A Densenet based Robust Face Detection Framework

Abhilash Nandy
Adobe Systems
India
nandyabhilash@gmail.com

Abstract

Face Detection has become important in various real-life applications such as face recognition, kinship verification, video surveillance, sentiment analysis using videos, etc. There has been significant progress in this field in recent years, thanks to the evolution of deep convolutional neural networks (CNNs). Images taken in real-world scenarios vary a lot in various aspects such as lighting, scale, pose, etc. WIDER FACE dataset contains such images, and is hence, quite challenging. In this paper, we propose a solution which takes the DSFD (Dual Shot Face Detector) as a baseline network, and we apply some tweaks to the network to improve performance with lesser memory usage and inference time. Specifically, we use a Densenet backbone, use focal loss function for classification, a function of IoU (Intersection over Union) metric as a regression loss function, and lastly, use the max-out operation before predicting class probabilities. Consequently, the proposed solution achieves state-of-the-art performance on the WIDER FACE Dataset, with added advantages of being more scalable and taking lesser time to infer than its original DSFD baseline. Also, it gives better face detection performance than many other state-of-the-art face detection frameworks.

1. Introduction

Face Detection is a very popular application of computer vision. It serves as a prerequisite for other important computer vision tasks such as face recognition, video surveillance, sentiment analysis using videos, kinship verification etc. Detecting faces comprises of two important parts - determining whether there is/are face(s) present in the image, and secondly, if there is atleast a face, determining the position and dimensions of the bounding box for each face. Technically, face detection comprises of two tasks - classification (whether a region within the image contains a face or not) and regression (to find the coordinates of the bounding box surrounding each face, if any). The hurdles faced in the field of face detection include various distortions in the image such as varying scale of faces (as in Fig. 2), varying pose, occlusion, blurry quality, facial makeup, spatially varying illumination, varying modality and reflection of faces.

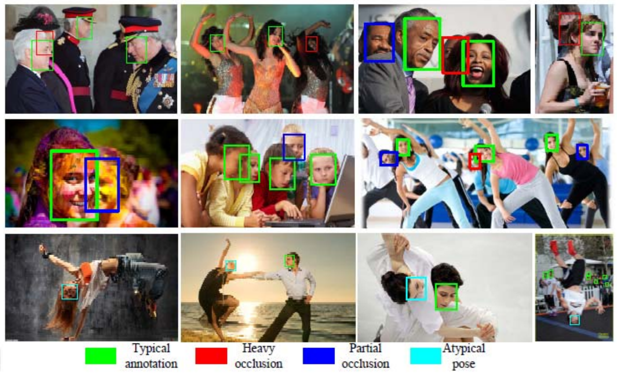


Figure 1: Different types of annotation in WIDER FACE Dataset [25]

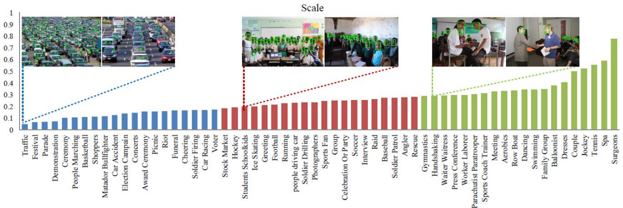


Figure 2: Variation in the scale of faces in the images of WIDER FACE Dataset [25]

In this paper, we lay our focus on a very challenging dataset that has various modes of distortions in the images as discussed above - the WIDER FACE Dataset [25]. Figure 1 shows the various types of annotation, and the variation in pose and occlusion in the images of the WIDER FACE Dataset. There has been a lot of work in increasing the performance metrics such as Average Precision (AP), Intersection over Union (IoU) etc. over the years. We propose a method, which uses the architecture of Dual Shot Face Detector proposed in [7] as the baseline, and modify this baseline in order to achieve better results. The modifications which we apply are: (1) Using Densenet [5] as the backbone. (2) applying focal loss function in order to estimate the probability of the presence of a face in a particular bounding box. (3) Using IoU (Intersection over Union) as a regression loss function instead of L1 loss. (4) Using the max-out operation for an improved classification performance

2. Prior Art

Face detection dates back to the early 1990s, which has since, served purpose in many fruitful applications such as identity verification and recognition, face alignment etc. The work of Viola-Jones [20] was one of the first works in this direction, which involved the training of multiple cascaded face detectors using Adaboost algorithm over Haar-like features, which worked quite well in simple situations. In [1], a cascaded network is used to perform both the tasks of face detection and alignment, giving quite promising results.

This was followed by the emergence of part-based models, Deformable Part Models (DPM) [13] being one of the most famous of all. DPM model the face as a collection of deformable parts, and the relationship between these parts can be established through the means of a latent SVM (Support Vector Machine) [2]. They worked better for occluded images as compared to the cascaded networks. However, these methods are not robust to most distortions in image quality and variations, since, these depend on hand-crafted features.

Aggregated Channel Feature solutions gave better results than the ones mentioned earlier, as can be seen in [24], which proved to be robust even when the images were taken from multiple views. This solution used features such as gradient histogram, integral histogram, and color channels in order to learn a boosting classifier with cascade structure.

With the advent of deep learning and greater processing power, it now became possible to handle more and more variations in the images, since, the data that could be used for training could now be huge. Inspired from cascade networks, [6] uses cascade-CNN technique, in which a series of multiple CNNs are trained in order to perform face detection, leading to both improved accuracy and efficiency at the same time. MTCNN [28] and PCN [17] train the network for multiple tasks such as detecting face angles and landmarks in addition to the primary task of face detection, in a coarse-to-fine manner.

As the datasets grew more diverse, especially in terms of the size of the face(s), it became more important to detect the face with smaller size, since the previous methods failed in such situations. Using multi-scale features for face detection was a plausible solution for detecting tiny faces. Inspired from [3, 11, 12], that use the fusion of features from multiple layers of the network for semantic segmentation and FPN [8], which uses a hierarchical architecture in order to fuse high level semantic features at all scales, Face Attention network [21] and PyramidBox [19] use such multi-layer fusion networks in order to detect faces. However, some models such as SSD [10] and RetinaNet [9] use a one-stage face detection network and give comparable performance. SSD has been used as the base network for many other better performing networks such as DSFD [7], S^3FD [29] etc., which have added modifications to DSFD, such as changing training strategy, applying reasoning based on context, using information from multiple layers etc.

3. Exposition to the solution

We use the DSFD (Dual Shot Face Detector) [7] as a baseline, and apply strategies mentioned in Section 1 in order to achieve state-of-the-art performance on the WIDER FACE [25] Dataset.

3.1. DSFD baseline

DSFD (Dual Shot Face Detector) is one of the state-of-the-art methods for performing face detection on the WIDER FACE dataset. Many methods involving multi-scale features fail to extract information from the current layer and ignore the context relationship between anchors [7]. DSFD solves these issues by using a feature enhance module that involves performs the operation of convoluted dilation at multiple layers in order to enhance the semantic of the features, as, it increases the receptive field of the network as a whole. DSFD uses backbone of VGG16 [18] or one of the versions of Resnet [4] or ResNeXt [22] at a time, removing the classification layers and incorporating some additional structures. (The architecture of DSFD is displayed in Fig. 3).

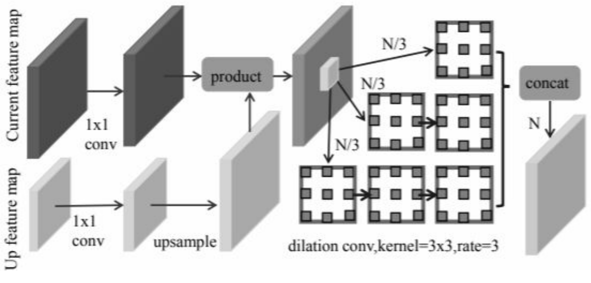


Figure 4: Feature Enhance Module architecture [7]

DSFD applies the feature enhance model on each of the feature maps (as shown in Fig. 4), generating the same number of 'enhanced' feature maps having the same size of the

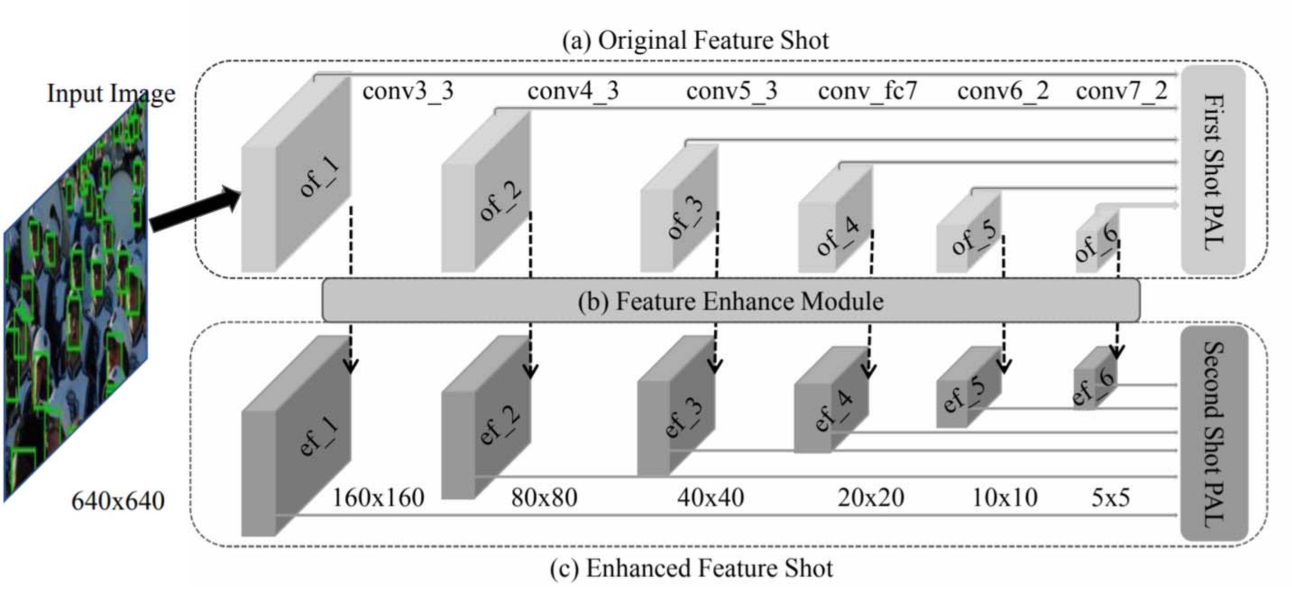


Figure 3: DSFD architecture [7] - Feature Enhance Module is applied over the VGG/resnet backbone in order to generate the enhanced feature shot pipeline. It also shows two loss layers, First Shot PAL (Progressive Anchor Loss) for the first pipeline and Second Shot PAL for the second pipeline

original feature maps. Initially, 1x1 convolutional filters are applied in order to the input feature maps. The resulting feature maps are split into three parts. Each part is followed by a sub-network consisting of dilated convolutional layers [26]. The number of convolutional layers for each part is different. These enhanced feature maps are fed into a SSD-style head in order to make the second shot detection layers.

The primary and the secondary shots have different loss functions, namely, First Shot progressive anchor Loss (FSL) and Second Shot progressive anchor Loss (SSL). The Progressive Anchor Loss (PAL) is not the same as the regular detection loss. As mentioned in [14], simpler features from the lower level of the network are more apt for smaller faces, the anchor sizes used in the first shot detection layers are small, and those used for the second shot detection are larger.

Mathematically, the loss function corresponding to the second shot is as follows -

$$L_{SSL}(p_i, p_i^*, t_i, g_i, a_i) = \frac{1}{N_{conf}} \sum_i L_{conf}(p_i, p_i^*) + \frac{\beta}{N_{loc}} \sum_i p_i^* L_{loc}(t_i, g_i, a_i), \quad (1)$$

where N_{conf} and N_{loc} are the number of positive and negative anchors respectively, L_{conf} is the loss function corresponding to classification between two classes - face and background, and L_{loc} is the localization loss or bounding box regression loss, calculated between the predicted bounding box t_i and ground truth bounding box g_i corresponding to the anchor a_i. Localization loss is applicable for positive anchor, i.e., when $p_i^* = 1$. β is a weighting factor, which gives a weight to the localization loss (L_{loc}) term relative to the classification loss term corresponding to the confidence score (L_{conf}).

Similarly, for the first shot detection layers, the corresponding feature maps have simpler information, but with a higher resolution, as compared to the feature maps in the second shot detection (due to smaller receptive field for the first shot detection layers). Thus, the first shot detection pipeline would be able to detect smaller faces. Mathematically, the loss function for the first shot detection is as follows -

$$L_{FSL}(p_i, p_i^*, t_i, g_i, sa_i) = \frac{1}{N_{conf}} \sum_i L_{conf}(p_i, p_i^*) + \frac{\beta}{N_{loc}} \sum_i p_i^* L_{loc}(t_i, g_i, sa_i), \quad (2)$$

where sa refers to the smaller-sized anchors in the first shot detection layers. (The other terms in equation 2 have the same meaning as those in equation 1).

The losses corresponding to the two shots can be summed in a weighted manner in order to give a combined loss function, known as the Progressive Anchor Loss

(L_{PAL}), given mathematically as -

$$L_{PAL} = L_{FSL}(sa) + \lambda L_{SSL}(a) \tag{3}$$

where λ is the weight factor given to the second shot detection loss relative to the first shot detection loss. (The other terms in equation 3 have the same meaning as those in equations 1 and 2).

In general, the number of positive anchors would be very much less than the number of negative anchors, since, the faces occupy a relatively smaller area as compared to the background. In order to tackle this problem, DSFD uses a probabilistic augmentation scheme. 2 out of 5 times, it augments by randomly picking a face in an image, crop sub-image containing that face, and set the spatial ratio between the sub-image and the face to be $640/rand(16, 32, 64, 128, 256, 512)$, where $16, 32, 64, 128, 256, 512$ refer to some anchor sizes used. Rest of the times, the data augmentation procedure is similar to that of SSD [10].

3.2. Using Densenet as the network backbone

DSFD uses FGG-16, Resnet and ResNeXt architectures as backbones. However, these architectures have a lot of parameters, thus increasing memory usage as well as time taken for training and inference. In order to tackle this problem, we use Densenet-121 and Densenet-169 [5] as the backbones instead. In the Densenet Architecture, a convolutional block not only receives the feature maps immediately preceding layer it as inputs, but also, it receives all other feature maps from all the layers before this layer. This is similar to Resnet in a way that it also has gradient flow paths across non-adjacent layers, but the difference in Densenet is that, the feature maps form the previous layers are concatenated, instead of being added as in Resnet.

The feature maps used in the detection framework comprise the feature map after the first convolutional layer, and the feature maps obtained after the application of each Dense Block of Densenet. Table 1 depicts the architecture of Densenet, where, the feature maps highlighted in blue are the ones used in our detection framework.

3.3. Focal Loss for classification

The DSFD framework uses the softmax loss for classification. As already discussed in Section 3.1, if we consider the face as a class and background as the second class, there is a lot of imbalance among the two classes. Also, in order to have commendable performance, the well-classified examples need to have lower weights as compared to the highly mis-classified ones. The softmax loss does not address these issues. However, using the α-based variant of the focal loss function [9] as the classification loss solves both these problems. Mathematically, L_{conf} can be written as a function of p_t in the following manner -

$$L_{conf}(p_t) = -\alpha_t (1 - p_t)^{\gamma} log(p_t), \tag{4}$$

where

$$p_t = \begin{cases} p, & \text{if } y = 1 \\ 1 - p, & \text{otherwise} \end{cases} \tag{5}$$

$p \in \{0, 1\}$ is the predicted probability corresponding to the positive class (here it is the presence of face), $y \in [\pm 1]$ is the ground-truth class, α_t is the weight corresponding to the class for p_t and $\gamma \geq 0$ is a tunable parameter, that is responsible for the training to focus more on the mis-classified examples, rather than the easier examples. If p_t is closer to 1, i.e., if the model predicts correctly, $1 - p_t$ is closer to 0, and so, the term $(1 - p_t)^{\gamma}$ lowers the contribution of p_t towards the total loss, as desired. The focal loss function gives better results as compared to the softmax loss on the WIDER FACE dataset, as expected.

3.4. IoU regression Loss

IoU (Intersection over Union) is one of the most popular metrics used for the evaluation of face detection. The higher the IoU metric, the better is the quality of face detection. The DSFD framework used smooth L1 loss function for regression. However, according to [15], optimizing the smooth L1 loss function for bounding box regression is not sufficient for the IoU metric value to be maximized. Inspired from [27], in order to maximize the IoU value, we consider the negative logarithm of the IoU value as the regression loss function and minimize it. Mathematically, the regression/localisation loss function L_{loc} corresponding to the i_{th} ground truth bounding box can be written as follows -

$$L_{loc} = -log \frac{Intersection(t_i, g_i)}{Union(t_i, g_i)} \tag{6}$$

where t_i and g_i are the i_{th} predicted bounding box and i_{th} ground truth bounding box respectively, and $Intersection()$ and $Union()$ give the intersection area and the union area respectively between t_i and g_i.

3.5. Max-out operation

The background region is much larger in area than the region that contains faces in an image. Hence, it becomes necessary to reduce the number of false positives to improve performance. As mentioned in [29], the max out operation is used in order to solve this issue. In this paper, we apply the max-out operation while performing classification not only reduce to the number of false positives, but also to increase the number of true positives, thus increasing precision.

For datasets such as the WIDER FACE dataset, where the images have not been taken in restricted conditions, the

Table 1: Architecture of Densenet - The feature maps highlighted in blue are the ones that are used in our proposed solution, in order to create the first shot pipeline

Layers	Output Size	DenseNet-121		DenseNet-169		DenseNet-201		DenseNet-264	
Convolution	112 x 112	7 x 7 conv, stride 2							
Pooling	56 x 56	3 x 3 max pool, stride 2							
Dense Block (1)	56 x 56	1 x 1 conv 3 x 3 conv	x 6	1 x 1 conv 3 x 3 conv	x 6	1 x 1 conv 3 x 3 conv	x 6	1 x 1 conv 3 x 3 conv	x 6
Transition Layer (1)	56 x 56	1 x 1 conv							
	28 x 28	2 x 2 average pool, stride 2							
Dense Block (2)	28 x 28	1 x 1 conv 3 x 3 conv	x 12	1 x 1 conv 3 x 3 conv	x 12	1 x 1 conv 3 x 3 conv	x 12	1 x 1 conv 3 x 3 conv	x 12
Transition Layer (2)	28 x 28	1 x 1 conv							
	14 x 14	2 x 2 average pool, stride 2							
Dense Block (3)	14 x 14	1 x 1 conv 3 x 3 conv	x 24	1 x 1 conv 3 x 3 conv	x 32	1 x 1 conv 3 x 3 conv	x 48	1 x 1 conv 3 x 3 conv	x 64
Transition Layer (3)	14 x 14	1 x 1 conv							
	7 x 7	2 x 2 average pool, stride 2							
Dense Block (4)	7 x 7	1 x 1 conv 3 x 3 conv	x 16	1 x 1 conv 3 x 3 conv	x 32	1 x 1 conv 3 x 3 conv	x 32	1 x 1 conv 3 x 3 conv	x 48
Classification Layer	1 x 1	7 x 7 global average pool							
		1000D fully-connected, softmax							

background region has a lot of spatial variations. Hence, it is insufficient to assign only one class to the entire background. Rather, we assign C_n number of latent sub-classes to the background. Similarly, we assign C_p number of latent sub-classes to the face class. Then, maximum of the C_p outputs and the maximum of the other C_n outputs are taken, and softmax function is applied over these two values, in order to calculate the probabilities of the anchor belonging to the face class and the background class respectively. Predicting multiple outputs for each class increases the classification accuracy for both the classes.

4. Experiments and Results

4.1. Dataset Description

WIDER FACE dataset is one of the largest face detection datasets. The images in this dataset are taken from the publicly available WIDER dataset [23]. The WIDER FACE dataset consists of 32,303 images and 393,703 annotated faces. As discussed in [7], the train:validation:test split for the dataset is 40:10:50, in a random fashion for each of the 60 event classes. Again, each subset is again split into three levels of difficulty - 'Easy', 'Medium' and 'Hard', which is based on the performance of a baseline detector.

As already discussed in Section 1, the images have been taken in highly unconstrained conditions. There are large variations in lighting, scale (as in Fig. 2), pose, occlusion, background clutters etc. The images belong to 60 different events. All these variations within the dataset make the dataset challenging for performing face detection.

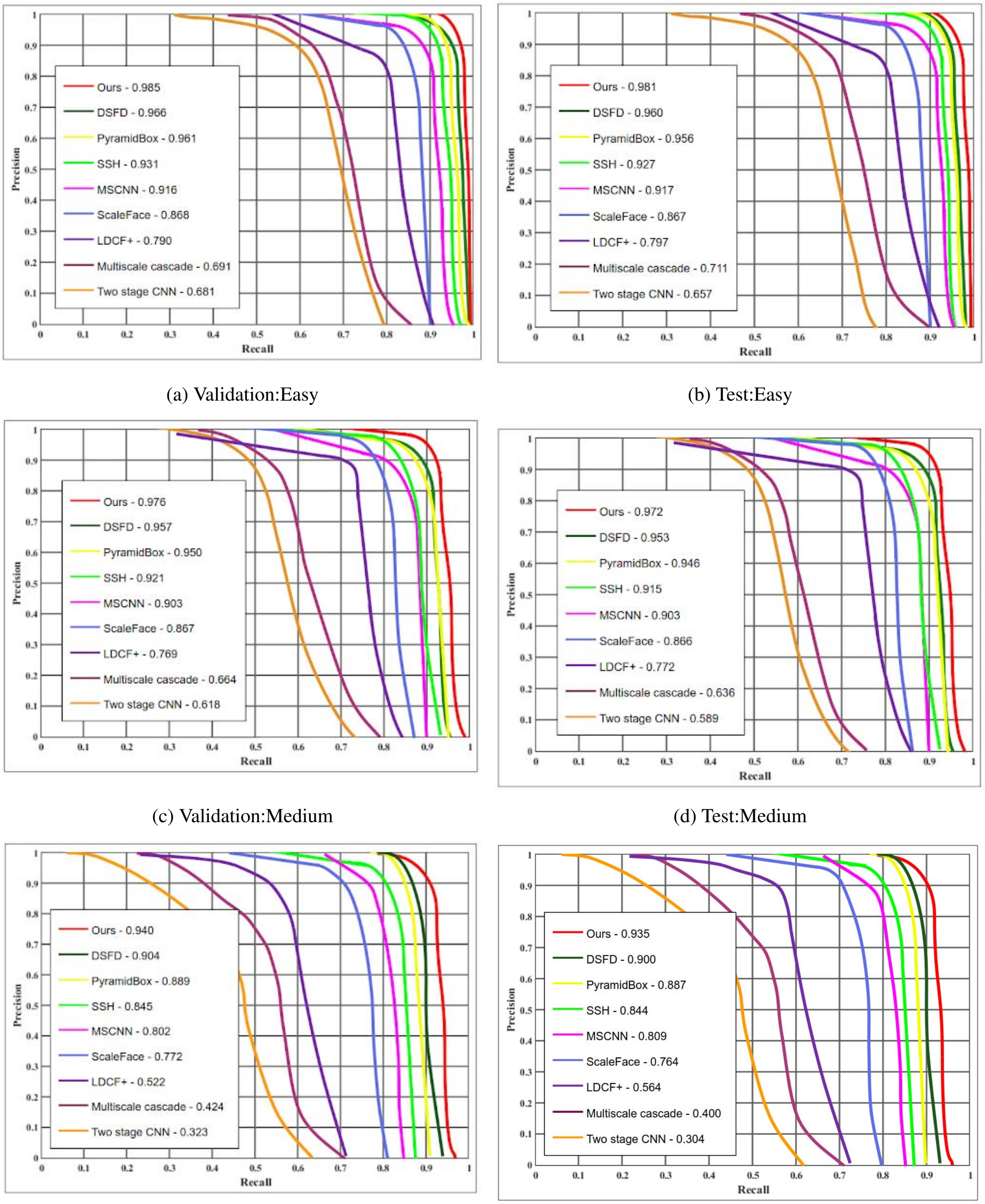
4.2. Implementation Details

The backbone used in our solution is a version of Densenet. This is initialized by the pretrained weights obtained by training the Densenet network on ImageNet [16] Dataset. Parameters of the additional convolution layers are initialized by the Xavier method, and the optimizer used in order to fine-tune our model is SGD with 0.9 momentum and 0.0005 weight decay, as used in the original DSFD paper. The batch size is set to 16 and the learning rate is set to 0.001 for the first 40k steps, and we decay it to 10^{-4} and 10^{-5} for two $10k$ steps, again as stated in the original DSFD paper.

During inference, multi-scale testing [29] is applied in order to improve performance. The input image is fed to the trained model several number of times with varying sizes, and then, these detection results are combined along with the voting operation of bounding boxes. The outputs from the first shot detection pipeline are ignored and the second shot pipeline is used in order to predict top $5k$ detections according to confidence, of which, 750 bounding boxes of high confidence are obtained by performing Non-maximum suppression with a Jaccard Overlap of 0.3.

4.3. Analysis on the tweaks applied to DSFD

In this subsection, the effectiveness of adding the tweaks mentioned earlier in Section 3 is analyzed, by performing evaluation on the WIDER FACE Dataset based on



(a) Validation:Easy

(b) Test:Easy

(c) Validation:Medium

(d) Test:Medium

(e) Validation:Hard

(f) Test:Hard

Figure 5: Precision-recall curves on WIDER FACE testing subset (The scores are the Average Precision values, and the score mentioned for the DSFD baseline is achieved using the Resnet-50 backbone)

Precision-Recall Curves, performance metrics such as Average Precision (AP) and Top-1 Accuracy, and the num-

Table 2: Comparison of various models based on Top-1 Accuracy and Average Precision

Network	Top-1 Accuracy	Easy	Medium	Hard
DSFD_resnet152	80.19%	96.6%	95.7%	90.4%
DSFD_resnext101	78.42%	95.7%	94.8%	88.9%
DSFD_densenet121	80.54%	96.0%	95.0%	91.0%
DSFD_densenet169	82.56%	96.8%	95.8%	92.3%
DSFD_densenet169_focal	84.47%	96.9%	96.2%	92.5%
DSFD_densenet169_focal_iou	84.51%	97.2%	96.1%	92.7%
DSFD_densenet169_focal_iou_maxout	85.22%	98.1%	97.2%	93.5%

ber of parameters involved in the network. The tweaks are added one of top of another. The Table 2 depicts the results of the various models, where 'DSFD' refers to the baseline network, 'resnet152', 'resnext101', 'densenet121', 'densenet169' refer to the backbones of Resnet-152, ResNeXt-101, Densenet-121 and Densenet-169 respectively, 'focal', 'iou' and 'maxout' refer to the focal loss for classification, IoU regression loss and max-out operation respectively.

Using Densenet as backbone Using Densenet-121 as the backbone beats ResNeXt backbone, whereas, it performs nearly as well as Resnet-152 backbone. However, using Densenet-169 improves the Top-1 Accuracy and the Average Precision in the Hard category considerably as compared to when using the ResNeXt-101 backbone. Also, the number of parameters required for each of the Densenet backbones was far less than the same for Resnet-152 and ResNeXt-101 (as can be seen in Table 3). Lesser number of parameters in the Densenet backbones led to lesser usage of memory and lesser inference time, as compared to the Resnet and ResNeXt backbones.

Table 3: Number of parameters for various backbones used

Backbone used	Params
Resnet-152	459M
ResNeXt-101	416M
Densenet-121	275M
Densenet-169	324M

Focal Loss for classification Using this loss function instead of the classical categorical cross-entropy function (stated otherwise as the softmax loss function) again improves all performance metrics, as desired.

IoU Regression Loss Using the negative logarithm of the IoU as the regression loss function created a marginal change in the average precision as well as the Top-1 Accuracy.

Max-out operation Introducing latent sub-classes for both the face and the background classes increases both the average precision and the classification accuracy, as expected.

As can be seen in Table 2 and the Fig. 5, our final model ('DSFD_densenet169_focal_iou_maxout') outperforms the DSFD baselines by achieving a Top-1 Accuracy of 85.22%. It also beats many state-of-the-art methods, by achieving a 98.1% AP in the 'Easy' category, 97.2% AP in the 'Medium' category, and 93.5% AP in the 'Hard' category in the 'Test' subset, and a 98.5% AP in the 'Easy' category, 97.6% AP in the 'Medium' category, and 94.0% AP in the 'Hard' category in the 'Validation' subset.

5. Conclusion

The paper introduces a robust face detection framework, by bringing some changes to the DSFD(Dual Shot Face Detector), which are - (1) Using the Densenet Backbone (2) Using the focal loss function for classification (3) Using the IoU regression loss (4) Using the max-out operation. With these changes introduced, our face detection framework outperformed various state-of-the-art baselines, and at the same time, required lesser parameters, thus reducing memory usage and inference time. This suggests that our framework is more scalable than many other state-of-the-art face detectors.

References

[1] D. Chen, S. Ren, Y. Wei, X. Cao, and J. Sun. Joint cascade face detection and alignment. In *European Conference on Computer Vision*, pages 109–122. Springer, 2014. 2

[2] 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*, 32(9):1627–1645, 2009. 2

[3] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 447–456, 2015. 2

[4] 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*, pages 770–778, 2016. 2

[5] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE*

Conference on Computer Vision and Pattern Recognition (CVPR), pages 2261–2269, July 2017. 2, 4

[6] 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*, pages 5325–5334, 2015. 2

[7] J. Li, Y. Wang, C. Wang, Y. Tai, J. Qian, J. Yang, C. Wang, J. Li, and F. Huang. Dsfd: dual shot face detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5060–5069, 2019. 2, 3, 5

[8] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 2

[9] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2, 4

[10] 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*, pages 21–37. Springer, 2016. 2, 4

[11] W. Liu, A. Rabinovich, and A. Berg. Parsenet: Looking wider to see better. In *Proceedings of International Conference on Learning Representations Workshop*, 2016. 2

[12] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 2

[13] M. Mathias, R. Benenson, M. Pedersoli, and L. Van Gool. Face detection without bells and whistles. In *European conference on computer vision*, pages 720–735. Springer, 2014. 2

[14] 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*, pages 91–99, 2015. 3

[15] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 658–666, 2019. 4

[16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 5

[17] X. Shi, S. Shan, M. Kan, S. Wu, and X. Chen. Real-time rotation-invariant face detection with progressive calibration networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2295–2303, 2018. 2

[18] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[19] X. Tang, D. K. Du, Z. He, and J. Liu. Pyramidbox: A context-assisted single shot face detector. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 797–813, 2018. 2

[20] P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004. 2

[21] J. Wang, Y. Yuan, and G. Yu. Face attention network: An effective face detector for the occluded faces. arxiv 2017. *arXiv preprint arXiv:1711.07246*. 2

[22] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 2

[23] Y. Xiong, K. Zhu, D. Lin, and X. Tang. Recognize complex events from static images by fusing deep channels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1600–1609, 2015. 5

[24] B. Yang, J. Yan, Z. Lei, and S. Z. Li. Aggregate channel features for multi-view face detection. In *IEEE international joint conference on biometrics*, pages 1–8. IEEE, 2014. 2

[25] 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. 1, 2

[26] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *CoRR*, abs/1511.07122, 2015. 3

[27] J. Yu, Y. Jiang, Z. Wang, Z. Cao, and T. Huang. Unitbox: An advanced object detection network. In *Proceedings of the 24th ACM international conference on Multimedia*, pages 516–520. ACM, 2016. 4

[28] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 2016. 2

[29] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li. S3fd: Single shot scale-invariant face detector. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 192–201, 2017. 2, 4, 5