INDIAN INSTITUTE OF TECHNOLOGY HYDERABAD

भारतीय प्रौद्योगिकी संस्थान हैदराबाद
Indian Institute of Technology Hyderabad

Partial Face Detection using Regions with Convolutional Neural Networks

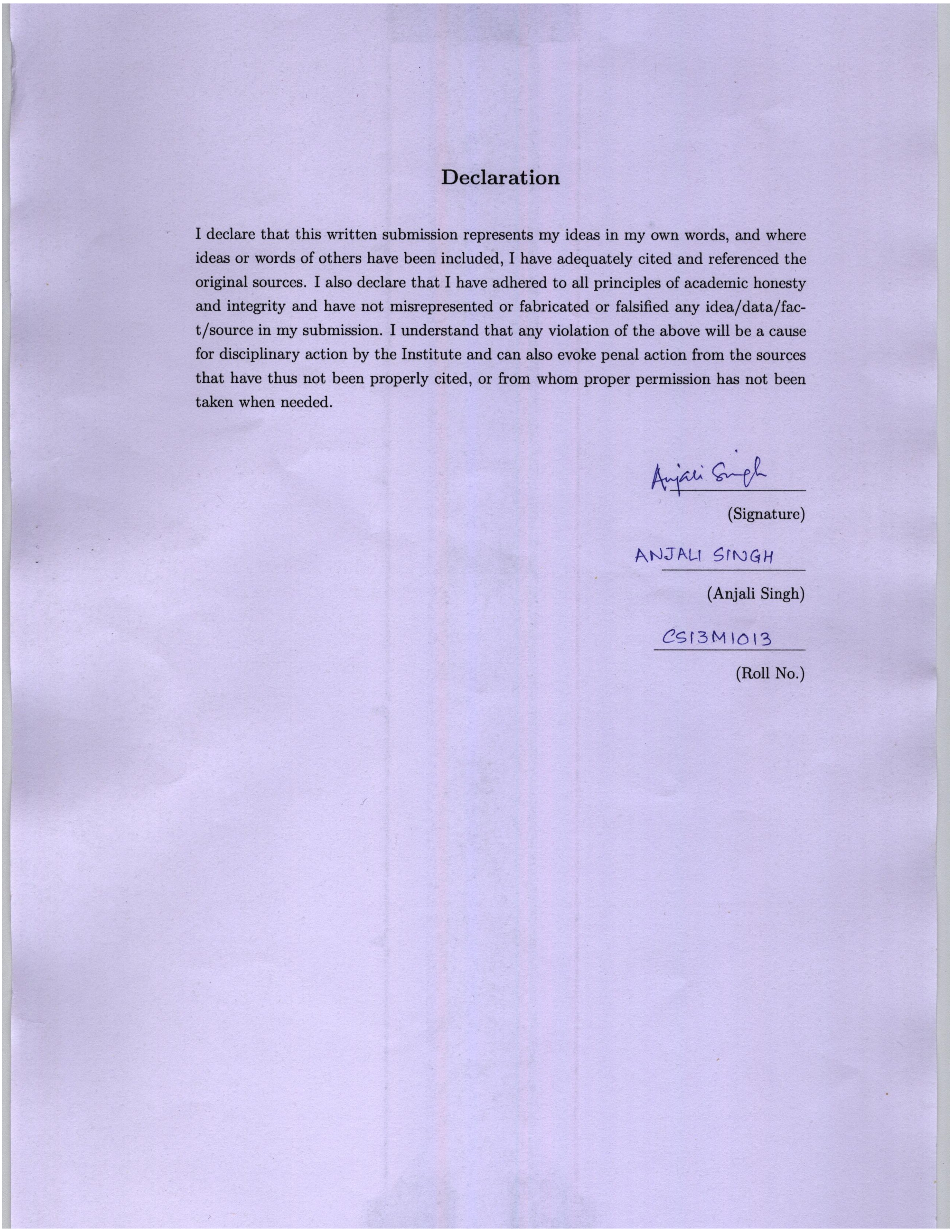
by

Anjali Singh

A thesis submitted in partial fulfilment for the
degree of Master in Technology

in the
Department of Computer Science and Engineering
Indian Institute of Technology Hyderabad

July 2015



Declaration

I declare that this written submission represents my ideas in my own words, and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

Anjali Singh

(Signature)

ANJALI SINGH

(Anjali Singh)

CS13M1013

(Roll No.)

Approval Sheet

This Thesis entitled Region-Face: A Region Based Convolutional Neural Network Approach to Partial Face Detection by Kiran Bhos is approved for the degree of Master of Technology from IIT Hyderabad

C. Krishna Mohan
DR. C. KRISHNA MOHAN
(———) Examiner
Dept. of CSE
IITH

DR. K. SRI. RAMA MURTY
(———) Examiner
Dept. EE
IITH

V.N.B Vineeth
DR. VINEETH N. BALASUBRAMANIAN
(———) Adviser
Dept. of CSE
IITH

~~(———) Co-Adviser~~
~~Dept. of ....~~
IITH

DR. NAVEEN SIVADASAN
(———) Chairman
Dept. of CSE
IITH

“Imagination is the highest form of research.”

Albert Einstein

Abstract

Although many methods have been developed for holistic face detection, detecting partial faces hasn't been a successful endeavour yet. Partial faces frequently appear in real world environments like in surveillance videos and are quite difficult to detect. Recently, CNNs have shown very promising results with object detection in PASCAL VOC challenges. We propose an approach to detect partial faces along with holistic faces (frontal and profile views) present in natural scenarios. In the proposed method, we use CNN for feature extraction and representation. The drawback of CNN being computationally expensive is dealt with Selective Search using Segmentation, which reduces the search space to a great extent. We used FDDB Benchmarking for evaluating our method and the results were near the best results of all recent methods in the face detection domain.

Acknowledgements

I would like to thank my project advisor Dr. Vineeth N. Balasubramanian, for his continuous support and encouragement, and Mr. Adepu Ravi Shankar for the valuable discussions. . .

Contents

Dedicated to my parents. . .

Chapter 1

Introduction

Face detection has advanced significantly in the last a few decades, however detection of partial faces has not been dealt with much attention. Most of the face detectors work well with images having only frontal view or profile view of faces and struggle to detect faces which are only partially visible. For feature extraction and representation, although HOG and SIFT have conventionally been used extensively in detection, they seem to be bottleneck in further performance improvement as we move towards deep learning architectures. Features learned by Neural Networks are much more promising as seen in the recent PASCAL VOC Challenge, 2010.

1.1 Face Detection

Face Detection is the first step of automatic face recognition, which is important and well explored due to its significance in applications like access control or video surveillance. A lot of variations in appearance of a face occur in images due to orientation, pose variation, occlusion, illuminating condition and facial expression making face detection a difficult task.

1.1.1 Partial Faces

From surveillance cameras capturing involuntary images to casual personal photography, we constantly generate images in which human faces appear. Automatic face tagging is expected to be as accurate as manual tagging, so that further usage of the data may be possible. Specially in public places, where surveillance is important for management and security, it will be very useful if there is a real-time face detection and recognition system. Due to the ad hoc nature of such images, faces are not always captured in frontal pose(See Figure 1.1). Many times the faces are hidden behind occlusions or facial/head accessories and/or are tilted. This leads to very poor performance of face detection methods. In this thesis we propose a method to detect faces in such real world scenarios.

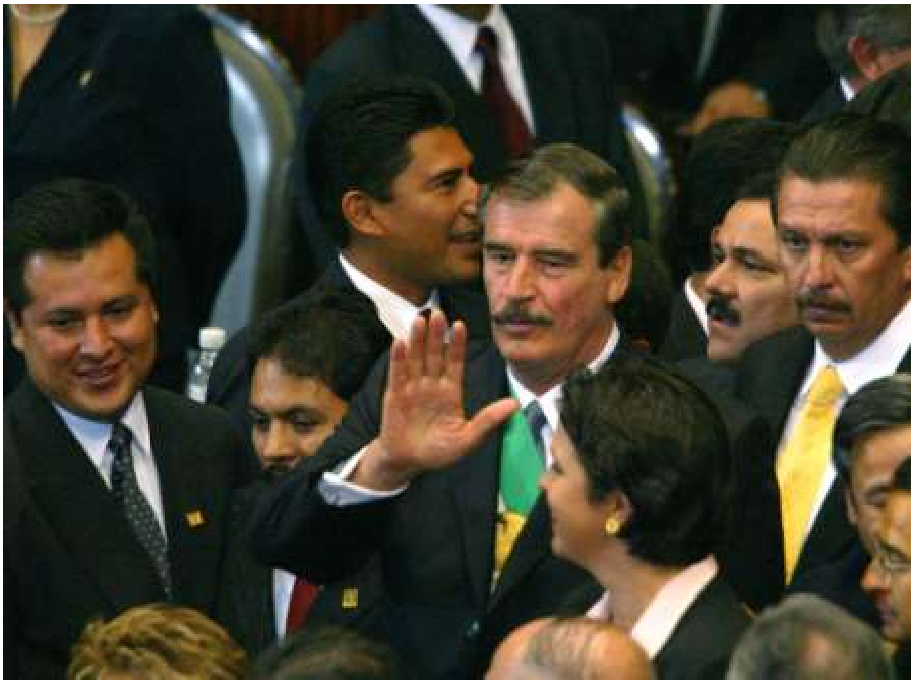


FIGURE 1.1: Partial faces occurring in real world scenario

1.2 Deep learning and CNN

Until recent past, most of in-practice classification and detection techniques were dependent on handcrafted features like HOG and SIFT. However, they seem to be bottlenecks for improvement now. With advent of use of deep learning and CNN in classification[1] and object detection[2] and their promising results, they appear to be a good choice for face detection. Deep learning is a technique using which the system can define feature descriptors on its own using input training data. It helps the network learn descriptors based on underlying patterns in the data. One drawback of deep learning architectures is that they are computationally very expensive, which has hindered their use in detection problems. Because a face can occur at any location in an image and can be of any scale, trying to detect a face at each of those possible locations exhaustively is not feasible.

1.2.1 Deep Learning

Deep learning is a branch of machine learning which aims at learning features as well as feature representation of data. Data can be represented in many ways, say for an instance, an image can be represented as vector of pixel values, or set of edges and intensity gradients. It is easy to understand that some representations may make the task of learning easier. However, hand-crafting such a representation is a difficult challenge due to large scale and diverse nature of data. Deep learning aims at replacing hand-crafted feature representation with efficient algorithms for feature learning and hierarchical feature extraction.

It intends to generate better representations for underlying features in data and creating models which can learn such representations from large-scale unlabeled data. Some of the approaches in deep learning are inspired by the human nervous system and are loosely based on interpretation of how the brain works via stimulus and neuronal responses and the information processing and communication patterns.

There are many deep learning architectures such as deep neural networks, convolutional neural networks(CNNs), deep belief networks and recurrent neural networks.

1.2.2 Convolutional Neural Networks

Traditionally, in learning algorithms, hand-crafted feature extractor collects relevant information from the input data, which may be raw or may come after going through certain pre-processing like contrast enhancement, geometric transformations, brightness interpolation, etc. The features generated from the extractor is used to train a classifier, which can later be used to classify new unseen data. Instead of having a hand-designed feature extractor, Neural Networks can learn an appropriate feature extractor from the data itself. Feature representation is also taken care of by the learned Neural Network model. However, Neural Networks in their crude form are not very promising for images as input data. An image can be typically of several hundred variables(pixels) which makes the input layer extensive. Further, multiple fully connected layers of the Neural Networks make the learning process expensive in terms of time and storage, as there will be millions of variables(weights) to be learnt. Also, Neural Networks do not have built-in invariance towards translation or local distortion in the input, whereas in real world images, such distortions are very frequent.

Convolutional Neural Networks, CNNs, obtain shift invariance as weights are shared across space, in other words, as a kernel is applied across all of the receptive fields(regions) in the image. Images have strong local structures which is naively ignored by Neural

Networks, whereas CNNs force the extraction of local features by introducing receptive fields in hidden layer units which work locally. Local receptive fields, shared weights and spatial sub-sampling enable CNNs to achieve some degree of invariance towards shift and distortion.

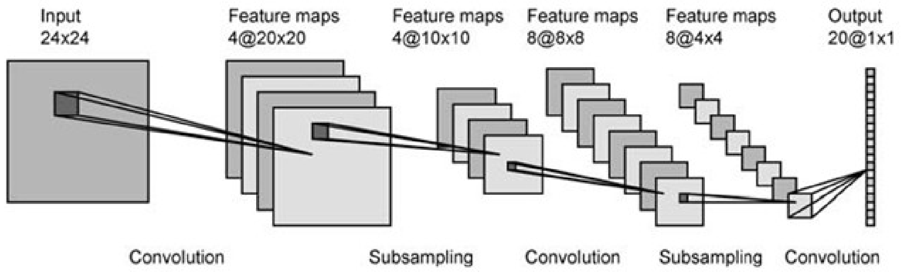


Figure 1.2: Convolutional Neural Network for image processing

Images are size-normalized and centered before being sent as input to the input layer of a CNN. Each pixel is considered as a neuron unit. Each unit in a layer receives input from a collection of units form a small neighborhood in the previous layer. With such locally working receptive fields, the CNN can extract elementary visual features such as edges and their orientation, end points or corners. These features are then combined by the succeeding layers,to get features at more abstract level. A set of neurons in a layer have identical weight vectors and have their receptive fields located at different places in the image. There are usually multiple such sets of neurons in each layer. Outputs of such a set of neurons makes a feature map. Implementation wise, a single neuron with the weights which are supposed to be shared, can be used to scan the input image and store the states at corresponding locations in a feature map. To do it in parallel a feature map can be implemented as a plane of neurons that share a weight vector, and its units perform the same operation on different regions in the image.

A convolution layer is a collection of several feature extractor which extract multiple features using feature maps at each location, each representing a certain type of learned feature. Once a feature has been extracted, the absolute location of the feature is not important as long as it relative position with other features is preserved. Therefore a subsampling layer reduces the resolution of the output feature map for the next convolution layer. CNNs usually have alternating convolution and sub-sampling layers, where after every layer the number of feature maps is increased and resolution is reduced. Each unit in the second convolution layer in Figure 1.2, may have input connections from multiple feature maps from the previous layer.

The weights in CNN are learned using an algorithm called back-propagation.Considering the way CNN uses these weights, it can been seen as learning kernels for convolution, which are used for feature extraction. Sharing of these kernel weights lead to reduction in number of variables making CNNs less expensive in terms of time and storage.

1.3 Selective Search and Segmentation

Inspite of being powerful in learning features and detection, CNNs have a drawback of being computationally expensive. Thus, exhaustive search in all possible locations in the image does not look very promising. In our method we try to reduce the search space using a technique called Segmentation.

1.3.1 Selective Search

The first step in the process of detection starts with finding out all possible locations for the object to be present in the image, which could be anywhere throughout the image and can be at any scale. This has, traditionally, been done using the sliding window approach which scans the image and returns windows of different scales throughout the image. Number of such windows is typically in millions, which has hindered the use CNNs in real-time situations.

This search space of millions of candidate windows can be reduced based on certain heuristics. Our method uses selective search to reduce the number of candidate windows for the main detection step which is done by the CNN.

1.3.2 Segmentation

Uijling et al[3] proposed Segmentation as an approach to do selective search. Starting with a very fine segmentation(pixel level), similar neighboring regions are combined to get higher level segments. Similarity of neighboring regions is calculated based on their sizes, textures and intensity values.

To obtain diversification and ensure that all complementary segments are obtained, segmentation is performed on different color channels having different invariance properties. These color spaces have different sensitivities to shadows, highlights and edges. For example, standard RGB is most sensitive to these variations, whereas normalized RGB is not sensitive to shadow and shading. Hue is most invariant of all. All the segments obtained, throughout the hierarchy are considered as potential object locations. However, number of locations still remain in between 1000-10000, instead of millions when sliding window is used.

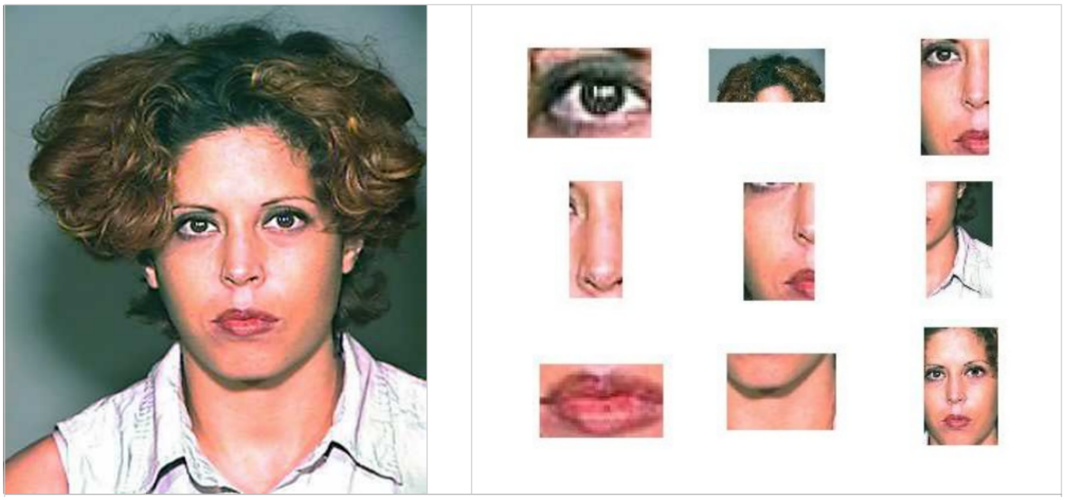


FIGURE 1.3: Left: Input Image, Right: Some of candidate windows generated by Segmentation. We can see how parts of face are also segmented out separately.

Segmentation is particularly useful for partial face detection because it can segment out parts of faces, as seen in Figure 1.3. Regions having similar intensity, texture and gradients in multiple color spaces are combined in hierarchical way to get segments at various scales. Hence, we get parts of face like an eye or a nose, side of a face or jawline. Such segments are more informative than random windows at various scales and are therefore more promising for our problem.

Chapter 2

Related Work

2.1 Previous Face Detection approaches

Finding faces in images with mono-colored background, or detection using color or motion were some of earlier methods which attempted to detect faces in very constrained settings. In practice, such constraints are hardly met. Later, approaches like model based face tracking were used for unconstrained scenes. In 2001, Froba et al.[4] used edge orientations to create a model describing a face, whereas Jesorsky et al.[5] used Hausdorff distance as a metric, and created a model using facial feature points. In 2004, Viola and Jones [6] presented a seminal paper where they used a cascade of weak classifiers and simple Haar features for face detection, along with Integral images which made the detection real-time. After excessive training it yields very impressive results.

Neural Networks were introduced in the domain of face detection as early as late 1990s.During that time Rowley, Baluja and Kanade ([7], [8]) introduced simple neural networks which looked into small regions of an image and decided if the region contained a face or not. Images were usually pre-processed with lighting correction, histogram equalization and alignment correction. They also proposed a rotation invariant method which predicted rotation along with detection of a face in the image. Jeffrey S. Norris used Principal Component Analysis and Neural Networks(PCA and ANN) in Face Detection and Recognition in Office Environments. During early 2000s, Hazem E-Bakry [9] proposed to reduce computation time for finding faces in images using fast neural networks(FNN). Lin Lin Huang et al. proposed face detection using polynomial neural networks(PNN) which classified regions in images generated by sliding window approach into a face or a non-face. PNN takes binomials of projections on PCA learned feature subspace. In 2005, Marian Beszedes & Milos Oravec [10] presented an approach for localizing human faces in images using multilayer perceptrons(MLP). After pre-processings like normalization, rotational changes and light conditions improvement, small windows from the input image were given to MLP, which detected rotation of input window and also decided if it contained a face or not.The following year, Zoran and Samcovic [11], proposed a method for face detection in surveillance videos using Neural Networks. The method used three representations, namely pixel, partial profile and eigenfaces for faces. These three representations were used by three independent sub-detectors to achieve final result.

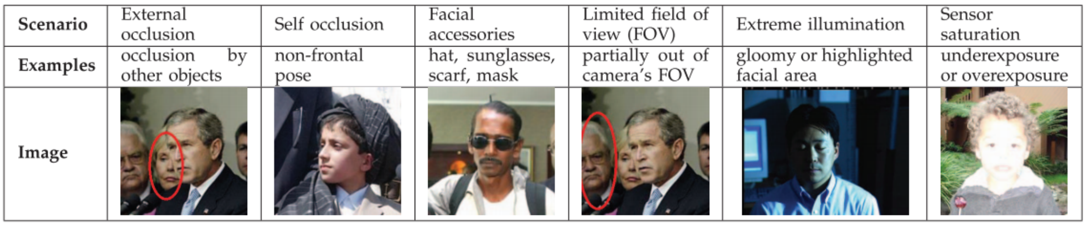
In 2003, Masakazu Matsugu [12] proposed an algorithm for facial expression recognition which also served the purpose face detection using a convolutional neural network(CNN).

Chapter 3

Proposed Methodology: Selective Search with Convolutional Neural Networks for Partial Face Detection

3.1 Partial Faces

Our method mainly tries to improve face detection by focusing on partial faces along with full frontal faces and profile view faces. Partial faces may occur in images due to reasons like occlusion by other objects, non-frontal pose, facial accessories, extreme illumination, shadows etc (Figure 3.1). Detection of such partial faces is significant in situations when images are captured without user cooperation, like in surveillance scenarios or through hand-held devices as they may not capture full frontal views of a face.



Scenario	External occlusion	Self occlusion	Facial accessories	Limited field of view (FOV)	Extreme illumination	Sensor saturation
Examples	occlusion by other objects	non-frontal pose	hat, sunglasses, scarf, mask	partially out of camera's FOV	gloomy or highlighted facial area	underexposure or overexposure
Image						

Figure 3.1: Partial faces occurring in images of real world scenarios (Liao et al. 2013)

3.2 Detecting faces and partial faces

For our detection problem, we use a similar approach as in R-CNN[2] which was used for Object Classification. In this approach we first generate region proposals from the input image using Segmentation, which is treated as the candidate windows. We use a CNN with similar architecture as in [1] to extract features from these windows. These features are then scored using pre-trained binary SVM to detect if a window contains a face or not.

3.2.1 Generating Region Proposals

To generate region proposals we use Segmentation[3]. Following is a brief overview of the algorithm:

1. Start with an over-segmentation. Usually each pixel is treated as a separate region.
2. Iteratively, two most similar regions are merged until the whole image becomes a single region. The similarity S between two regions say a and b is defined as: $S(a, b) = size - similarity(a, b) + texture - similarity(a, b)$

   Both of the similarity functions size-similarity() and texture-similarity() return a value in [0,1].

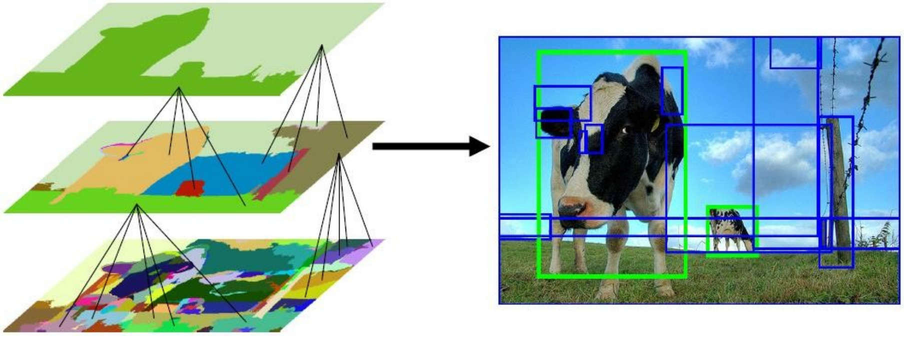


Figure 3.2: Hierarchical Segmentation, in which most similar regions are combined as we go up the hierarchy. This gives us windows at various scales(Right)[3]

Regions at each level of the hierarchy can be considered as potential object locations, with some limit set on minimum size of a segment. In our experiments we set minimum size of candidate windows to be 20x20 pixels.Also, we consider tight bounding boxes around the segment region as a candidate window.

To cover maximum possible locations in the proposals, the process is performed on variety of colour channels with different sensitivities to changes in lighting. For example, standard RGB is most sensitive to these variation, whereas normalized RGB is not sensitive to shadow and shading. Hue is most invariant of all. In our experiments we use 'Hsv (Hue, Saturation, Value)', 'Lab', 'RGI', 'H', 'Intensity' colour spaces.

3.2.2 Extracting Features

Each of the potential location windows(Regions) are then forwarded to a CNN, which generate features for them. The CNN we use has a similar architecture as in [1]. A pre-trained CNN model is finetuned using a dataset suited for the purpose, in our case, face dataset.

Each candidate window is warped to the size of 227x227 pixels and forward propagated through 5 convolutional layers and two fully connected layers to generate a 4096 dimensional feature vector representing the window.

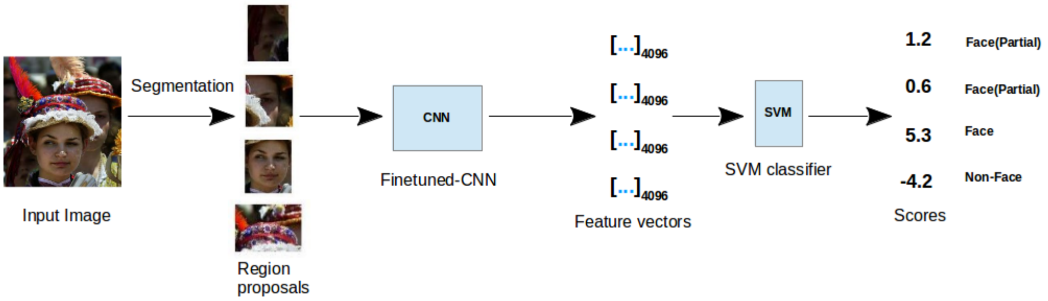


FIGURE 3.3: Pipeline to detect faces in an image.

3.2.3 Scoring Regions

A binary SVM which has been trained using the feature vectors generated from the same CNN using training data. The 4096 dimensional feature vector obtained from the CNN is used as input to find the score for the corresponding region using the pre-trained SVM.

Once scores of each region is calculated, we apply non-maximum suppression(NMS). This reduces the number final detected boxes by removing overlapping detections of a same face. In NMS, we greedily select high-scoring detections and skip detections that are significantly covered by a previously selected detection.We set the threshold to be 30%. So, once the highest scoring region is selected, all regions having overlap of greater than equal to 30% are removed.

Chapter 4

Experiments

4.1 Experimental Setup

We used R-CNN's code and Caffe framework for performing our experiments. FDDB was the main dataset used to evaluate the performance.

4.1.1 RCNN

"R-CNN[2] is a state-of-the-art visual object detection system that combines bottom-up region proposals with rich features computed by a convolutional neural network. At the time of its release, R-CNN improved the previous best detection performance on PASCAL VOC 2012 by 30% relative, going from 40.9% to 53.3% mean average precision. Unlike the previous best results, R-CNN achieves this performance without using contextual re-scoring or an ensemble of feature types." - rbgirshick/rcnn:GitHub[2].

It has mainly two modules, Region proposals and Feature extraction. Region proposals deals with generating category-independent region proposals using Segmentation[3]. It follows the approach as explained in Section 1.3.2. This also helps in enabling a controlled comparison with prior detection work. Feature extraction, it extracts a 4096-dimensional feature vector from each region proposal using the Caffe(Section 3.2) implementation of the CNN described by Krizhevsky et al. [1]. The CNN takes 227x227 mean subtracted RGB image as input. It is forward propagated through five convolutional layers and two fully connected layers to compute the features. These features are then scored using pre-trained class-specific SVMs, to classify the regions. Non-Maximal Suprression(NMS) is applied to remove overlapping bounding boxes.

4.1.2 Caffe

Caffe [13] is a deep learning framework developed by the Berkeley Vision and Learning Center (BVLC) and by community contributors, which has enabled us to integrate CNN training and finetuning and our code very easily.

Prerequisites of Caffe include: CUDA(required for GPU mode), BLAS via ATLAS, MKL, or OpenBLAS, Boost >= 1.55, OpenCV >= 2.4, protobuf, glog, gflags, and IO libraries like hdf5, leveldb, snappy, lmdb.

Caffe is written in C++ and require mex to interact with MATLAB. It comes with a MATLAB wrapper which helps in seamless integration of codes.

4.1.3 FDDB

FDDB [14], is a data set of face regions designed for studying the problem of unconstrained face detection. This data set contains the annotations for 5171 faces in a set of 2845 images taken from the Faces in the Wild data set. A wide range of naturally occurring hindrances towards face detection are present in the images including occlusions, varying poses, and poor quality capture of faces like out-of-focus and low resolution.

The face regions are specified as elliptical regions. For our experiments, we converted elliptical regions into rectangular boxes by taking the smallest rectangle covering the ellipses.

FDDB also comes with an evaluation code of its own, which can be used for benchmarking the performances of various methods in face detection. We have used their comparison code to generate the performance graphs.

4.1.4 Finetuning CNN

We used CNN pre-trained on a large auxiliary dataset (ILSVRC 2012) with image-level annotations (i.e., no bounding box labels) as done by [2]. The Pre-training was performed using the open source Caffe CNN library [13]. To adapt the CNN to the new task of detection and completely new domain of faces, we finetuned the pre-trained CNN using FDDB dataset. Following FDDB's 10-fold cross validation, we finetuned the CNN leaving out the test fold each time.

4.2 Experiments and Results

4.2.1 Experiment 1: Regions proposed by Segmentation

Ideally, we would like to get all possible locations for detecting a face, which when provided by sliding window method comes in millions. To reduce the large number, we employ Segmentation as in [3] and get a little over 2000 candidate windows. Minimum size of the windows is restricted to 20x20 pixels. To ensure that these windows covered most of the possible location of faces, we tested the output with one fold using IOU overlaps with the ground truth annotations. We found out that, with 30% IOU overlap, about 90% of all the faces in ground truth annotations were proposed as candidate windows. This seems to be a fairly good number, as the windows can later be adjusted using techniques like bounding box regression.

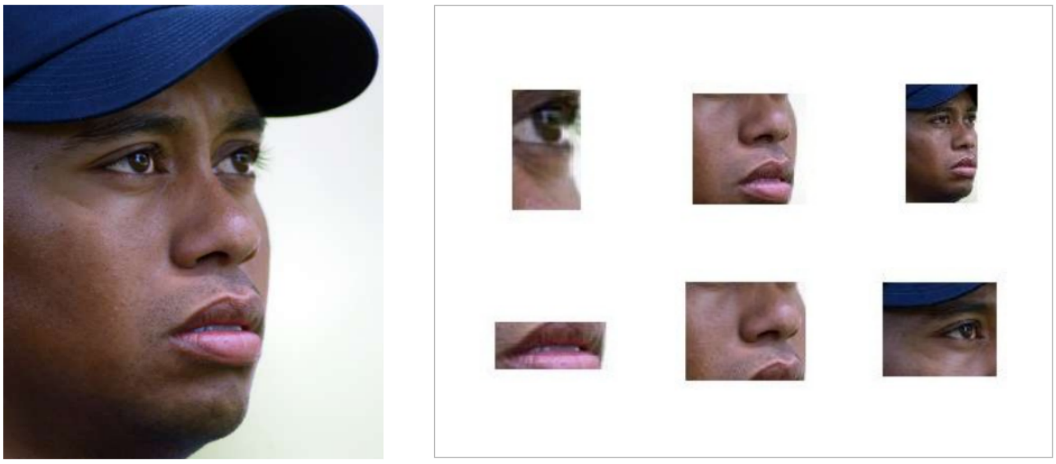


Figure 4.1: Left: Input Image(Side pose), Right: Some of candidate windows generated by Segmentation.

We also conducted experiments to check which kinds of segments were generated for faces. We found out that, apart from full face candidate windows, parts of faces were also generated as candidate windows. This is particularly helpful in case of partial faces, as local features of a face are captured in these windows. As seen in Figure 1.4, 3.1 and 3.2, parts of face like eyes, nose, lips, forehead, jawline, cheeks were segmented in separate windows. Such windows can be used to improve the performance further.

4.2.2 Experiment 2: Partial Face Detections

To find out how well our method performs with partial faces we compared its performance against state-of-the-art Viola Jones method. We considered all faces which were not holistic or in frontal pose as partial faces. We used Viola Jones method of MATLAB's

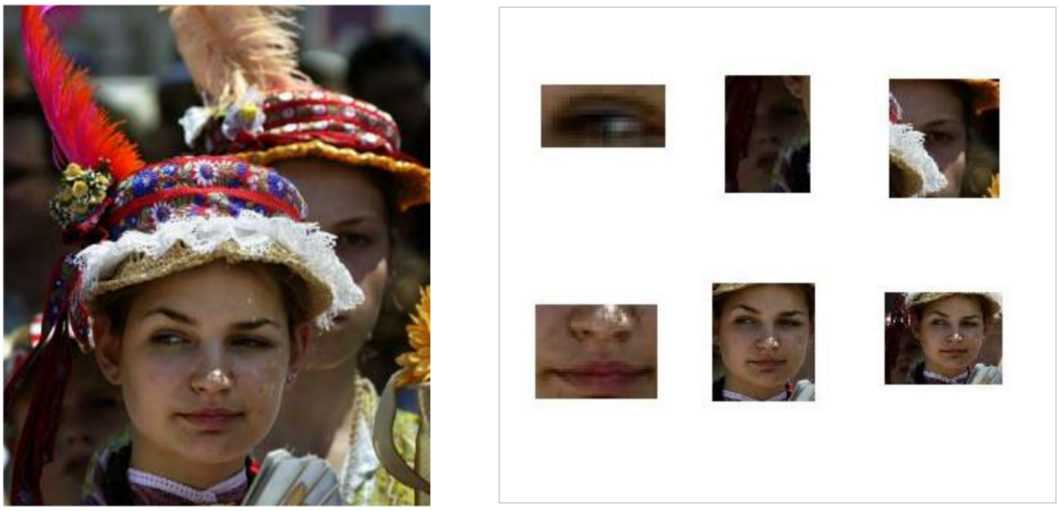


Figure 4.2: Left: Input Image(with Partial Faces), Right: Some of candidate windows generated by Segmentation.

vision package (CascadeObjectDetector System object, default:FrontalFaceCART) to detect faces in one of the folds. Then, we manually checked the number of partial faces detected by the said Viola Jones implementation, and compared it against our method. There were 276 images in the fold. We found out that number of partial faces detected by Viola Jones method was 22, whereas number of partial faces detected by our method was 97. Figures 4.3 and 4.4, and 4.5 and 4.6 show some of the outputs from the experiment.

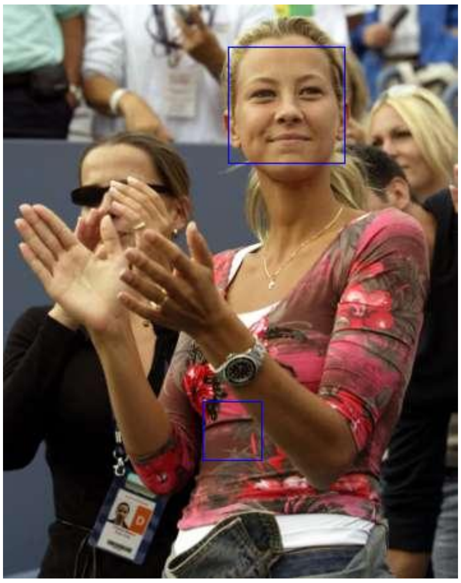


Figure 4.3: Detection of faces using Viola Jones method: Blue box represents the face detected.

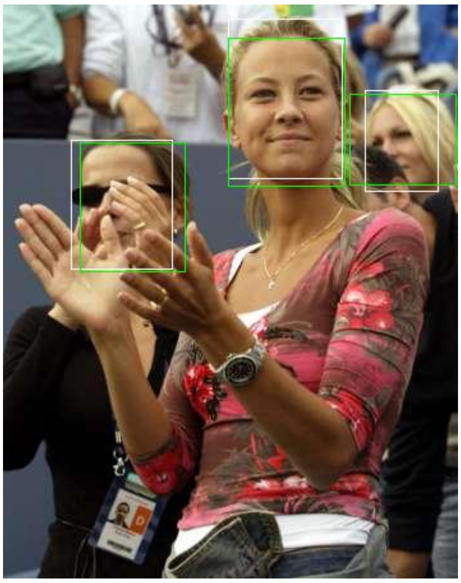


Figure 4.4: Detection of faces using our method: Green boxes represent the faces detected.

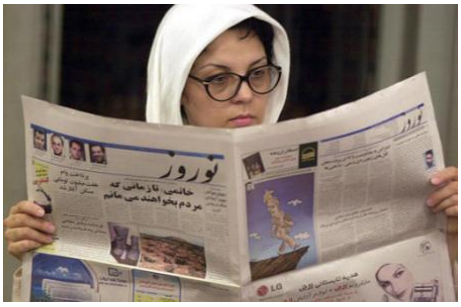


FIGURE 4.5: Detection of faces using Viola Jones method: Blue box represents a face detected.

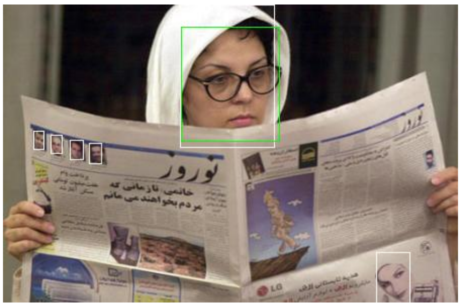


FIGURE 4.6: Detection of faces using our method: Green box represents a face detected.

4.2.3 Experiment 3: Face detection evaluation with FDDB

FDDB [14] has an evaluation kit which can match detections and annotations and compute the resulting scores. We followed FDDB's 10-fold cross validation and used each fold(in rotation) as a testing fold and trained our R-CNN model with remaining 9 folds.

For evaluation, FDDB has some assumptions like:

- A detection means a continuous image region.
- Any processing required to remove overlapping detections has already been done.
- Each detection bounding box corresponds to exactly one face.

The degree of match between a detection d_i and ground truth region l_j, is evaluated based on commonly used Intersection over Union(IOU) ratio, $S(d_i, l_j)$.

To address the problem of large number of false positives or multiple overlapping detections, it formulates the problem of matching annotations and detections as finding a maximum weighted matching in a bipartite graph. A graph G with the set of nodes $V = L \cup D$ is constructed using detections and annotations(ground truths) as nodes. Each detection node $d_i \in D$ is connected to each ground truth node $l_j \in L$ having the edge weight w_{ij} as the score computed IOU ratio. For each detection $d_i \in D$, a node n_i is added to take care of the case when the detection d_i has no overlapping face region(False Positive) in L.

The desired matching M, will be a selection of a set of edges $M \subseteq E$, that maximizes the cumulative matching score, while satisfying the condition that each detection node is connected to at most only one ground truth node, and every ground truth node is connected only one detection node.

Evaluation metrics: Two metrics are used to specify the score y_i for a detection:

- Discrete score(DS): $y_i = \delta_{S(d_i,l_i)>0.5}$
- Continuous score(CS): $y_i = S(d_i, v_i)$

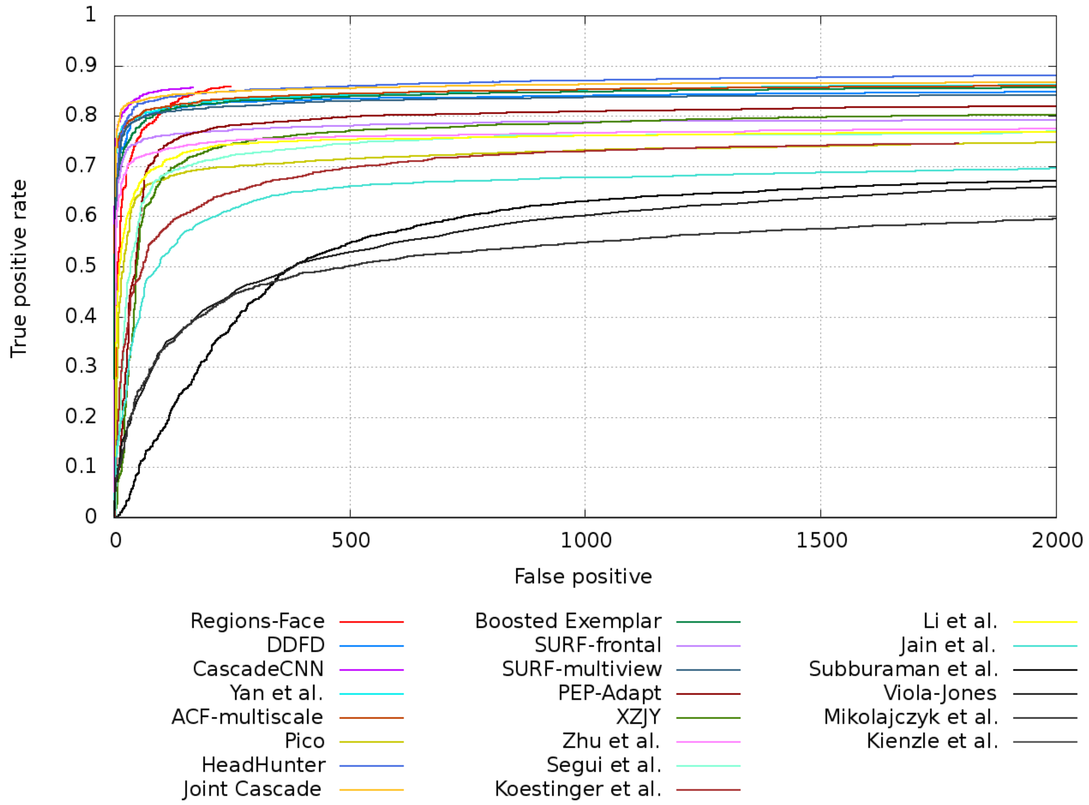


Figure 4.7: ROC Curves for Discrete scores of various methods, Regions-Face(red) represents performance of our method

Discrete Score(DS) metric uses a function to map scores of IOU ratio using a δ function against a threshold, whereas Continuous Scores uses the IOU score itself for evaluation. We use both metrics for evaluation, and get results as shown in Figure 4.7 and 4.8.

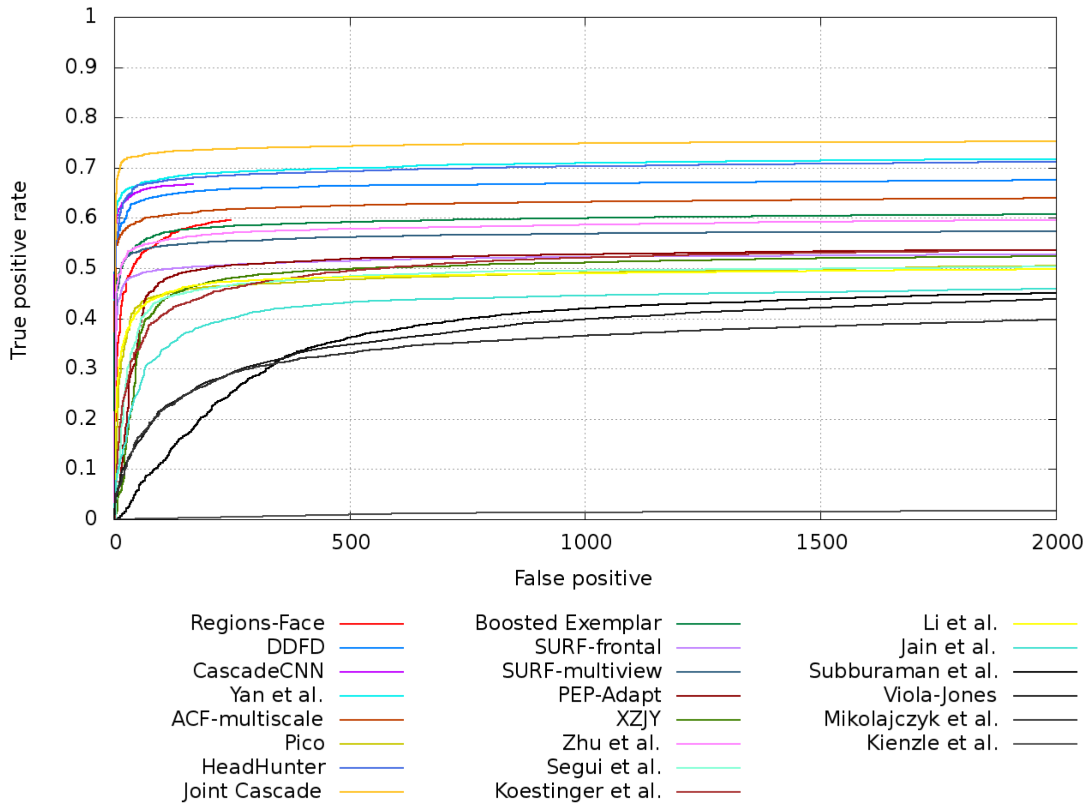


Figure 4.8: ROC Curves for Continous scores of various methods, Regions-Face(red) represents performance of our method

Chapter 5

Results and Discussion

5.1 Results

5.1.1 FDDB Benchmarking

Figure 4.7 and 4.8 show the ROC curves for various methods in face detection. In Discrete score ROC curve, we can see our method(Regions-Face) out-performing most of the methods. In continuous scores ROC curve, our method's performance is not the best as for this metric the evaluation is done using raw IOU ratios. This doesn't lead to good results for our method because of two reasons:

- Our region proposals are not exhaustive. In case of sliding window approach, where possible locations are in millions, it is easier to find a window with perfect IOU ratio. In our case, Regions generated by Segmentation which segments out similar regions and proposes each region as a posible location. So, it may not propose a window with a perfect IOU overlap.
- Some of the methods employ bounding box regression to improve localization performance. After getting a scored bounding box, a new bounding box is predicted using a class-specific bounding box regressor. Our method does not employ any such method, to improve the localization.

5.1.2 Scores for partial face regions

We tried to analyze how partial faces were scored against full faces. So, we mapped the scores of candidate regions as a trend along with the overlaps with the ground truth regions. As we can see in Figure 5.1, the score is highest when both Intersection with

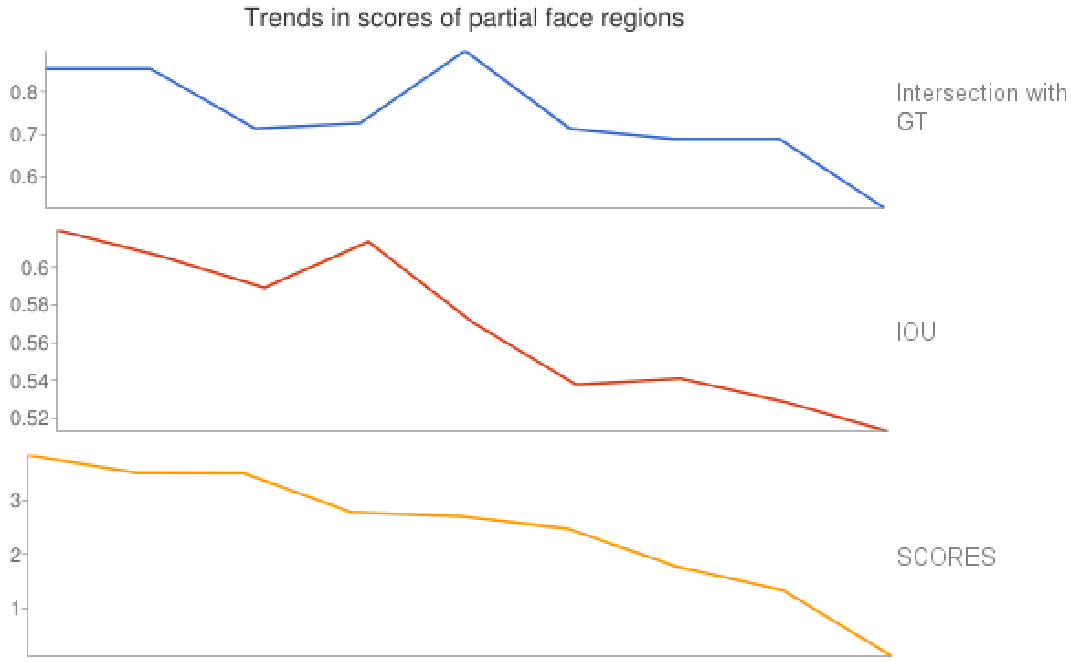


FIGURE 5.1: Trend of Scores for candidate bounding boxes along with their intersection overlap and IOU overlap

the ground truth region and Intersection over Union(IOU) with the ground truth region are highest. With about 85% of Intersection with ground truth region and about 70% of IOU, the region is scored somewhere about 4. As either of the overlaps reduce, the scores keep reducing until about 50% of overlap. Below this threshold, regions are not classified as faces.

Similar observations can be made from Figure 5.2. With about 95% of Intersection with ground truth region and about 90% of IOU, the region is scored above 5. As the overlaps reduce the scores keep reducing.

5.2 Discussion

According to Section 5.1.1, we can say that our method shows very promising results in case of Discrete scores with threshold set to 50% IOU overlap. In case of continuous scores, where the evaluation is done using the IOU overlap ratios themselves, there can be techniques which can be easily implemented. One technique would be bounding-box regression. Bounding-box regression predicts a new bounding-box which will have a better localization than the current scored bounding-box.

According to Section 5.1.2, we can say that the scores of a region might be useful in predicting how much of a face is present in that region. Regions covering the entire face

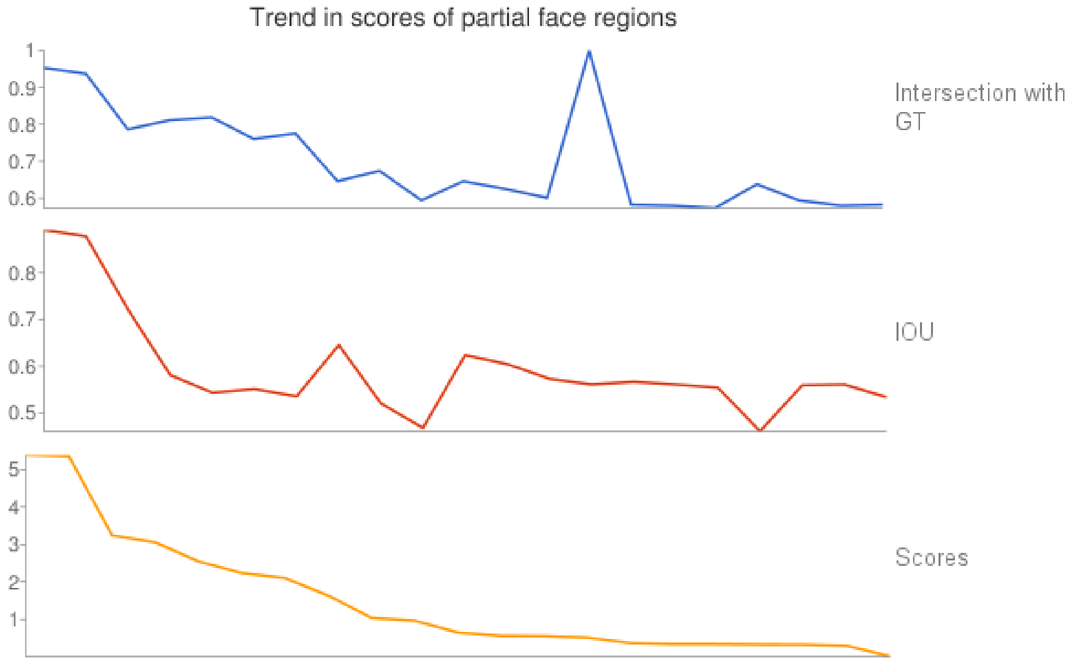


FIGURE 5.2: Trend of Scores for candidate bounding boxes along with their intersection overlap and IOU overlap

are always scored very high around 5, and as they cover lesser part of face the scores reduce.

Chapter 6

Conclusion

6.1 Where does our method work and why?

The proposed method can be used in real world scenarios like in a surveillance setting, where images are not captured in a constrained way and may have many occlusions which pose difficulty in detection of faces. It provides very high efficiency due to use of Rich Hierarchical Convolutional Neural Networks which are very good at feature extraction and representation. Selective Search using Segmentation decreases the number of candidate windows from millions to thousands. Candidate windows proposed by Selective Search contained parts of faces which can be very useful for partial face detection. Region-Face gave near best performance on the industry-standard Benchmark, FDDB.

6.2 Direction for Future Work

Performance of this method can be further improved by employing techniques like bounding box regression. Different ways to use local regions of faces generated by Selective Search to form part-based detection methods may be explored. For instance, using hierarchical segments to further improve efficiency. The region proposals generated by Selective Search may be further improved by using contextual information.

Bibliography

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[2] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition*, 2014.

[3] K. E. A. van de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders. Segmentation as selective search for object recognition. In *IEEE International Conference on Computer Vision*, 2011. URL https://ivi.fnwi.uva.nl/isis/publications/2011/vandeSandeICCV2011.

[4] Bernhard Frba and Christian Klbeck. Real-time face detection using edge-orientation matching. In Josef Bigun and Fabrizio Smeraldi, editors, *Audio- and Video-Based Biometric Person Authentication*, volume 2091 of *Lecture Notes in Computer Science*, pages 78–83. Springer Berlin Heidelberg, 2001. ISBN 978-3-540-42216-7. doi: 10.1007/3-540-45344-X_12. URL http://dx.doi.org/10.1007/3-540-45344-X_12.

[5] Oliver Jesorsky, Klaus J. Kirchberg, and Robert W. Frischholz. Robust face detection using the hausdorff distance. pages 90–95. Springer, 2001.

[6] Paul Viola and Michael J. Jones. Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154, May 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000013087.49260.fb. URL http://dx.doi.org/10.1023/B:VISI.0000013087.49260.fb.

[7] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade. Neural network-based face detection. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 20(1):23–38, 1998.

[8] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade. Rotation invariant neural network-based face detection. pages 38–44, 1998.

[9] Hazem M El-Bakry. Face detection using fast neural networks and image decomposition. *Neurocomputing*, 48(1):1039–1046, 2002.

[10] Marian Beszedes and Milos Oravec. A system for localization of human faces in images using neural networks, 2005.

[11] Zoran Bojkovic and Andreja Samcovic. Face Detection Approach in Neural Network Based Method for Video Surveillance. In *Seminar on Neural Network Applications in Electrical Engineering*, 2006. doi: 10.1109/NEUREL.2006.341172.

[12] Masakazu Matsugu, Katsuhiko Mori, Yusuke Mitari, and Yuji Kaneda. Subject independent facial expression recognition with robust face detection using a convolutional neural network. *Neural Netw.*, 16(5-6):555–559, June 2003. ISSN 0893-6080. doi: 10.1016/S0893-6080(03)00115-1. URL http://dx.doi.org/10.1016/S0893-6080(03)00115-1.

[13] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[14] Vidit Jain and Erik Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009, University of Massachusetts, Amherst, 2010.