), [86, 128), [128, 239), [239, 371), [371, -). The final scale range included all faces with a height equal to or larger than 371 pixels. For a ground-truth face within a certain scale range, if it has an IOU larger than 0.5 with any face proposal, it was counted as a true positive; otherwise, it was counted as a false negative.

4.1.2 Comparison with baselines

We compared the proposed MP-FVN with five baseline Face Verification Networks (FVN): FVN-conv2, FVN-conv3, FVN-conv4, FVN-conv5 and FVN-conv2345, as shown in Figure 4.3, 4.4, 4.5, 4.6 and 4.7, respectively. The network "FVN-conv(M)" means the ROI-pooling and fc layers are attached to the Mth convolutional stage, and all the subsequent conv-layers in VGG16 backbone are eliminated. Particularly, in FVN-conv2345, for each face proposal, features were pooled from conv-stage 2, 3, 4 and 5, respectively by separate ROI-pooling layers. These features were then concatenated as a single feature block which passed fc layers to obtain a classification score.

Table 4.1 shows the test results of MP-FVN and other baselines. We can see that the proposed MP-FVN outperforms other baselines for the overall recall rate. As for fine-grained comparisons, MP-FVN outperformed other baselines in all face scale ranges except the range [239, 371), where it lagged behind FVN-conv2345 by a slight margin. We also note that when using a single conv-layer for face verification, the overall recall rate is even less than MP-FPN, meaning that no single conv-layer contains enough information for improving the overall face detection recall rate. In contrast, when combining features from multiple conv-layers as in FVN-conv2345, the overall recall rate surpassed that of MP-FPN.

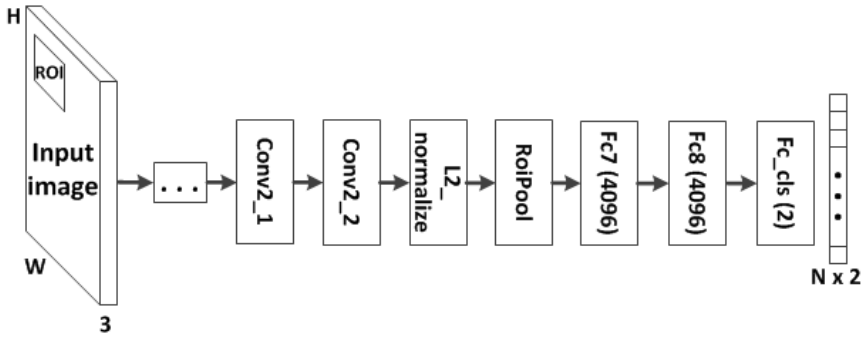


Fig. 4.3 The architecture of FVN-conv2.

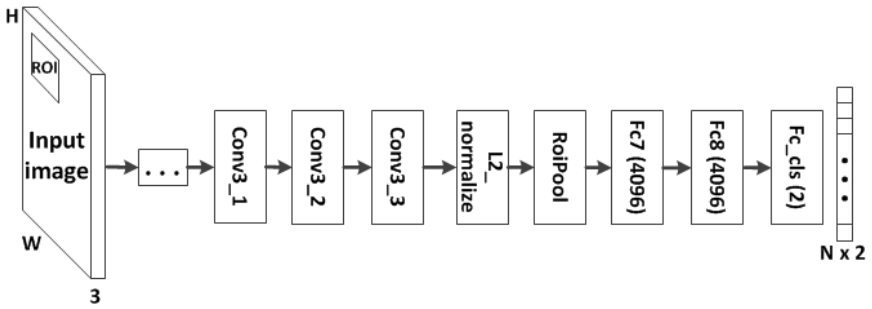


Fig. 4.4 The architecture of FVN-conv3.

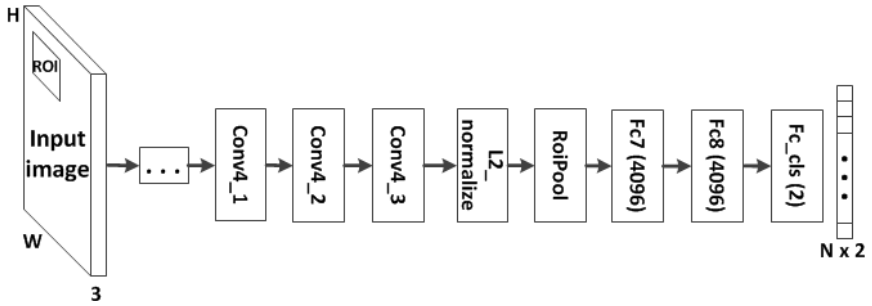


Fig. 4.5 The architecture of FVN-conv4.

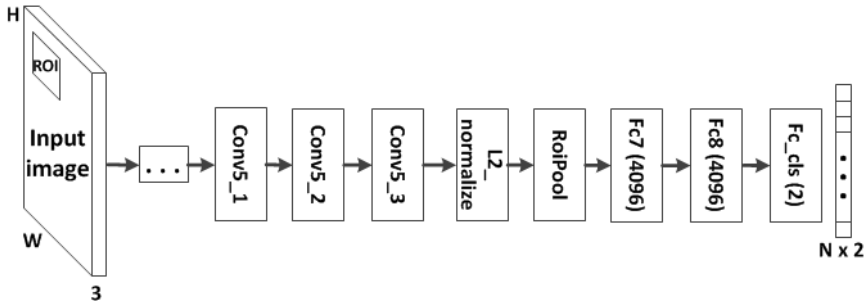


Fig. 4.6 The architecture of FVN-conv5.

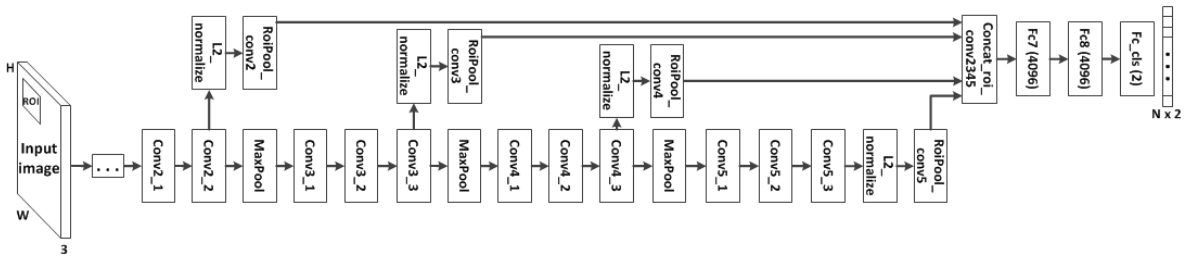


Fig. 4.7 The architecture of FVN-conv2345.

However, the recall rate can be further improved by employing the MP-FVN, where each path combines the conv-layers that are most sensitive to a certain scale range. We notice that MP-FVN improves the recall rates of MP-FPN in all scale ranges. This demonstrates that adding MP-FVN as a second stage of the proposed face detection process is necessary. As an overall comment, we note that the two lower scale ranges are the most difficult and do not perform very well.

4.1.3 Does context help?

The MP-FVN used in the previous subsection only pools features from face proposal regions. In this subsection, we also extract features from a larger contextual region to investigate the effect of adding contextual information to MP-FVN.

We investigate four types of contextual information, as shown in Figure 4-8. Suppose the original region is $[l, t, w, h]$, where l is the horizontal coordinate of its left edge, t the vertical coordinate of the top edge, and w, h the width and height of the region, respectively. The corresponding four types of contextual regions are defined as follows.

Table 4.1 Rate recall of MP-FVN and other baselines (without context information)

Network name	Scale range								
	5-11	11-47	47-65	65-86	86-128	128-239	239-371	371-	All
MP-FPN	0.284	0.655	0.847	0.891	0.888	0.93	0.966	0.959	0.603
FVN-conv2	0.245	0.649	0.841	0.862	0.908	0.933	0.955	0.946	0.589
FVN-conv2	0.26	0.645	0.849	0.871	0.91	0.942	0.958	0.949	0.592
FVN-conv2	0.247	0.642	0.856	0.872	0.913	0.947	0.964	0.956	0.588
FVN-conv2	0.25	0.601	0.842	0.867	0.914	0.942	0.965	0.953	0.565
FVN-conv2	0.267	0.681	0.864	0.891	0.921	0.949	**0.972**	0.966	0.616
MP-FVN	**0.293**	**0.691**	**0.879**	**0.912**	**0.928**	**0.955**	0.967	**0.97**	**0.631**

1. Context I: $[l - 0.5w, t, 2w, 2h]$, which is 2×2 bigger than the original region and approximately covers the hair below the forehead, the neck and part of the shoulder of a person (See Figure 4.8a).

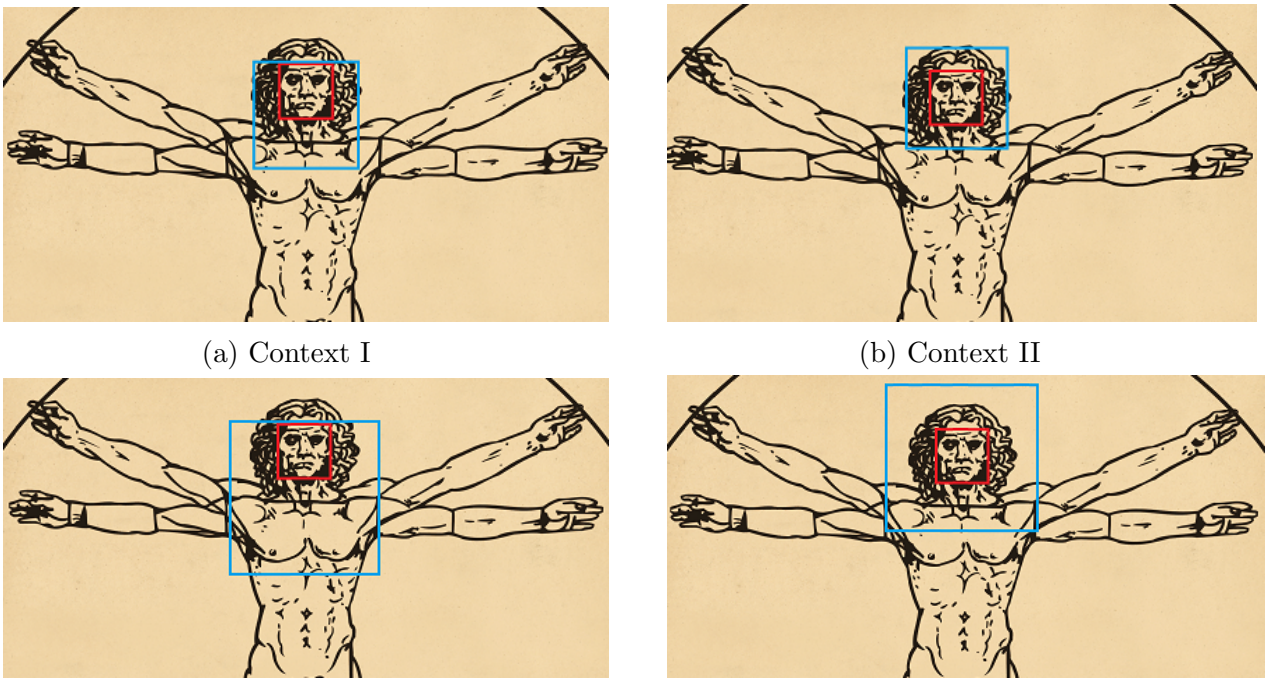
2. Context II: $[l - 0.5w, t - 0.5h, 2w, 2h]$, which is 2×2 bigger than the original region and approximately covers the hair and the neck of a person (See Figure 4.8b). This contextual information is similar to what Torralba and Sinha [58] used in their human vision experiments.

3. Context III: $[l - w, t, 3w, 3h]$, which is 3×3 bigger than the original region and approximately covers the hair below the forehead, the neck and part of the upper body of a person (See Figure 4.8c).

4. Context IV: $[l - w, t - h, 3w, 3h]$, which is 3×3 bigger than the original region and approximately covers the hair, the neck and shoulder of a person (See Figure 4.8d).

Accordingly, we modified MP-FVN so that it could pool features from both a face proposal region and its corresponding contextual region. Then features of these two regions were concatenated to represent a face proposal as a whole (See Figure 4.9). All other structures were exactly the same as in Figure 4.2.

Table 4.2 shows the test results of MP-FVN with different contextual information. We can observe that the overall recall rate has been improved by about 2% by adding contextual information. Noting the fine-grained scale partitions, we discern that contextual



(a) Context I (b) Context II

(c) Context III (d) Context IV

Fig. 4.8 Different contextual information (blue boxes) used in experiments. Red boxes are face regions.

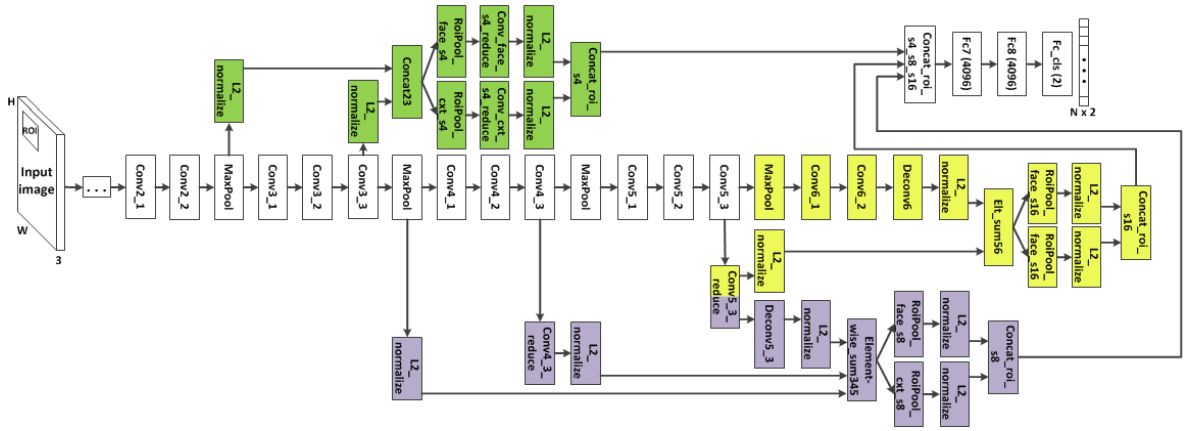


Fig. 4.9 The architecture of the proposed Multi-Path Face Verification Network (MP-FVN) with contextual information.

information behaves differently for different scale ranges. For small and medium face sizes (5-128 pixels in height), contextual information can improve recall rates by a relatively large margin (more than 1%). However, contextual information does not affect the recall rate of large-size faces very much. When comparing the four types of contextual information, we find that they have a quite similar performance, indicating the center and the size of the contextual region has a certain degree of robustness. Since Context III outperforms other context types by a slight margin, we use it in all of the following experiments.

Table 4.2 Rate recall of MP-FVN with different context information

MP-FVN type	**Scale range**								
	5-11	11-47	47-65	65-86	86-128	128-239	239-371	371-	All
No context	0.293	0.691	0.879	0.912	0.928	0.955	0.967	0.97	0.631
Context I	0.311	0.719	0.89	0.925	0.934	0.958	0.974	0.97	0.652
Context II	0.311	0.722	0.889	0.924	0.934	0.963	0.973	0.963	0.653
Context III	0.313	0.721	0.891	0.922	0.935	0.962	0.974	0.966	**0.654**
Context IV	0.313	0.72	0.892	0.924	0.936	0.958	0.972	0.959	0.653

4.1.4 What is the best ROI-pooling size?

ROI-pooling size[5] is an import factor that affects detection performance. The Fast R-CNN [16], which was used for object classification in PASCAL VOC dataset [30], employed a ROI-pooling layer of size 7×7 to pool the features from the conv5_3 feature map. This map has a stride of 16 with respect to the input image, meaning that 16 pixels in the input image are mapped to 1 pixel in the conv5_3 feature map. Since the ROI-pooling size is 7×7, the object region in the conv5_3 feature map must be equal to or larger than 7×7 to avoid a reduction in the number of bins. Accordingly, the object size in input image should at least be $(7 \times 16) \times (7 \times 16) = 112 \times 112$. Because the size of an object in PASCAL VOC dataset [30] is typically larger than 112×112, a 7×7 ROI-pooling size is a reasonable choice. However, the situation is completely different for face detection, especially for faces

[5] A ROI-pooling size of $N \times N$ means that, given an object region of arbitrary size, the ROI-pooling layer will pool features from it and output a feature block of the spatial size of $N \times N$.

of small size. Typically, for an 8×8 face processed by the s4 path, its effective resolution in the feature map of the "Concat23" layer (See Figure 4.9) is only 2×2. Even if we extract features from its corresponding 24×24 contextual region ("Context III"), the effective resolution is 6×6, still less than 7×7. As indicated in [59], when an ROI-pooling layer's effective input resolution is less than output resolution, the pooling bins collapse and the features become less discriminative. Therefore, in all the previous experiments, we used an ROI-pooling size of 4×4 as a tradeoff to extract both facial and contextual regions for all detection paths. In the following controlled experiments, we further support our choice by comparing four kinds of the ROI-pooling sizes: 3×3, 4×4, 5×5 and 7×7. The results are shown in Table 4.3. We observe that the overall recall rate drops considerably when using the traditional 7×7 pooling size. This verifies our previous discussion. In contrast, when using a smaller pooling size, such as 3×3, 4×4 or 5×5, the recall rates are similar to each other. Since 4×4 outperforms the other two by a small margin we use it in all following experiments.

Table 4.3 Rate recall of MP-FVN with different ROI-pooling sizes

MP-FVN type	Scale range								
	5-11	11-47	47-65	65-86	86-128	128-239	239-371	371-	All
3×3	0.312	0.721	0.893	0.921	0.935	0.961	0.973	0.956	0.653
4×4	0.313	0.721	0.891	0.922	0.935	0.962	0.974	0.966	**0.654**
5×5	0.31	0.719	0.891	0.923	0.937	0.958	0.974	0.966	0.652
7×7	0.299	0.656	0.866	0.907	0.925	0.947	0.963	0.939	0.612

4.1.5 Does OHEM help?

The training samples for MP-FVN are typically extremely unbalanced. This is because face regions are scarce compared to background (non-face) regions, so only a few face proposals are positive (matched to ground-truth face regions) and most of the proposals are negative (matched background regions). In previous experiments, we randomly selected a fixed number of positive and negative samples for each detection path (see section 4.1.1 for details) to provide a balanced training set. However, as indicated by [60], explicitly mining

hard examples with high training loss leads to better training and testing performance than randomly sampling all examples. In this thesis, we propose an Online Hard Example Mining (OHEM) layer specifically for MP-FVN. It is inserted between "Fc8" and "Fc_cls" layers to mine hard positive and negative examples, as shown in Figure 4.10. These selected hard examples are then used in back-propagation for updating network weights.

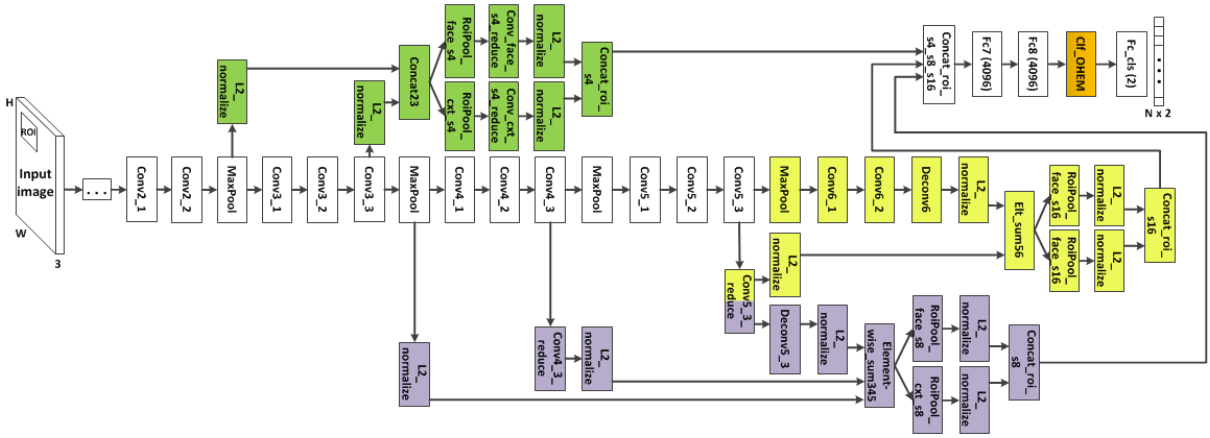


Fig. 4.10 The architecture of the proposed Multi-Path Face Verification Network (MP-FVN) with OHEM layer (orange box).

Two steps are involved in the OHEM layer:

Step 1: For each detection path (s4, s8 or s16), instead of randomly selecting a fixed number of positive and negative examples, all face proposals are processed and their facial and contextual features are extracted by ROI-pooling layers and then concatenated into a single ROI feature.

Step 2: The ROI features of face proposals from different paths are concatenated and fed forward to fc layers to obtain initial classification scores. All face proposals are sorted in the descending order of their classification loss and the top 48 samples are selected according to this order. Note that we do not set a positive-negative ration for data balancing here. As indicated in [60], if any class were neglected, its loss would increase until it has a high probability of being sampled. When there are images where face ROIs are easy (e.g., big faces with frontal views), the network can use only negative samples in a mini-batch. In contrast, when negative samples are trivial, the mini-batch can be full-face ROIs.

The proposed OHEM layer is "online" in the sense that it is seamlessly integrated into the forward pass of the network to generate a mini-batch of hard examples. Thus we do

not need to freeze the training model to mine hard examples from all training data; it is sufficient to use the hard examples to update the current model. Note that unlike [60], which freezes all fc layers for hard example mining, here our OHEM layer only freezes the last fc layer ("Fc_cls") when mining hard examples. This modification saves training time and memory usage.

Table 4.4 shows the recall rate of MP-FVN with and without an OHEM layer. We notice that the OHEM layer does not help improve the overall recall rates. We attribute this result to the following reason. In the process of mining hard examples, an OHEM layer breaks the balance of the number of examples among different detection paths. In previous experiments, we randomly selected 16, 24 and 8 examples from each detection path, so that the parameters of each detection path could be updated in a single training iteration. However, an OHEM layer may select examples from a single path and thus only update parameters of that path in a training iteration. This leads to an imbalanced training among different paths, which thus causes a drop in the recall rate in certain scale ranges.

In view of the results discussed above, we decided NOT to include the OHEM layer in our final network structure.

Table 4.4 Rate recall of MP-FVN with and without OHEM layer

OHEM	**Scale range**								
	5-11	11-47	47-65	65-86	86-128	128-239	239-371	371-	All
with	0.313	0.721	0.891	0.922	0.935	0.962	0.974	0.966	**0.654**
without	0.307	0.723	0.894	0.923	0.936	0.962	0.975	0.959	0.653

4.2 Multi-Path Face Detection Network: Putting MP-FPN and MP-FVN Together

According to the experiments and discussions in Chapter 3 and in this chapter, we are able to put forward a Multi-Path Face Detection Network (MP-FDN) for unconstrained face detection. MP-FDN is composed of two stages as shown in Figure 4.11.

Stage 1: MP-FPN

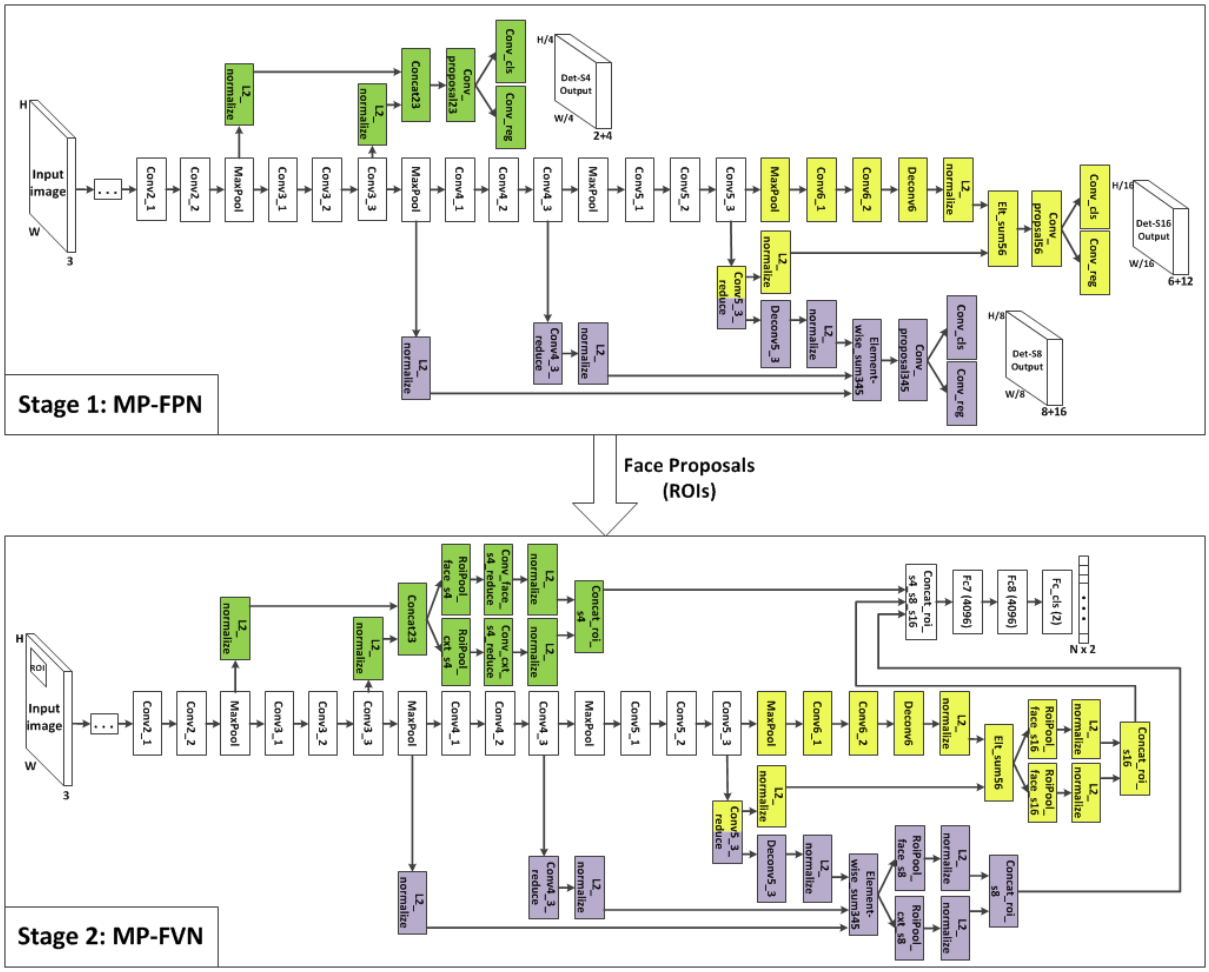


Fig. 4.11 The architecture of the proposed Multi-Path Face Detection Network (MP-FDN).

The first stage is a Multi-Path Face Proposal Network (MP-FPN) as proposed in Chapter 3. MP-FPN leverages the scale sensitivity variations of different conv-layers for generating small-, medium- and large-size face proposals through three parallel paths. As explained in Section 3.1, face proposals are parameterized relative to a set of pre-defined reference boxes, called anchors. The RPN head[6] of a detection branch predicts the "faceness" probability of each anchor allocated to this branch. At the same time, the position and size of the anchor are regressed to obtain a new box that can tightly enclose a face region. According to the scale sensitivity obtained from experiments in Chapter 3, we allocate anchors of different sizes to each detection path, as has already been shown in Table 3.6 in Chapter 3.

MP-FPN was introduced and compared with other baseline networks in Chapter 3. However, since Chapter 3 focused on comparing various face proposal networks, the training details of a specific network, like MP-FPN, was not presented. As MP-FPN is used as the first stage of our face detection framework and has not yet been discussed in detail, we provide the training details as follows.

During training, the parameters W_{mpfpn} of the MP-FPN are learned from a set of N training samples $S = \{(X_i, Y_i)\}_{i=1}^{N}$, where X_i is an image patch associated with an anchor, and $Y_i = (p_i, b_i)$ is the combination of its ground truth label $p_i = \{0, 1\}$ (0 for non-face and 1 for face) and ground truth box regression target $b_i = (b_i^x, b_i^y, b_i^w, b_i^h)$ that is associated with the ground truth face region. The latter parameterizations are the same as the four coordinates in [20]: $b_i^x = (x_{gt} - x_i)/w_i, b_i^y = (y_{gt} - y_i)/h_i, b_i^w = \log(w_{gt}/w_i), b_i^h = \log(h_{gt}/h_i)$, where x, y, w, h denote the two coordinates of the box center, width, and height. Variables x_i, x_{gt} are for the image patch X_i and its ground truth face region X_i^{gt} respectively (likewise for y, w, and h). We define the loss function for MP-FPN as

We define the loss function for MP-RPN as

$$l(W_{mpfpn}) = \sum_{m=1}^{M} \alpha_m L^m(\{(X_i, Y_i)\}_{i \in S^m} | W_{mpfpn}) \tag{4.1}$$

where $M = 3$ is the number of detection branches, α_m is the weight of loss function L^m, and $S = \{S^1, S^2, ..., S^M\}$, where S^m contains the training samples of the m^{th} detection branch. The loss function for each detection branch contains two objectives

[6]See section 3.2 for the definition of a RPN head.

$$L^m(\{(X_i, Y_i)\}_{i \in S^m} | W_{mpfpn}) = \frac{1}{N_m} \sum_{i \in S^m} L_{cls}(p(X_i), p_i) + \lambda \, [\![p_i = 1]\!] \, L_{reg}(b(X_i), b_i) \quad (4.2)$$

where N_m is the number of samples in the mini-batch of the m$^{\text{th}}$ detection branch, $p(X_i) = (p_0(X_i), p_1(X_i))$ is the probability distribution over the two classes, non-face and face, respectively. L_{cls} is the cross-entropy loss (aka., "softmax loss"), $b(X_i) = (b^x(X_i), b^y(X_i), b^w(X_i), b^h(X_i))$ is the predicted bounding box regression target, L_{reg} is the smoothL1 loss function defined in [16] for bounding box regression and λ is a trade-off coefficient between classification and regression. Note that L_{reg} is computed only when a training sample is positive ($[\![p_i = 1]\!]$).

The trained model that uses the loss function in equation 4.1 provides a face proposal (including a bounding box and its confidence score) corresponding to each anchor. Only high-scoring face proposals are fed forward to the second stage for further verification.

Stage 2: MP-FVN

The second stage is a Multi-Path Face Verification Network (MP-FVN) as proposed in section 4.1. MP-FVN utilizes the same three parallel paths to extract both facial and contextual features for small-, medium- and large-size face proposals, respectively, and a final classification score is given to each face proposal according to its concatenated facial and contextual features. As discussed in Section 4.1, we allocate face proposals to different branches according to the size of the face. This follows the scale sensitivity rule[7] that was determined from experiments described in Chapter 3. Table 4.5 shows the face proposal height range allocated to each detection branch.

Table 4.5 Face Proposal Allocation for MP-FVN

Detection Path	**s4**	**s8**	**s16**
Face Proposal Height (x)	$x < 12$	$12 \leq x < 128$	$x \geq 128$

During training, the parameters W_{mpfvn} of the MP-FVN are learned from a set of K

[7]The scale sensitivity rule was described in Section 3.3. Specifically, the conv23_s4 path is most sensitive to facial regions below 12 pixels in height, conv345_s8 is most sensitive to facial regions with a height between 12 and 128, and conv56_s16 is most sensitive to facial regions larger than 128 pixels in height.

training samples $T = \{(X_i, Y_i)\}_{i=1}^{K}$, where X_i is a face proposal obtained from MP-FPN, and $Y_i = l_i$ is its ground truth label[8] $l_i = \{0, 1\}$ (0 for non-face and 1 for face). We define the loss function for MP-FVN as

$$l(W_{mpfvn}) = \sum_{m=1}^{M} \beta_m \Phi^m(\{(X_i, Y_i)\}_{i \in T^m} | W_{mpfvn}) \tag{4.3}$$

where $M = 3$ is the number of detection branches, β_m is the weight of the loss function Φ^m, and $T = \{T^1, T^2, ..., T^M\}$, where T^m contains the training samples of the m^{th} detection branch. The loss function for each detection branch is

$$\Phi^m(\{(X_i, Y_i)\}_{i \in T^m} | W_{mpfvn}) = \frac{1}{G_m} \sum_{i \in T^m} L_{cls}(p(X_i), l_i) \tag{4.4}$$

where G_m is the number of samples in the mini-batch of the m^{th} detection branch, $p(X_i) = (p_0(X_i), p_1(X_i))$ is the probability distribution over the two classes, non-face and face, respectively. L_{cls} is the cross-entropy loss (aka., "softmax loss").

With a trained MP-FVN model using the loss function in equation 4.3, we can obtain an updated confidence score for each face proposal. The new confidence scores are used as final "faceness" probabilities.

4.3 Conclusion

This chapter first proposed a Multi-Path Face Verification Network (MP-FVN) for eliminating difficult false positives and include different false negatives from the set of face proposals given by MP-FPN. A series of controlled experiments were conducted to select the optimal hyper-parameters and structure for MP-FVN, including contextual region size, ROI-pooling size, and whether to include an OHEM layer or not. Next, this MP-FVN and the MP-FPN proposed in Chapter 3 were assembled to form the final Multi-Path Face Detection Network (MP-FDN). The loss functions and overall training and testing procedures of MP-FDN are exposed in details.

At this point, having assembled all the necessary components, we turn to the next chapter where we describe the experiments that were performed on actual data.

[8] A face proposal is labeled as 1 when it has an Intersection-Over-Union (IOU) ratio equal or larger than 0.5, and 0 otherwise.

Chapter 5

Experiments and Results

This chapter validates the effectiveness of the proposed Multi-Path Face Detection Network (MP-FDN). The first section presents the training and testing datasets. The second section describes the specifics of the experimental settings. Experimental results on WIDER FACE [4] and FDDB [1] datasets will be reported in detail in the next two sections. Lastly, a chapter conclusion is presented.

5.1 Datasets

As introduced and discussed in Chapter 2, we will use the WIDER FACE [4] training set to train the proposed MP-FDN. The effectiveness of MP-FDN will be benchmarked on the WIDER FACE test set, WIDER FACE validation set and FDDB dataset [1]. The main characteristics of WIDER FACE and FDDB datasets are summarized as follows.

WIDER FACE

The WIDER FACE dataset [4] contains 32,203 images with 393,703 labeled human faces (each image has an average of 12 faces). Faces in this dataset have a high degree of variability in scale, pose, occlusion, lighting conditions, and image blur. They are organized based on 61 event classes. For each event class, 40%, 10% and 50% of the images are randomly selected for the training, validation and test sets. As a result, there are 12,880, 3,226 and 16,097 images in the training, validation and test sets, respectively. Both the images and their associated ground truth labels used for training and validation are available

online[1] for training a face detection model and selecting the hyper-parameters for the model. However, for the test set, only the images are available and the detection results need to be submitted to an evaluation server in order to obtain the Precision-Recall curves.

Moreover, both the validation and test sets are divided into three levels of difficulty: "Easy", "Medium", "Hard" based on the detection rate of EdgeBox [5], so that the Precision-Recall curves need to be reported for each difficulty level[2]. Although we have used the WIDER FACE validation set to select the hyper-parameters for MP-FPN and MP-FVN, images in the validation set are never used in the training process. Therefore, we are safe to evaluate the detection results on both the test and the validation sets of the WIDER FACE dataset. The WIDER FACE dataset employs the PASCAL VOC [30] evaluation metric for evaluating the detection results. Specifically, if the ratio of the intersection of a predicted face region with an annotated face region over the union of these two regions is greater than 0.5, a score of 1 is assigned to the detected region, and 0 otherwise.

FDDB

The FDDB dataset [1] has been a standard database for evaluating face detection algorithms over the past eight years. It contains the annotations for 5,171 faces in a set of 2,845 images. Each image in the FDDB dataset has less than two faces on average. These faces mostly have large sizes compared to those in the WIDER FACE dataset. FDDB uses a bounding ellipse to annotate each face region. Two types of evaluation metrics are provided for evaluating detection results on the FDDB dataset, discrete score and continuous score. The discrete score metric is the same as the PASCAL VOC evaluation metric used in WIDER FACE. For the continuous score criterion, the Intersection-over-Union (IoU) ratio is used directly as the score for the detected region without any thresholding.

Like most face detection algorithms, MP-FDN uses bounding rectangles[3] to describe predicted face regions, while the FDDB dataset applies bounding ellipses to annotate ground-truth face regions. Due to the shape inconsistency, the continuous score for a face is lower when directly computing the IoU ratio of a rectangle and an ellipse. Some

[1] `http://mmlab.ie.cuhk.edu.hk/projects/WIDERFace/index.html`

[2] Users of this test set have no knowledge about the difficulty level of the images in the test set. In fact, it is necessary to submit all predicted face boxes to the server, which then provides three ROC curves based on the unknown "hard", "medium" and "easy" data partitions.

[3] This is because MP-FDN is trained on the WIDER FACE training set, which employs bounding rectangles to annotate ground-truth face regions.

face detection algorithms employ a post-hoc elliptical regressor to transform the predicted bounding rectangles to bounding ellipses in order to improve the continuous score. This may cause confusion when comparing different algorithms since we do not know whether a top-ranking algorithm is due to its original bounding box prediction or its special post-hoc elliptical regression. In contrast, under the discrete score criterion, various algorithms can directly compare their original bounding box predictions with the ground truth ellipses because an IoU threshold of 0.5 can mitigate the shape differences between a predicted bounding box and a ground-truth bounding ellipse. Therefore, we only employ the discrete score metric to report our detection results.

5.2 Training and Testing Settings

The code for MP-FDN (including MP-FPN and MP-FVN) was built using Caffe [56] and its MATLAB interface (aka matcaffe [56]). The detailed training and testing settings are as follows.

5.2.1 Training Settings

MP-FPN

We use all of the 12,880 images in the WIDER FACE training set to train an MP-FPN detector. These images are processed as follows. Given a training image I and a set of m bounding box annotations $\{[x_1, y_1, w_1, h_1],\ ,\ [x_m, y_m, w_m, h_m]\}$, that indicate the center coordinates, width and height of all the m faces in this image, we first obtain the maximum height of all these bounding boxes: $h_{max} = max(h_1, h_2, , h_m)$. If h_{max} is larger than 500 pixels, we down-sample image I to a half to obtain image $I_{\times 0.5}$. If h_{max} is smaller than 250 pixels, we up-sample image I twice to obtain $I_{\times 2}$. Both down-sampling and up-sampling were done using bicubic interpolation. Otherwise, if hmax is between 250 and 500 pixels, we flip the columns of image I left to right to obtain I_{flip}. Therefore, for each image I, we can obtain a transformed image I_{trans}, where

$$I_{trans} = \begin{cases} I_{\times 0.5}, & \text{if } h_{max} > 500 \\ I_{flip}, & \text{if } 250 \leq h_{max} \leq 500 \\ I_{\times 2}, & \text{if } h_{max} < 250 \end{cases}$$

Then we randomly cropped an 800 x 800 image patch[4] from I and I_{trans}, respectively. In this way, two 800×800 images are generated to replace the original image I and employed for training[5]. See Figure 5.1 as an illustration.

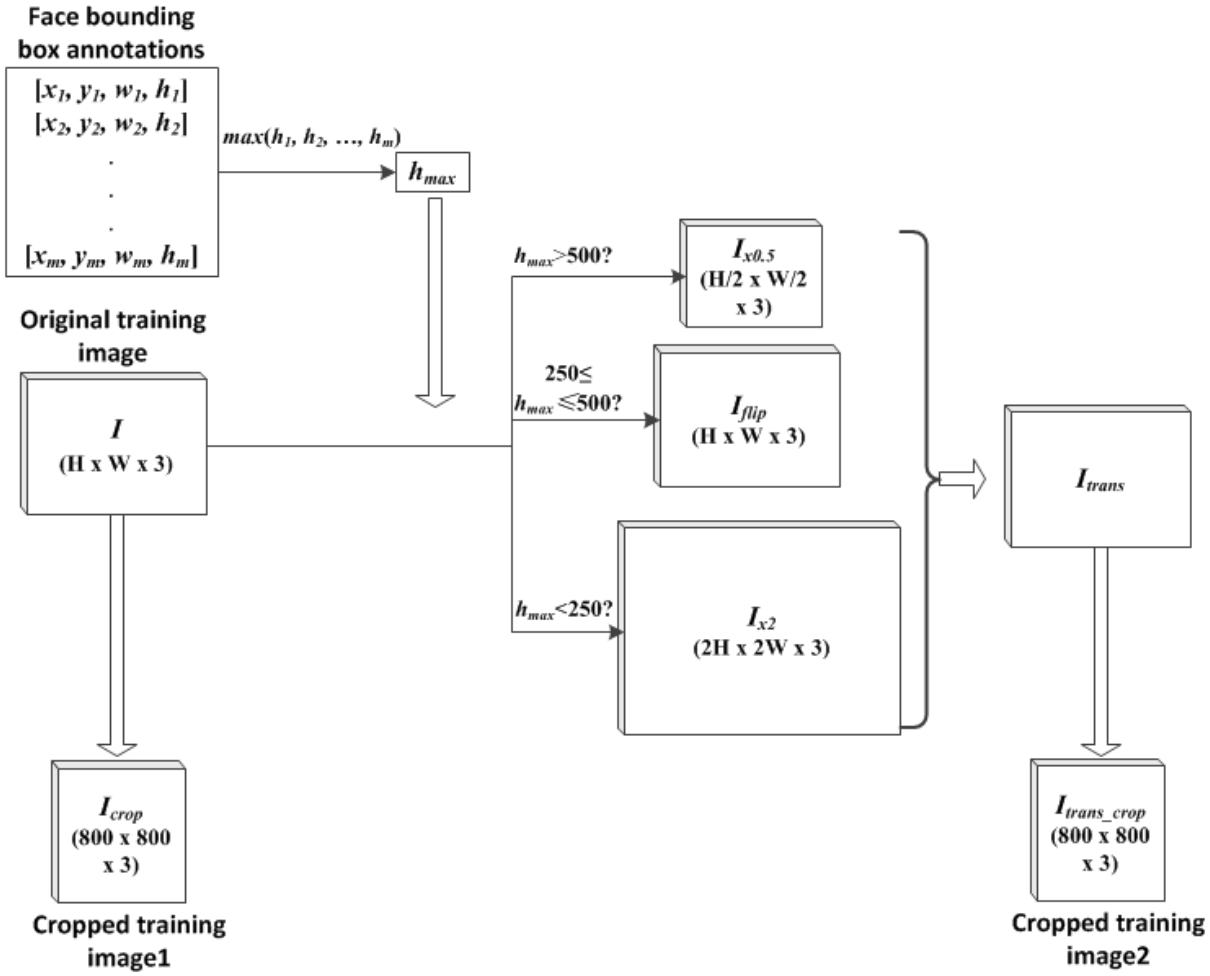


Fig. 5.1 Illustration of training data preparation.

As a result, the training set contains 25,760 images rather than 12,880 images. This choice of data preparation has three advantages. First, the original images in WIDER FACE

[4]If the original image has one or both sides containing less than 800 pixels, we pad the cropped image patch with zeros.

[5]The advantage of this data preparation method has been explained in detail in Section 3.2. In all controlled experiments in chapter 3 and 4, we used 512×512 image patches for training. But here we use larger 800×800 image patches to include more face regions.

have a relatively large and variable size, which fluctuates around 900×1024. However, after cropping to a fixed size (800×800), multiple images could be trained simultaneously (mini-batch size¿1), which improves the training efficiency. Second, as shown in Figure 3.1a, small faces (between 10-50 pixels in height) dominate the dataset. Up-sampling images can help enlarge the size of these faces, thus increasing the number of medium (between 50-300 pixels in height) and large (over 300 pixels in height) ones. Third, a small number of extra large faces are over 500 pixels in height, so we down-sample these images to reduce their size so that they can be associated with at least one of the anchors in the training process[6].

The backbone architecture of MP-FPN, VGG16, was pretrained on the ImageNet dataset as found in [53]. The weights of all newly added convolutional layers were randomly initialized from a zero-mean Gaussian distribution with a standard deviation of 0.01. We used the following $k = 7$ anchor[7] scales: 8, 16, 32, 64, 128, 256 and 480. These are allocated to three detection paths according to Table 3.6. The aspect ratio was set to 1 for all anchors. Furthermore, an anchor was assigned as a positive sample if it had an intersection-over-union (IOU) ratio greater than 0.5 with any ground truth box, and as a negative sample if it had an IOU ratio less than 0.3 with any ground truth box. Each mini-batch contained 2 images of the size 800×800. In addition, each image had 56 sampled anchors: 16 for Det-s4 path, 32 anchors for Det-s8 path and 8 anchors for Det-s16 path (see the structure of MP-FPN in Figure 4.11). This allocation ratio was based on the number of ground truth faces within the scale range of each detection path. The ratio of positive to negative samples was set to 1:3 for all detection branches. All RPNs were trained by back-propagation and stochastic gradient descent (SGD) [57], using a learning rate of 0.0005 for 90k mini-batches, and 0.00005 for another 30k mini-batches. A momentum of 0.9 and a weight decay of 0.0005 were employed.

We used the trained MP-FPN model to obtain a set of face proposals for each training image. Then we eliminated all proposals with a confidence score less than 0.1. Finally, non-maximum suppression (NMS) with a threshold of 0.7 was adopted to filter the remaining

[6] As discussed below, the largest anchor size that we used in training was 480×480. When a face is much larger than 480×480, it cannot be associated with any anchors and thus will not contribute to the training process. To avoid this situation, we down-sample an image that contained faces large than 500 pixels in height by half so that they could be associated with an anchor. See section 4.2 for details of the association of an anchor and a ground-truth bounding box.

[7] See section 3.1 of an explanation of the anchors.

proposals based on their confidence scores. The remaining proposals were later used for training MP-FVN.

MP-FVN

We used the same 25,760 training images, as well as their corresponding face proposals as described above, to train MP-FVN. The trained MP-FPN model was used to initialize MP-FVN parameters. Specifically, since MP-FVN and MP-FPN share convolutional layers in conv-stage1-6, as well as "Conv4_3_reduce" and "Conv5_3_reduce" (see Figure 4.11), the parameters of these conv-layers in MP-FVN were initialized by the parameters of corresponding conv-layers in MP-FPN. The weights of all newly added conv-layers and fully-connected layers in MP-FVN were randomly initialized from a zero-mean Gaussian distribution with a standard deviation of 0.01. A face proposal was assigned as a positive sample if it had an intersection-over-union (IOU) ratio greater than 0.5 with any ground truth box, and as a negative sample if it had an IOU ratio with any ground truth box less than 0.3. Each mini-batch contained 1 image of the size of 800×800. For each training image, we assigned 16 sampled face proposals for the s4 path, 32 for the s8 path and 8 for the s16 path. This allocation ratio was based on the number of ground truth faces within the scale range of each detection path. The ratio of positive and negative samples was set to 1:3 for all detection paths. MP-FVN was trained by back-propagation and stochastic gradient descent (SGD) [57], using a learning rate of 0.0005 for 25k mini-batches, and 0.00005 for another 25k mini-batches. A momentum of 0.9 and a weight decay of 0.0005 were used.

5.2.2 Testing Settings

Given a test image I_{test}, we first down-sampled it by half to obtain image $I_{test0.5}$. Then both I_{test} and $I_{test0.5}$ were fed to MP-FPN to obtain two sets of face proposals. The proposals of $I_{test0.5}$ were enlarged twice so that they could match the face sizes in the original image I_{test}. Next, the two sets of proposals are combined into a single face proposal set. We eliminated all proposals with a confidence score less than 0.6 from the face proposal set. After that, non-maximum suppression (NMS) with a threshold of 0.7 was adopted to filter the proposals based on their confidence scores. The remaining proposals and the original image I_{test} were fed to MP-FVN to obtain an updated confidence score for each proposal. Finally, non-maximum suppression (NMS) with a threshold of 0.33 was adopted to filter the

proposals based on their new confidence scores. The resulting face proposals were regarded as predicted face regions. Note that we use $I_{test0.5}$ for proposing extra-large faces since there are some extra-large faces in the WIDER FACE dataset. For example, some faces have a height larger than 1000 pixels. These faces cannot be captured in the original image since the largest anchor scale of MP-FPN is only 480 pixels[8]. By down-sampling the original image, the face size is also down-sampled by a half. This guarantees that these large faces can be proposed by a certain detection path of MP-FPN. Since the area of $I_{test0.5}$ is 25% of the original image area, it only increases the computational load of MP-FPN by 25%, and does not affect the computational load MP-FVN at all.

5.3 Results on the WIDER FACE Dataset

In this section, we first compare the proposed MP-FDN to other strong baseline algorithms on WIDER FACE dataset. Then qualitative results of MP-FDN are presented. Next, we show the sensitivity of MP-FDN to different facial attributes. Finally, false positives and false negatives are analyzed in details.

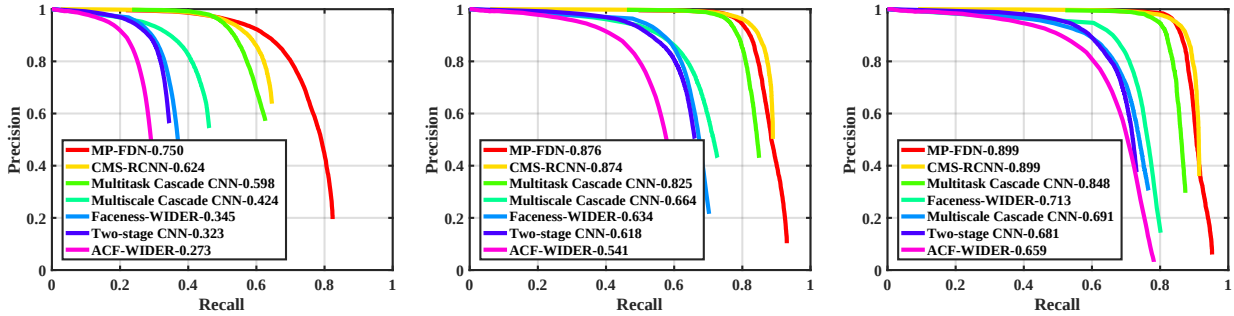
5.3.1 Precision-Recall Curves

We compare the proposed MP-FDN with all six strong face detection methods available on the WIDER FACE website: Two-stage CNN [4], Multiscale Cascade [4], Multitask Cascade [40], Faceness [9], Aggregate Channel Features (ACF) [8] and CMS-RCNN [43].

Figure 5.2a, 5.2b, 5.2c shows the Precision-Recall curves and the Average Precision values of the different methods on the Hard, Medium and Easy partitions of the WIDER FACE *validation set*, respectively. The proposed MP-FDN algorithm consistently ranks in first place on all partitions. On the hard partition, the proposed MP-FDN outperforms all six strong baselines by a large margin. Specifically, it achieves an increase of 12.6% in Average Precision compared to the 2$^{\text{nd}}$ place CMS-RCNN method. On the medium partition, MP-FDN still ranks in first place, outperforming the 2$^{\text{nd}}$ place CMS-RCNN

[8] As explained in Chapter 3, face proposals are generated by regressing the position and size of an anchor that is located close to a ground-truth face region. More importantly, the anchor and the ground-truth face region should have similar sizes. For example, a 500×500 face region can be proposed by a 480×480 anchor. However, an 800×800 face region cannot be proposed by a 480×480 anchor due to its large size difference.

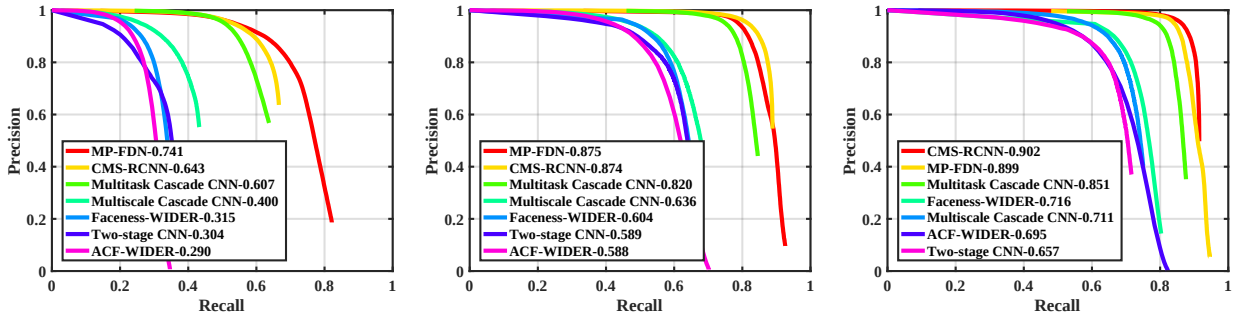
method by a small margin of 0.2% in Average Precision. Lastly, on the easy partition, the proposed MP-FDN and the strong baseline CMS-RCNN method are tied for the first place.



(a) Hard Set (b) Medium Set (c) Easy Set

Fig. 5.2 Precision-Recall Curves of WIDER FACE validation set.

Figure 5.3a, 5.3b, 5.3c shows the Precision-Recall curves and the Average Precision values on the Hard, Medium and Easy partitions of the WIDER FACE *test set*. On the hard partition, the proposed MP-FDN still outperforms all other strong baselines by a large margin. Specifically, it achieves an increase of 9.8% in Average Precision compared to the 2nd place CMS-RCNN method. On the medium partition, MP-FDN outperforms the 2nd place CMS-RCNN method by a small margin of 0.1% in Average Precision. However, on the easy partition, the proposed MP-FDN lags behind CMS-RCNN method by a small margin of 0.3% in Average Precision.



(a) Hard Set (b) Medium Set (c) Easy Set

Fig. 5.3 Precision-Recall Curves of WIDER FACE test set.

The above results demonstrate the overall superior performance of the proposed MP-FDN for tackling the challenge of large-scale variation in unconstrained face detection.

Specifically, for the easy and medium partitions of the WIDER FACE dataset that contains large- and medium-size faces, MP-FDN matches or even slightly outperforms the state-of-the-art face detection algorithms. For the hard partition of the WIDER FACE dataset that contains mostly tiny faces, MP-FDN outperforms the state-of-the-art methods by a large margin. We note that the hard partition contains ALL faces greater than 10 pixels in height, so not only tiny faces, but also medium and large-size faces are included in this partition[9]. Therefore, MP-FDN's remarkable improvement in Average Precision on the hard partition accurately represents its superior performance on the full range of face scales.

5.3.2 Qualitative Results

Qualitative results of MP-FDN on the WIDER FACE validation and test sets are shown in Figure 5.4. We observe that MP-FDN exhibits a high level of robustness to variations in pose, occlusion, illumination, facial scale, facial expression, and out-of-focus blur.

5.3.3 Fine-grained Attributes Analysis

In this subsection, we study the sensitivity of the proposed MP-FDN to six facial attributes: Aspect Ratio, Bounding Box Area (BBox Area), Bounding Box Height (BBox Height), Blur, Expression, Illumination, Occlusion and Pose. The study is based on the hard partition of the WIDER FACE validation set[10]. This partition contains 31,958 faces with their ground-truth bounding boxes.

Aspect Ratio is defined as ground-truth face height divided by ground-truth face width. BBox Area is the pixel area of the ground-truth face bounding box. BBox Height is the height (in pixels) of the ground-truth face bounding box. Inspired by [61], we sort these

[9]We note that in the validation set of WIDER FACE dataset, the hard partition contains all 31,958 faces equal to or larger than 9 pixels in height. The medium partition contains all 13,319 faces equal to or larger than 16 pixels in height. The easy partition contains all 7,211 faces equal to or larger than 34 pixels in height. In other words, the hard partition is a super-set of both the medium and easy partitions, and the medium partition is a super-set of the easy partition. Although we do not know the detailed number of faces in each partition in the WIDER FACE test set, faces in the test set should be allocated to the easy, medium and hard partitions, respectively following the same distribution rules.

[10]We use the validation set rather than the test set because the former has ground-truth labels while the latter does not. Moreover, we only use the hard partition of the validation set because it contains all of the faces in this set that are greater than 10 pixels in height. In other words, it is a superset of both the medium and the easy partitions.

Fig. 5.4 Qualitative results on the WIDER FACE [4] validation and test sets

three facial attributes in the descending order and then partition them into five categories: extra-small (XS: bottom 10%), small (S: next 20%), medium (M: next 40%), large (L: next 20%), and extra-large (top 10%). The other five attributes, Blur, Expression, Illumination, Occlusion and Pose follow the definitions in the WIDER FACE technical report [4]. The category partitions of the eight facial attributes are summarized in Table 5.1.

Table 5.1 Category partitions of facial attributes

Attribute	**Partition**
Aspect Ratio (x)	XS($x \leq 1.02$), S($1.02 < x \leq 1.17$), M($1.17 < x \leq 1.38$), L($1.38 < x \leq 1.6$), XS($x > 1.6$)
BBox Area (y)	XS($y \leq 112$), S($112 < y \leq 224$), M($224 < y \leq 1260$), L($1260 < y \leq 6030$), XS($y > 6030$)
BBox Height (z)	XS($z \leq 12$), S($12 < z \leq 17$), M($17 < z \leq 40$), L($40 < z \leq 89$), XS($z > 89$)
Blur	Clear, Normal Blur, Heavy Blur
Expression	Typical expression, Exaggerated Expression
Illumination	Normal illumination, Extreme illumination
Occlusion[11]	No occlusion (None), Partial occlusion, Heavy occlusion
Pose[12]	Typical pose, Atypical Pose

For each facial attribute, the number of ground-truth faces in each of its category partitions is imbalanced. For example, there are much more faces in the typical expression than in the exaggerated expression category. To enable a fair comparison of imbalanced categories of a facial attribute, we employed the average normalized precision (APN) in [61] to describe the face detection performance for each category of a facial attribute. The average normalized precision (AP_N) is calculated as follows:

$$AP_N = \frac{1}{N_c} \sum_{c} \frac{R(c).N}{R(c).N + F(c)}$$

[11]Partial occlusion is when 1%-30% facial area is occluded. Heavy occlusion is when over 30% of the facial area is occluded.

[12]A facial pose is atypical when either the roll or pitch is larger than 30 degrees, or the yaw is larger than 90 degrees. Otherwise, the facial pose is typical.

where $R(c)$ is the fraction of faces detected with confidence of at least c. N is roughly equal to the average number of faces in each facial attribute category. $F(c)$ is the number of incorrect detections with at least a confidence of c. N_c is number of confidence score partitions. We set N to 9500[13]. All other parameters were set by default. Refer to the publicly available code[14] in [61] for more details.

Figure 5.5 shows the sensitivity of MP-FDN to the eight facial attributes indicated by the average normalized precision. For BBox Height (Fig. 5.5a), MP-FDN performs under average when the face bounding box height is small or extra-small. Since BBox Height is positively correlated with BBox Area, a similar result appears when the face bounding box area is small or extra-small, as shown in Figure 5.5b. For Aspect Ratio (Fig. 5.5c), MP-FDN performs under average when the facial height-width ratio is either extra-small or extra-large.

MP-FDN also performs under average in the case of heavy blur, partial or heavy occlusion, and atypical pose. In contrast, MP-FDN is quite robust to exaggerated expression and extreme illumination: it performs above average in these two situations.

Figure 5.6 summarizes the impact of different facial attributes in the same plot. We observe that the MP-FDN is most sensitive to BBox Height (AP_N ranges from 0.168 to 0.945) and BBox area (AP_N ranges from 0.144 to 0.946). The next two important factors are Blur (from 0.418 to 0.906) and Occlusion (from 0.266 to 0.777), followed by Aspect Ratio and Expression. The latter two influence face detection performance in different directions: extreme aspect ratio negatively affects the AP_N (0.436) while exaggerated expression positively affects the AP_N (0.891). Lastly, MP-FDN is least sensitive to Illumination and Pose. While atypical pose causes a little drop of AP_N from 0.656 to 0.550, extreme illumination leads to a small increase of AP_N from 0.648 to 0.696.

5.3.4 Hard False Positive Analysis

We select the top-100 high-scoring false positives as shown in Figure 5.7. Due to their high confidence scores, they should the non-face objects that most resemble facial patterns. However, we observe that most of them are real human faces miss-labeled by the authors

[13] 9500 was obtained as follows. There are 31,958 faces in total. They need to be partitioned according 8 facial attributes and there are 27 categories in total for the 8 attributes. So for each category, the average number of face is $31,958 \times 8/27 \approx 9469$. We approximate this to obtain 9500.

[14] `http://dhoiem.web.engr.illinois.edu/projects/detectionAnalysis/`

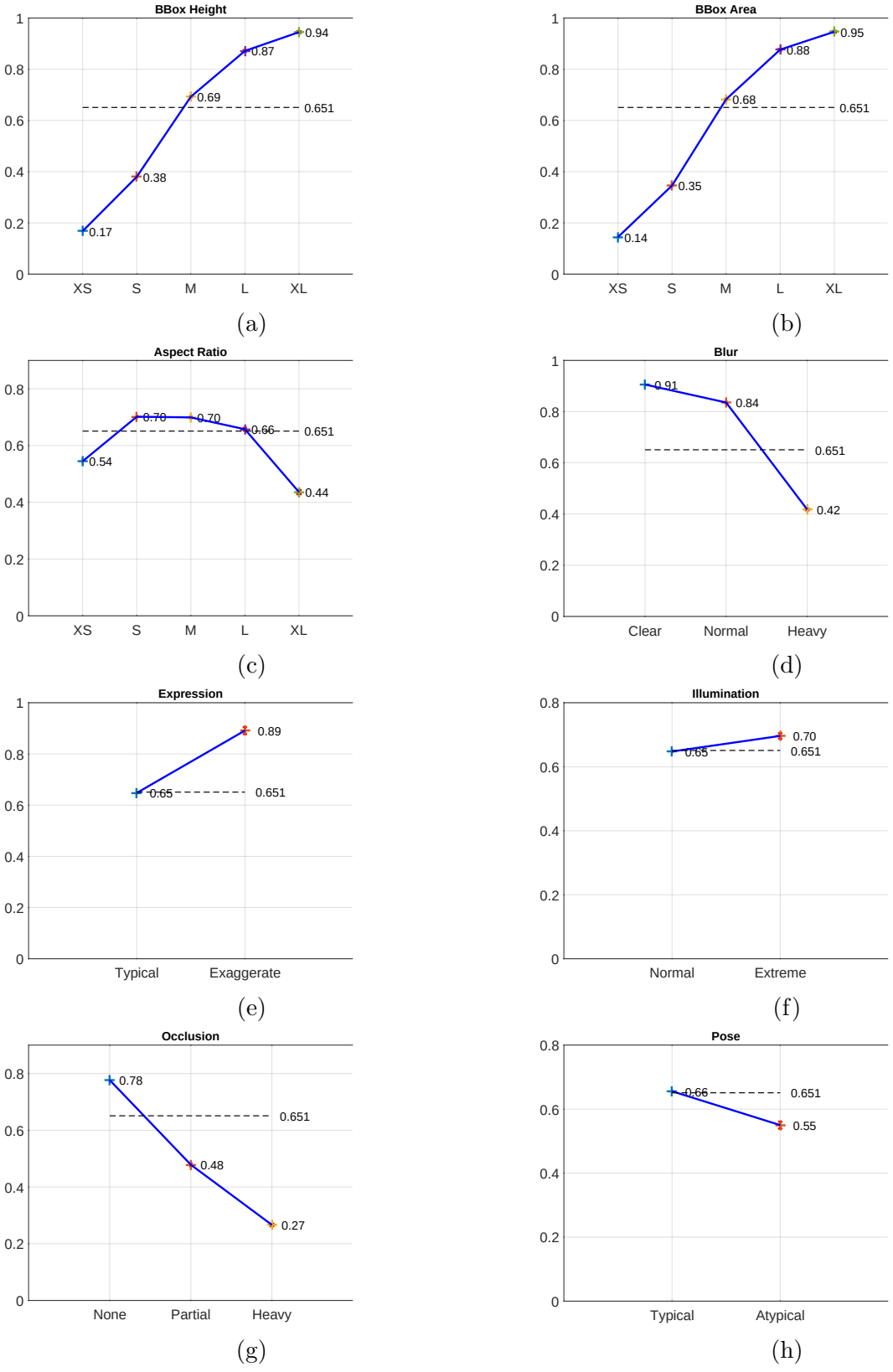


Fig. 5.5 Sensitivity to different facial attributes. The normalized Average Precision (AP_N) is shown for each facial attributes.

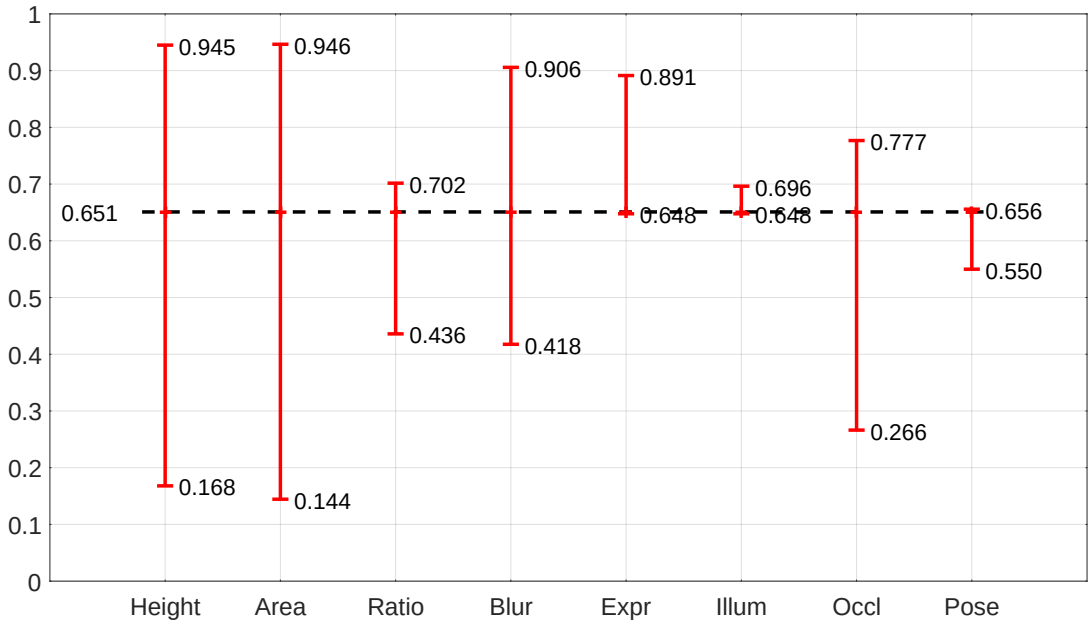


Fig. 5.6 A summary of the impact of different facial attributes. The maximum and minimum average normalized precision (AP_N) is plotted for each attribute. "Height" indicates "BBox Height" in Figure 5.5. Similarly, "Area" indicates "BBox Area", "Ratio" indicates "Aspect Ratio", "Expr" indicates "Expression", "Illum" indicates "Illumination", and "Occl" indicates "Occlusion".

of WIDER FACE dataset [4].

A detailed analysis of the reasons for the top-100 false positives is given in Table 5.2.

Table 5.2 Reasons for the top-100 false positives

Reason	**Percentage**
Miss-labeling	60%
Cartoon character/ mask/ figure carving	10%
Inaccurate localization	9%
Non-face object	21%

5.3.5 Hard False Negative Analysis

We also selected the top-100 low-scoring false negatives as shown in Figure 5.8. We found that about half of these faces would hardly be recognized as such, even by a human. The other half are due to heavy occlusion, blur, low-resolution, atypical pose, facial incompleteness, or a combination of these factors. A detailed analysis of the reasons of these false negatives is shown in Table 5.3.

Table 5.3 Reasons for the top-100 false negatives

Reason		**Percentage**
Undetectable by humans		51%
Detectable by humans	Heavy occlusion	29%
	Blur or Low resolution	11%
	Atypical pose	6%
	Incomplete face	3%

5.4 Results on the FDDB Dataset

To show the general face detection capability of the proposed MP-FDN method, we directly apply MP-FDN previously trained on the WIDER FACE training set to the FDDB dataset.

Fig. 5.7 Top-100 high-scoring false positives obtained for the WIDER FACE [4] validation set. These are sorted in a descending order from left to right and from top to bottom, according to their confidence scores.

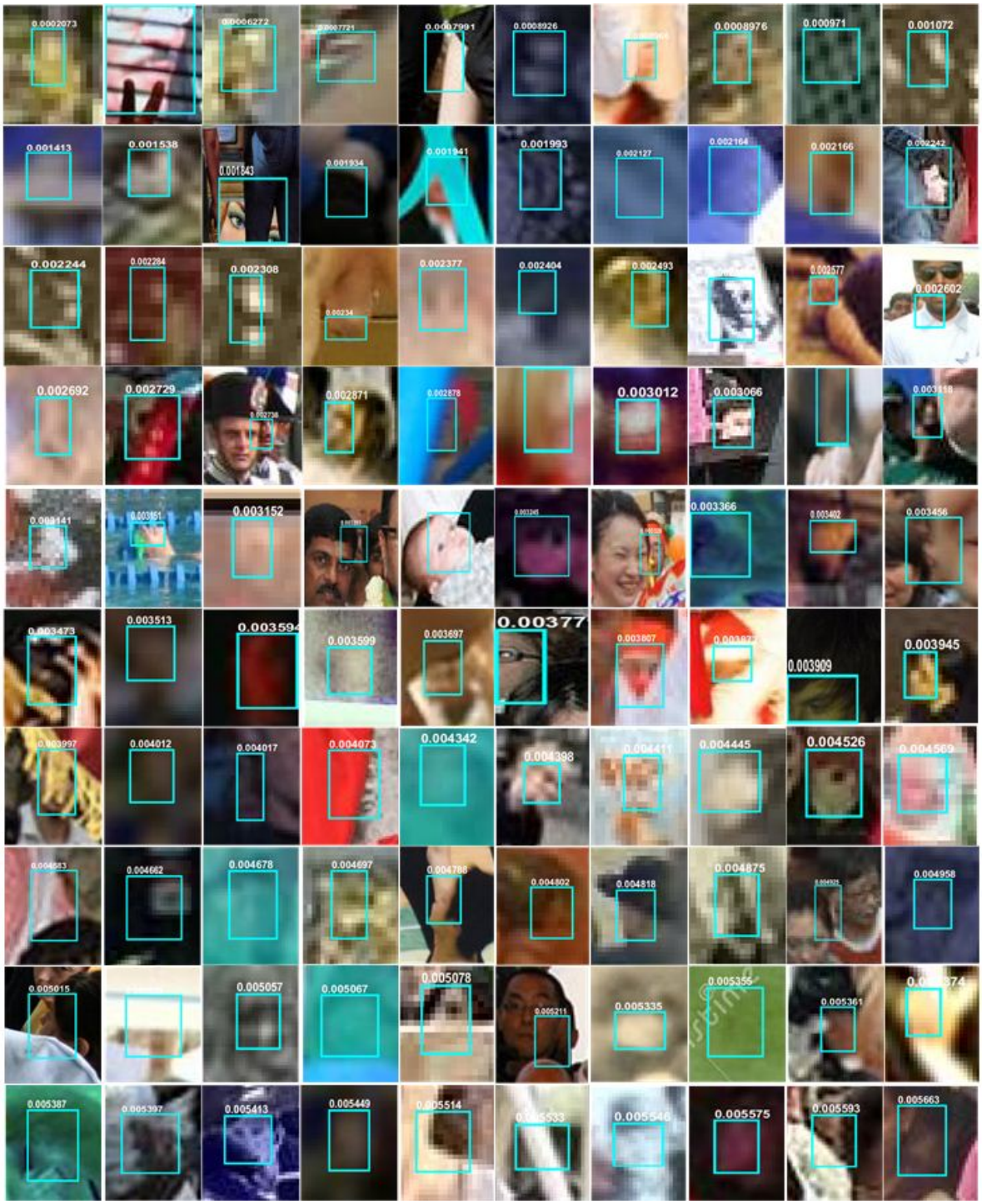


Fig. 5.8 Top-100 low-scoring false negatives of WIDER FACE [4] validation set. They are sorted in ascending order from left to right and from top to bottom, according to their confidence score.

We first compare the result of MP-FDN with other strong baseline algorithms. Then, qualitative results of MP-FDN are presented.

5.4.1 ROC Curves

We make a comprehensive comparison with 15 other typical baselines: Viola-Jones [6], SurfCascade [44], ZhuRamanan [3], NPD [36], DDFD [38], ACF [8], CascadeCNN [39], CCF [62], JointCascade [37], HeadHunter [7], FastCNN [63], Faceness [8], HyperFace [41], MTCNN [40] and UnitBox [64]. Figure 5.9 shows the ROC curves of these sixteen methods using the discrete score criterion. As shown in Figure 5.9, the proposed MP-FDN outperforms ALL of the other 15 methods and has the highest average recall rate (0.973).

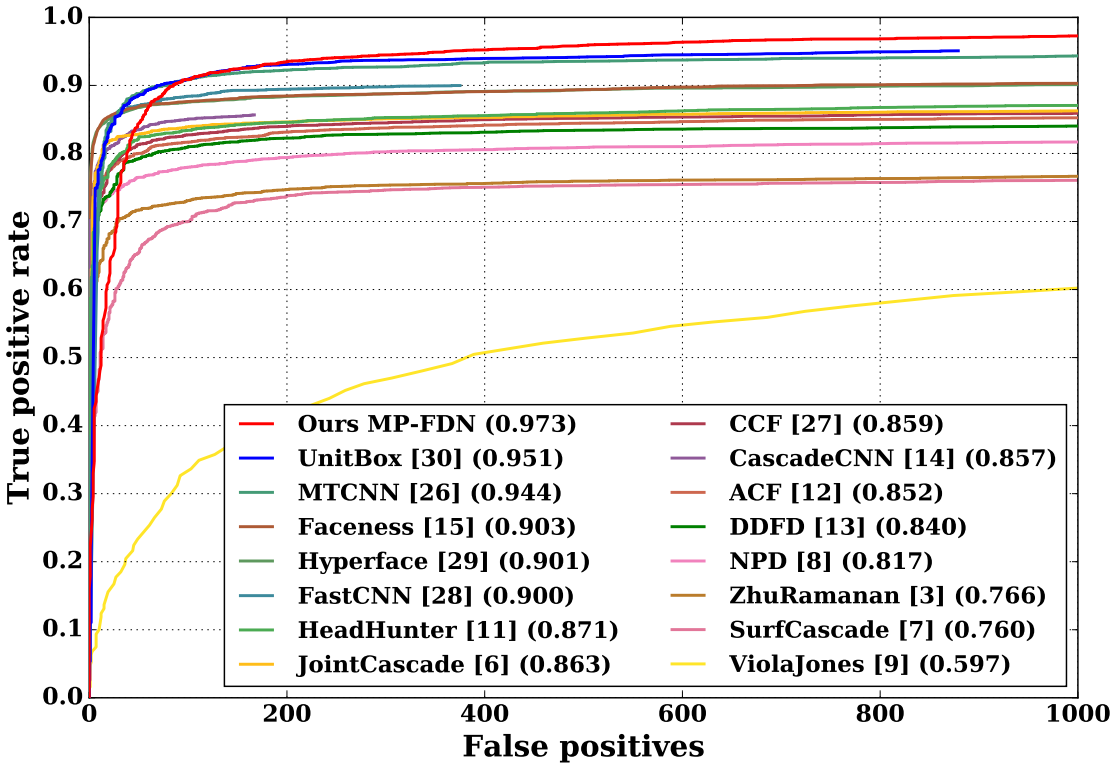


Fig. 5.9 ROC curves of MP-FDN and other published methods on the FDDB dataset [1].

5.4.2 Qualitative Results

Qualitative results for MP-FDN on the FDDB dataset are shown in Figure 5.10. We observe that proposed MP-FDN face detector is robust to variations in scale, pose, occlusion, expression, and out-of-focus blur.

Fig. 5.10 Qualitative results on the FDDB dataset [1].

5.5 Conclusion

This chapter benchmarked the proposed MP-FDN method on two representative face detection datasets, WIDER FACE [4] and FDDB [1]. MP-FDN consistently achieved the highest performance on these datasets. In particular, on the "hard partition" of the WIDER FACE validation and test sets that contain faces of height as small as 10 pixels while as large as more than 1000 pixels, MP-FDN outperforms the previous best method by 12.6% and 9.8%, respectively, in Average Precision. This demonstrates the superior capability of the proposed MP-FDN at detecting faces across a large span of scales. Besides facial scale, the qualitative results on the WIDER FACE and FDDB datasets also illustrate that our algorithm exhibits a high level of robustness to other important factors, including illumination, pose, occlusion, facial expression, and out-of-focus blur.

In addition, we have made a fine-grained analysis of eight facial attributes that may affect the face detection performance of MP-FDN: face bounding box height, face bounding box area, facial height-width ratio, blur, expression, illumination, occlusion and facial pose. We also made a detailed analysis of the hard false positives and hard false negatives. Both the fine-grained facial attribute analysis and hard false negative analysis showed that face scale[15], blur and occlusion affect the face detection performance the most. This shows us directions for further improving face detection performance. However, this is not enough: the analysis of the hard false positives and hard false negatives on WIDER FACE dataset showed that many hard false positives are actually human faces, and many hard false negatives are undetectable even by humans. This result indicates that, not only face detection algorithms, but also face detection benchmarks need to be improved to promote the development of face detection research.

[15]Face scale here refers to both face bounding box height and face bounding box area.

Chapter 6

Conclusion

As stated at the outset, this thesis focuses on answering the following two questions:

Question 1: What is the reason behind the phenomenon that tiny faces cannot be accurately detected by a Convolutional Neural Network (ConvNet)?

Question 2: Is there any way that we can adapt the deep learning framework so as to detect tiny facial patterns with high accuracy?

The series of controlled experiments in Chapter 3 uncovered the *scale sensitivity rule* that directly answers the first question. The scale sensitivity rule can be described as follows: *A low-level convolutional layer (conv-layer) with a small receptive field is most sensitive to small object patterns, while a high-level conv-layer with a large receptive field is most discriminative to large object patterns.* Accordingly, the answer to ***Question 1*** is that a common convolutional neural network simply employs the feature map of its last conv-layer to predict faces. This last conv-layer, with a large receptive field, is not sensitive to small facial patterns, thus leading to a low detection accuracy of these tiny patterns.

Not content at merely finding the answer to ***Question 1***, we further proposed a new ConvNet that leverages the scale sensitivity rule to achieve simultaneous and accurate detection of faces across a large span of scales, thus giving a definite answer "YES" to ***Question 2***. Specifically, we classified all conv-layers in a ConvNet into three groups, namely, "Det-S4", "Det-S8" and "Det-S16"[1], that are sensitive to small, medium and large facial patterns, respectively. For "Det-S4", the conv-layers in this group are combined to form a sub-network that is proficient at proposing small-sized faces (5-11 pixels in height).

[1]See Figure 3.14.

Similarly, the conv-layers in "Det-S8" form a sub-network that is proficient at proposing medium-sized faces (12-128 pixels in height), and the conv-layers in"Det-S16" form a sub-network proficient at proposing large-sized faces (larger than 128 pixels in height). Thus, the three groups form a three-pronged network for proposing faces of three scale ranges. We refer to this as a Multi-Path Face Proposal Network (MP-FPN). MP-FPN can simultaneously predict small, medium and large faces through the three parallel detection branches. To further improve the discriminative power for the hard false positives and hard false negatives generated by MP-FPN, we designed an additional follow-on Multi-Path Face Verification Network (MP-FVN) to verify each face proposal. MP-FVN still adopts the same three paths to deal with small, medium and large face proposals. However, for each such detection path, MP-FVN combines the deep features of each facial region with a larger contextual region to verify the confidence of the face proposal. Furthermore, MP-FPN and MP-FVN share the majority of conv-layers and parameters (see Figure 4.11), thereby composing an end-to-end Multi-Path Face Detection Network (MP-FDN).

We have verified the effectiveness of MP-FDN by employing two large face detection benchmark datasets that together contain over 20,000 test images. On the WIDER FACE [4] validation and test sets, for the so-called "medium partition" and "easy partition" that contain medium- and large-size faces, the proposed MP-FDN matched or even outperformed state-of-the-art methods by a small margin. More importantly, for the so-called "hard partition" that contains mostly tiny faces, it outperforms the previously best result by 12.6% on the validation set, and 9.8% on the test set, in terms of average precision. On the FDDB dataset [1], MP-FDN achieves an average recall of 97.3%, outperforming all the other 15 strong face detection algorithms. These results demonstrate that the proposed MP-FDN is a viable and accurate algorithm for face detection.

Next, we suggest four possible improvements and additions as future work:

(1) In the investigation of the impact of different facial attributes to MP-FDN (see Section 5.3.3), we found that the normalized average precision of tiny faces lagged much behind that of large faces (see Figure 5.5a and Figure 5.5b), implying that there is significant room for improving the detection accuracy of tiny faces. We plan to introduce the super-resolution technique [65] in the tiny face detection path ("Det-S4" in Figure 4.11). Since there already exist deep ConvNets for super-resolution, for example, [65], we can seamlessly combine these super-resolution conv-layers to "Det-S4". Moreover, this modification will not affect other two paths for detecting medium- and large-size faces because all detection

paths are run independently and in parallel.

(2) We also note that occlusion exerts the second largest influence on the face detection performance of MP-FDN (see Figure 5.5g), next to facial scale. We observe that most occluded faces often have occluded contextual information so that MP-FDN cannot leverage contextual features for an effective verification of these faces (see Figure 5.8). As an alternative to the contextual information that was already used in MP-FDN, we plan to introduce data regarding facial parts to alleviate the occlusion problem. Specifically, we plan to create a new training set for this purpose based on MP-FDN. In order to achieve this, we will mask parts of the face and even other body regions (e.g., neck, shoulder and upper body) with random colors for a proportion of the training images. These artificially occluded faces, as well as natural faces, will be used to train the MP-FDN. This will enhance MP-FDN's sensitivity to facial parts and thus probably improve its robustness to facial occlusion.

(3) We note that although the proposed MP-FDN achieved the highest average recall rate (0.973) among all 16 face detection methods on the FDDB dataset (see Figure 5.9), its true positive rate (aka recall rate) is lower than several other methods when the number of false positives are less than about 80 (see Figure 5.9). This implies that MP-FDN generates a small number of high-scoring false positives that even a high threshold cannot separate them from the true positives. We plan to investigate the reason behind the high-scoring false positives in the FDDB dataset and introduce a mechanism to mine these false positive patterns during the training process. This might lead to a MP-FDN that is more robust at detecting these hard false positives.

(4) In the visualization of the hard false positives of WIDER FACE dataset (see Figure 5.7), we found many miss-labeled true faces (taking up 60% of the top-100 false positives). Moreover, in the visualization of the hard false negatives in the WIDER FACE dataset (see Figure 5.8), we found many labeled faces are hardly recognized even by humans (taking up 51% of the top-100 false negatives). These cases of miss-labeling and wrong labeling can possibly mislead any face detection algorithm that is trained or tested on the dataset. Therefore, we make it one of our future tasks to improve the labeling of WIDER FACE dataset[2]. We also observed that the proposed MP-FDN detects “faces” of cartoon charac-

[2]We expect to communicate with the authors of WIDER FACE dataset, and possibly assist them to improve the face labeling, so that this currently largest face detection dataset will become a more effective training and evaluation tool for face detection research.

ters and figure carvings, as well as faces wearing a mask (see Figure 5.7), which apparently are not counted as faces by WIDER FACE dataset. In fact, it is still controversial whether these patterns should be counted as faces [7]. Thus it would seem that a clear-cut definition of "face" in the context of face detection research is necessary.

Last but not least, we believe that the scale sensitivity rule and the multi-pronged parallel proposal & verification network structure embodied in MP-FDN could be very useful to not only face detection, but other related computer vision problems.

References

[1] V. Jain and E. Learned-Miller, "Fddb: A benchmark for face detection in unconstrained settings," University of Massachusetts, Amherst, Tech. Rep. UM-CS-2010-009, 2010.

[2] J. Yan, X. Zhang, Z. Lei, and S. Z. Li, "Face detection by structural models," *Image and Vision Computing*, vol. 32, no. 10, pp. 790–799, 2014.

[3] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark localization in the wild," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2879–2886.

[4] S. Yang, P. Luo, C. C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *European Conference on Computer Vision*. Springer, 2014, pp. 391–405.

[6] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[7] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool, "Face detection without bells and whistles," in *European Conference on Computer Vision*. Springer, 2014, pp. 720–735.

[8] B. Yang, J. Yan, Z. Lei, and S. Z. Li, "Aggregate channel features for multi-view face detection," in *Biometrics (IJCB), 2014 IEEE International Joint Conference on*. IEEE, 2014, pp. 1–8.

[9] 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*, 2015, pp. 3676–3684.

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[11] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[12] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[13] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[14] 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*, 2014, pp. 580–587.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[16] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1440–1448.

[17] S. Bell, C. Lawrence Zitnick, K. Bala, and R. Girshick, "Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2874–2883.

[18] S. Zagoruyko, A. Lerer, T.-Y. Lin, P. O. Pinheiro, S. Gross, S. Chintala, and P. Dollár, "A multipath network for object detection," *arXiv preprint arXiv:1604.02135*, 2016.

[19] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2147–2154.

[20] 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*, 2015, pp. 91–99.

[21] K.-H. Kim, S. Hong, B. Roh, Y. Cheon, and M. Park, "Pvanet: Deep but lightweight neural networks for real-time object detection," *arXiv preprint arXiv:1608.08021*, 2016.

[22] Y. Li, K. He, J. Sun *et al.*, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 379–387.

[23] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A unified multi-scale deep convolutional neural network for fast object detection," in *European Conference on Computer Vision*. Springer, 2016, pp. 354–370.

[24] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," *arXiv preprint arXiv:1612.03144*, 2016.

[25] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[26] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European Conference on Computer Vision*. Springer, 2016, pp. 21–37.

[27] P. Dollár, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1532–1545, 2014.

[28] 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.

[29] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[30] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[31] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*. Springer, 2014, pp. 740–755.

[32] H. Jiang and E. Learned-Miller, "Face detection with the faster r-cnn," *arXiv preprint arXiv:1606.03473*, 2016.

[33] W. Shang, K. Sohn, D. Almeida, and H. Lee, "Understanding and improving convolutional neural networks via concatenated rectified linear units," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2016.

[34] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1–9.

[35] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *arXiv preprint arXiv:1606.00915*, 2016.

[36] S. Liao, A. K. Jain, and S. Z. Li, "A fast and accurate unconstrained face detector," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 211–223, 2016.

[37] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun, "Joint cascade face detection and alignment," in *European Conference on Computer Vision*. Springer, 2014, pp. 109–122.

[38] S. S. Farfade, M. J. Saberian, and L.-J. Li, "Multi-view face detection using deep convolutional neural networks," in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*. ACM, 2015, pp. 643–650.

[39] 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*, 2015, pp. 5325–5334.

[40] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint face detection and alignment using multitask cascaded convolutional networks," *IEEE Signal Processing Letters*, vol. 23, no. 10, pp. 1499–1503, 2016.

[41] R. Ranjan, V. M. Patel, and R. Chellappa, "Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition," *arXiv preprint arXiv:1603.01249*, 2016.

[42] D. Chen, G. Hua, F. Wen, and J. Sun, "Supervised transformer network for efficient face detection," in *European Conference on Computer Vision*. Springer, 2016, pp. 122–138.

[43] C. Zhu, Y. Zheng, K. Luu, and M. Savvides, "Cms-rcnn: contextual multi-scale region-based cnn for unconstrained face detection," *arXiv preprint arXiv:1606.05413*, 2016.

[44] J. Li, T. Wang, and Y. Zhang, "Face detection using surf cascade," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2183–2190.

[45] P. Dollár, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," 2009.

[46] H. A. Rowley, S. Baluja, and T. Kanade, "Rotation invariant neural network-based face detection," in *Computer Vision and Pattern Recognition, 1998. Proceedings. 1998 IEEE Computer Society Conference on.* IEEE, 1998, pp. 38–44.

[47] H. Schneiderman and T. Kanade, "Object detection using the statistics of parts," *International Journal of Computer Vision*, vol. 56, no. 3, pp. 151–177, 2004.

[48] http://mplab.ucsd.edu, "The MPLab GENKI Database, GENKI-SZSL Subset."

[49] B. Yang, J. Yan, Z. Lei, and S. Z. Li, "Fine-grained evaluation on face detection in the wild," in *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, vol. 1. IEEE, 2015, pp. 1–7.

[50] B. F. Klare, B. Klein, E. Taborsky, A. Blanton, J. Cheney, K. Allen, P. Grother, A. Mah, and A. K. Jain, "Pushing the frontiers of unconstrained face detection and recognition: Iarpa janus benchmark a," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1931–1939.

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

[52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[53] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on.* IEEE, 2009, pp. 248–255.

[54] W. Luo, Y. Li, R. Urtasun, and R. Zemel, "Understanding the effective receptive field in deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 4898–4906.

[55] W. Liu, A. Rabinovich, and A. C. Berg, "Parsenet: Looking wider to see better," *arXiv preprint arXiv:1506.04579*, 2015.

[56] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia.* ACM, 2014, pp. 675–678.

[57] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[58] P. Sinha and A. Torralba, "Detecting faces in impoverished images," *Journal of Vision*, vol. 2, no. 7, pp. 601–601, 2002.

[59] L. Zhang, L. Lin, X. Liang, and K. He, "Is faster r-cnn doing well for pedestrian detection?" in *European Conference on Computer Vision*. Springer, 2016, pp. 443–457.

[60] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 761–769.

[61] D. Hoiem, Y. Chodpathumwan, and Q. Dai, "Diagnosing error in object detectors," *Computer Vision–ECCV 2012*, pp. 340–353, 2012.

[62] B. Yang, J. Yan, Z. Lei, and S. Z. Li, "Convolutional channel features," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 82–90.

[63] D. Triantafyllidou and A. Tefas, "A fast deep convolutional neural network for face detection in big visual data," in *INNS Conference on Big Data*. Springer, 2016, pp. 61–70.

[64] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang, "Unitbox: An advanced object detection network," in *Proceedings of the 2016 ACM on Multimedia Conference*. ACM, 2016, pp. 516–520.

[65] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2016.