Assessing the Effect of Image Quality on SSD and Faster R-CNN Networks for Face Detection

Mosab Rezaei
Dept. of Engineering
Shahid Chamran University of Ahvaz
Ahvaz, Iran
m-rezaei@stu.scu.ac.ir

Elhamossadat Ravanbakhsh
Dept. of Engineering
Shahid Chamran University of Ahvaz
Ahvaz, Iran
e-ravanbakhsh@stu.scu.ac.ir

Ehsan Namjoo
Dept. of Engineering
Shahid Chamran University of Ahvaz
Ahvaz, Iran
e.namjoo@scu.ac.ir

Mohammad Haghighat
Dept. of ECE
University of Miami
Coral Gables, FL, USA
haghighat@umiami.edu

***Abstract*—Face detection is one of the most challenging and long-studied areas in computer vision. In real-world, images are exposed to the noise and degradation. In this paper, we investigate the robustness of two networks namely SSD and Faster R-CNN in confrontation with salt and pepper noise, Gaussian blur, as well as JPEG compression. Our experiments are conducted on the well-known Wider Face dataset. These experiments show that the Faster R-CNN is more robust against Gaussian blur, while SSD is much more sensitive to the edges. On the other hand, SSD is more robust against reduced-quality JPEG compressed images. The reason should be due to the sensitivity of Faster R-CNN to the texture of the objects. Moreover, our experiments demonstrated that both networks have a relatively similar resistance under salt and pepper noise.**

Keywords- Face detection, SSD, Faster R-CNN, image quality

I. Introduction

Face detection is one of the most popular and still challenging hot topics in the field of computer vision and machine learning. Nowadays, by fast development of deep learning algorithms, an expanding range of different networks have been designed specifically to work on former well-known problems. Among diverse network structures, some like R-CNN [1], Fast R-CNN [2], Faster R-CNN [3], SSD [4], MTCNN [5] and Hyperface [6] have been designed for object detection. Although a lot of studies on developing optimum architecture to improve precision have been accomplished, there is not adequate research to assess the behavior of deep learning networks in confrontation with artifacts such as blur, noise and lossy compression.

In [7], the robustness of three well-known convolutional networks, namely Alex-Net [8], VGG-Net [9] and Google-Net [10], under different noise levels and different degradations like blurriness and occlusion has been assessed. In a similar work, in [11], a related work has been done on Squeeze-Net [12]. In other related work, in [13], the robustness of some architectures under different kinds of noise is assessed. In [14], an evaluation of four deep neural networks under different quality distortions is provided. Also, in [15] the effect of drop-out on robustness of the network is investigated. For this network, authors test the effect of the noise on generative adversarial examples, and use the achieved results to design more robust networks. In [16], the performance of Faster R-CNN and S^3FD [17] is investigated on low quality images with different levels of blur, noise and contrast.

In this paper, the effect of different kinds of degradations namely Gaussian blur and salt and pepper noise as well as JPEG compression on the detection performance of SSD and Faster R-CNN is investigated when they are used as face detectors. According to [16], deep face detectors are categorized into three classes: Cascade CNN, Faster R-CNN and SSD based networks. Since Cascade CNN approach is not able to find tiny and blurry faces in crowd, the paper is focused on SSD and Faster R-CNN networks.

SSD, as a fast and reliable network, has attracted many machine vision researchers. On the other hand, although Faster R-CNN is not as fast as SSD, it is a more reliable and precise network. Assessing their performance under different kinds of degradation could be beneficial to use them wisely in appropriate detection scenarios. In this work, the Mobile-Net [18] architecture is utilized as the base feature extractor for the SSD network. Also for the Faster R-CNN, the ResNet50 [19] is utilized as the base feature extractor. The simulation is based on [20].

The remaining of the paper is organized as follows: in Section II, the SSD architecture used in this paper is presented. In Section III, the Faster R-CNN is explained. In Section IV, the dataset that is used in this paper is introduced. Section V provides simulation results and finally the paper is concluded in Section VI.

II. SSD Architecture

SSD network is composed of three parts: The first part is actually a truncated base network. A truncated base network can be any convolutional neural network structure that is truncated before fully connected layers.

Input image passes through the base truncated network and produces a tensor. The produced tensor acts as the input to the second part of the SSD network.

In the second part, SSD adds some convolutional layers to the end of the base network. The dimensions of these layers are being decreased by moving towards the third part. The decreased

978-1-7281-1508-5/19/$31.00 ©2019 IEEE

size convolutional layers have a more general perception of the input image while the primitive convolution layers focus on more specific details. Since all of these convolutional layers are involved in the object detection procedure, these layers act like a pyramid image.

In Fig. 1, the base network gets the input image (a) and produces the last feather map (b). For simplicity, the base network is not shown and only the input and output of the network is illustrated. As seen in part (c), some convolutional operations are performed on the last feature map and gradually the new feature maps' size is decreased.

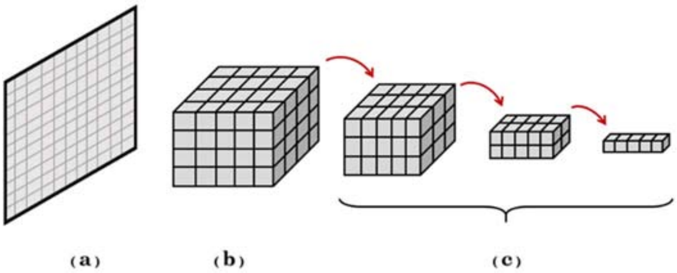


Fig. 1. a) Input image. b) Last feature map. c) Extra feature maps produced by some consecutive convolutional operations on the last feature map

Finally, in the third part, the network tries to find the exact location of each object. In order to find any specific object in the input image, there is a set of filters associated to any specific class that is applied separately on convolutional layers in part (b) and (c) from Fig. 1. The main differences between filters of any specific class refers to default boxes defined for each filter. In Fig. 2, applying different filters from a specific class on a convolutional layer is shown. Similar procedure is performed for other distinguished classes and related convolutional layers.

Default boxes are produced by applying filters with different aspect ratios on input image. SSD evaluates a set of default boxes with different aspect ratios at each location in several feature maps with different scales. In the case that a default box overlaps over a half with the ground truth box, it is considered positive.

As shown in Fig. 2, for a single class, there are three default boxes. It means that three different filters are being applied on a specific region of the feature map to search for a similar class. The major difference among different default boxes is their aspect ratios. In other words, when a filter is sensitive to a region, the overlapping rate with the ground truth is investigated just for that region.

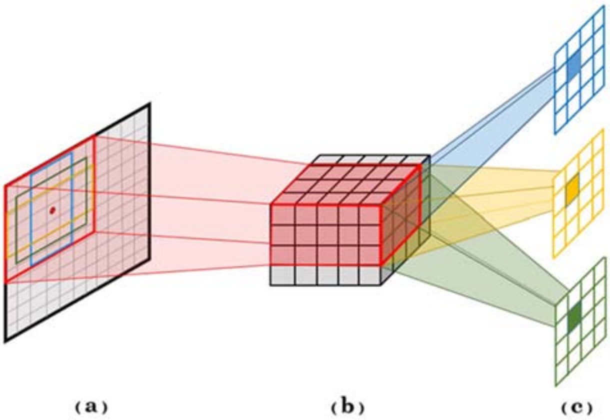


Fig. 2. a) Input image. b) Feature map matrixes (A region that the 3×3 filter is looking at, has been shown in red). c) three different outputs for three default boxes

Since the output of each filter relates to the existance of an object in a specific location in input image, considering the results from all other filters leads to a strong detector that is able to find the exact objects' locations. It is worth mentioning that in this architecture, there are four bounding box regression filters to modify the object position in the best way possible. For simplicity, these filters are not shown in Fig 2.

III. FASTER R-CNN ARCHITECTURE

Similar to the SSD network, there are also three main parts in Faster RCNN. First part is a base network. In this work, we have used ResNet50 as the base network. Similar to SSD network, the base network gets the input image and produces a tensor that is actually the output of the last convolutional layer of the ResNet50. This tensor will be fed into the second part of the Faster R-CNN.

The second part, Region Proposal Network (RPN), is obligated to find places that are likely to contain an object from predefined classes. After finding these places, they are passed to the next layer to figure out the correct class. In fact, the third part acts like a classifier and determines the class that the object belongs to. The outstanding difference between what is done by the RPN and what is performed by classic methods like selective search [21] is that RPN is sensitive just to the objects that are predefined for the whole network. In other words, RPN doesn't try to find the class of objects, instead, it tries to find objects that are supposed to be detected by the whole network. For this purpose, at first 3×3 filters are applied to the feature map generated by the base network (as depicted in Fig. 3) so that another feature map with the same dimensions is generated. Then, a collection of nine 1×1 filters are applied to generate the feature map. Every collection of filters is actually a different anchor.

One can say anchors act like default boxes in the SSD network. It means that even though all anchors take a look at the same region, but any of them are sensitive to specific part of it. Since the classification is not performed in this step, the output result for each anchor includes just two options YES and NO that determines whether the object exists or not. Similar to the SSD network, for each anchor, besides YES and NO filters, there exists four extra filters for bounding box regression (six filters in total). In this case, nine different anchors have six filters. So, the total number of filters in this case is equal to 54.

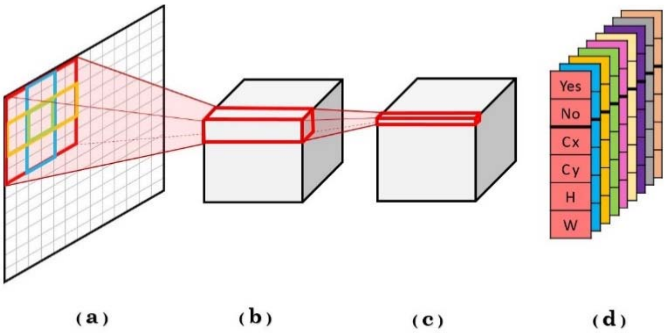


Fig. 3. a) Input image. b) The last feature map generated by base network. c) The feature map generated by performing 3×3 filters on feature map in part b. d) A table with 54 cells each contains the result of applying distingushing 1×1 filters over the previous featur map.

In Fig. 3, each anchor is distinguished with a different color. For simplicity, only the mapping of four regions in which each anchor is interested to is depicted by four colors green, blue, yellow, and red. The related region for each color is also depicted in part (a).

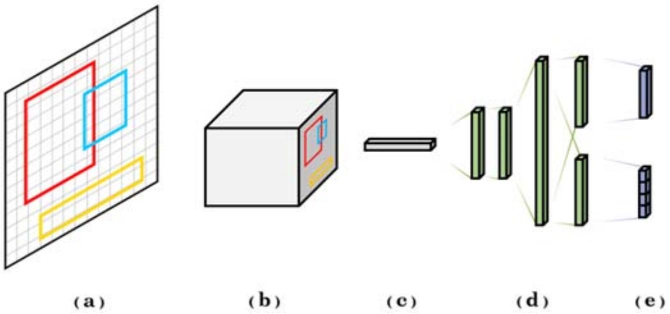


Fig. 4. a) Input image with some parts that are distingushed as objects. b) Final featur map and their corresponded region from original input image. c) Spatial pyramid pooling. d) fully connected layers. e) Network's final arrays.

As mentioned before, the third part is the classifier. A schematic of this part is depicted in Fig. 4. This part uses the output of pervious step, i.e., the regions of the interest with high probability. In this step, according to position of every candidate object in original image, its corresponding position on the last feature map is determined. Then the determined region is cut and passed to the spatial pyramid pooling [22] layer to become a fixed length array. After that, the array is passed thorough a set of fully connected layers and finally is separated into two distinguished parts. The first part, regression, have four outputs that is in fact the Bounding Box Regression (BBR). The next part is the softmax array cells that include a value for each predefined class and another cell that is dedicated to the background. Finally, if a region proposal is distinguished as a member of any of the predefined classes, its position gets modified by the BBR; otherwise it is assumed as background and there is no need for BBR.

IV. Datasets

To train the network, the well-known Wider Face dataset [23] is used. Wider face consists of 32,203 images in 61 different event classes. In total, there are 393,703 labeled face sub-images in this database. For each face sub-image, labels include blurriness, size, pose and occlusion. Also there is possibility for researchers to validate their algorithms in three different modes namely easy, medium, and hard. Since the test images are unlabeled, the images from validation set is used to assess the precision of face detection procedure. Fig. 5 shows some images from Wider Face dataset.

Fig. 5. Some images from Wider Face dataset.

V. Experiments

To investigate the effect of image quality on the performance of SSD and Faster R-CNN networks, two different experiments are considered.

A. *The First Experiment*

In the first experiment, the procedure is commonplace; it means that the network is trained on original training sample images without any additional noise or compression and then the accuracy is also evaluated with normal test samples. The experiment is considered as a baseline reference to assess the degradation effect in the following experiments. TABLE I shows the accuracy in three different modes of the Wider Face dataset.

TABLE I. MAP OF SSD AND FASTER R-CNN

Network	Easy	Medium	Hard
SSD	0.63	0.58	0.43
Faster R-CNN	0.82	0.81	0.56

B. *The Second Experiment*

In the second experiment, the network that is trained with normal samples, is tested on different kinds of noisy, blurry and reduced-quality compressed JPEG images. The experimental results from the second experiment provide some rational clue to find out how much the SSD and Faster R-CNN could tolerate noise and degradation in case that the network is trained with sound and perfect samples.

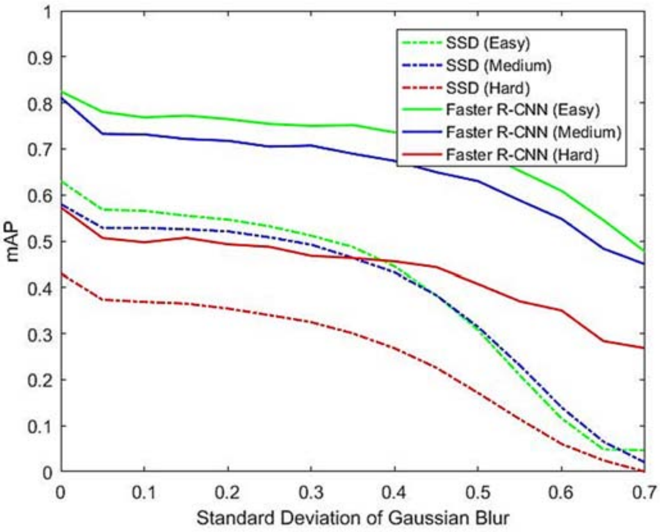


Fig. 6. The effect of Gaussian blur on test samples.

TABLE II. PERCENTAGE OF REDUCTION IN MAP FOR GAUSSIAN BLUR

σ	Network	Easy	Medium	Hard
0.1	SSD Faster R-CNN	-10.3% -6.9%	-8.9% -9.9%	-14.2% -13.3%
0.2	SSD Faster R-CNN	-13.3% -7.3%	-10.2% -11.6%	-17.6% -14%
0.3	SSD Faster R-CNN	-18.8% -9%	-15.2% -12.8%	-24.4% -18.3%
0.4	SSD Faster R-CNN	-29.4% -10.8%	-25.6% -17%	-37.7% -20.5%
0.5	SSD Faster R-CNN	-51.2% -15.7%	-45.8% -22.4%	-60.23% -28.9%
0.6	SSD Faster R-CNN	-81.6% -26.2%	-75.9% -32.5%	-86% -38.9%
0.7	SSD Faster R-CNN	-92.6% -48.3%	-96.6% -44.6%	-100% -53.2%

In Fig. 6, the effect of Gaussian blur on test images running by SSD and Faster R-CNN has been shown. It can be concluded that both networks demonstrate noticeable declining in mAP under the highest variance of blurry images. To analyze the robustness of two networks, two factors are considered: percentage of reduction in mAP of two networks in confrontation with degradation and percentage of reduction with the increment of degradation level gradually.

The percentage of reduction in mAP for blurry test samples is shown in TABLE II. Based on the results, for a Gaussian blur with standard deviation of 0.7, the mAP of Faster R-CNN for Easy, Medium and Hard samples has reduced 48.3%, 44.6% and 53.2%, respectively while the mAP of SSD has reduced 92.6%, 96.6% and 100%, respectively. The results in TABLE II shows that the Faster R-CNN has less reduction in mAP than SSD. Another interesting phenomenon is that the accuracy of the Faster R-CNN does not change much when increasing the standard deviations from 0.05 and 0.4, which shows a kind of stability in this range. All in all, it can be concluded that Faster R-CNN demonstrates a higher robustness on blurry samples in comparison to SSD.

Considering the fact that blur destroys the edges, one can conclude that SSD is more sensitive to the edges. Hence, blurring the image drastically declines the accuracy of the network. Consequently, in the situations in which there is the possibility of blurriness, e.g., object detection in videos with high amount of motion blur, Faster R-CNN seems to be a better choice.

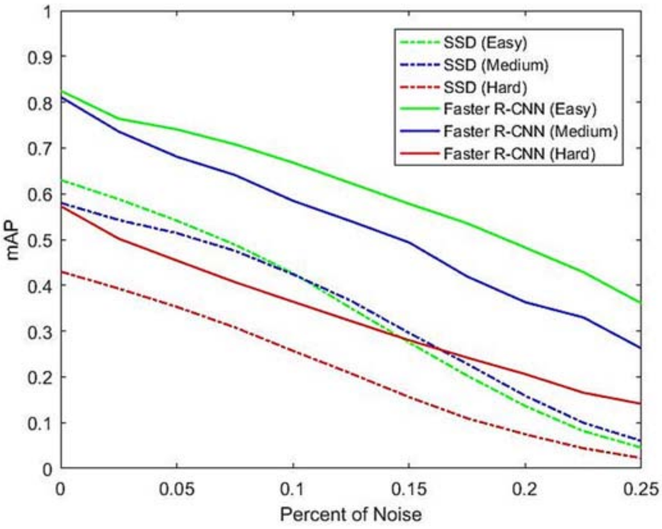


Fig. 7. The effect of salt and pepper noise on test samples

TABLE III. PERCENTAGE OF REDUCTION IN MAP FOR SALT AND PEPPER NOISE

%	Network	Easy	Medium	Hard
0.05	SSD Faster R-CNN	-14.1% -10.2%	-11.4% -16.1%	-17.9% -20.8%
0.10	SSD Faster R-CNN	-32.4% -19%	-26.9% -28%	-40.2% -36.5%
0.15	SSD Faster R-CNN	-56.4% -29.9%	-49% -39.2%	-63.8% -51.2%
0.20	SSD Faster R-CNN	-78.4% -41.5%	-72.7% -55.3%	-82.7% -64.1%
0.25	SSD Faster R-CNN	-92.8% -56.3%	-89.7% -67.7%	-94.7% -75.4%

The next experiment is to measure the robustness of these two networks against salt and pepper noise, which is a common artifact in low exposure images. According to the results in Fig. 7, the amount of precision in both SSD and Faster R-CNN is relatively similar in confrontation with salt and pepper noise and the declination for both networks are almost with the same slope. As shown in TABLE III, the accuracy of Faster R-CNN has reduced considerably even in the lowest level of noise. For example, for the noisy samples with percent of noise 0.05, the mAP of Faster R-CNN has reduced almost 10.2%, 16.1% and 20.8% which is relatively high. Similar to SSD, Faster R-CNN do not demonstrate robustness against noise, so, the criterion is limited to the compromise between precision and speed.

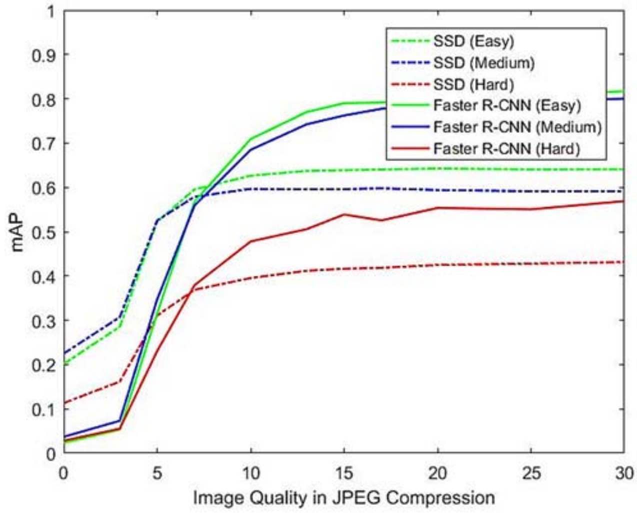


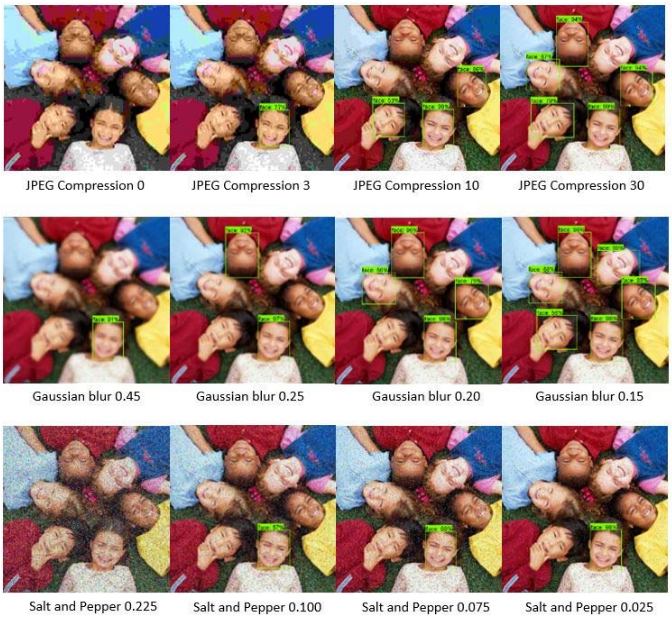
Fig. 8. The effect of JPEG compression on test sample. Quality is a value in the range [0,100] which 0 shows the highest level of compression with the lowest image quality and 100 demonstrates the highest quality image with minimum compression.

TABLE IV. PERCENTAGE OF REDUCTION IN MAP FOR JPEG COMPRESSION

Quality Level	Network	Easy	Medium	Hard
25	SSD Faster R-CNN	0% -1.1%	0% -0.4%	0% -3.2%
20	SSD Faster R-CNN	-0.3% -1.8%	-0.5% -1.6%	-1.5% -2.7%
15	SSD Faster R-CNN	-0.3% -3.3%	-0.8% -4.7%	-3.5% -5.3%
10	SSD Faster R-CNN	-2.3% -13.1%	-0.9% -14.2%	-8.4% -15.9%
5	SSD Faster R-CNN	-18.8% -60.8%	-11.4% -55.9%	-28% -59.1%
0	SSD Faster R-CNN	-69.8% -96.2%	-63.2% -94%	-74% -94.4%

The final experiment is to measure the robustness of these two networks against the effect of JPEG compression. As it is shown in Fig. 8, both networks have resisted the JPEG compression under the quality factor up to 15 which is also shown clearly in TABLE IV. However, as the quality factor decreases, the precision of networks plummet. TABLE IV shows that, for the highest level of compression (lowest quality) the mAP of SSD has reduced for the three modes of test samples 69.8%, 63.2% and 74% while the mAP of Faster R-CNN has reduced 96.2%, 94% and 94.4%, respectively. Moreover, it is shown that besides the fact that the mAP of SSD has reduced less than Faster R-CNN, the percentage of its reduction is low for the higher levels of quality factor. For example, for quality factor 10, the percentage of reduction in mAP of SSD is 2.3%, 0.9% and 8.4% which is considerably low. Hence it can be concluded that in comparison to Faster R-CNN, SSD has a higher robustness in almost all of the three modes. While an aggressive JPEG compression destroys the texture of images by integrating adjacent pixels, SSD is less sensitive to the texture. Therefore, it has a more resistance to the compression.

Fig. 9 shows some samples of the networks' output on different levels of JPEG compression, Gaussian blur and salt and pepper noise.


JPEG Compression 0 JPEG Compression 3 JPEG Compression 10 JPEG Compression 30

Gaussian blur 0.45 Gaussian blur 0.25 Gaussian blur 0.20 Gaussian blur 0.15

Salt and Pepper 0.225 Salt and Pepper 0.100 Salt and Pepper 0.075 Salt and Pepper 0.025

Fig. 9. Some samples of the network's behavior

VI. CONCLUSION

In this paper, the effect of three different image degradations has been investigated on two well-known object detection architectures: SSD and Faster R-CNN. Based on our experiments, it can be concluded that if the selection criteria of a network is based on the resistance under different artifacts, the results can be achieved as follows: Faster R-CNN is a superior choice for blurred images. SSD is a more proper architecture for JPEG compressed images. Both networks have a relatively similar resistance for salt and pepper noise.

Aforementioned experiments demonstrate that Faster R-CNN probably is more sensitive to the texture of images, while SSD is more sensitive to the edges. Therefore, destruction of edges extendedly damages SSD results while destruction of textures mostly dropped the precision of Faster R-CNN.

REFERENCES

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2014), 2014.

[2] R. Girshick, "Fast RCNN", IEEE Int. Conf. on Computer Vision (ICVV 2015), Chile, 2015.

[3] Sh. Ren, et al., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39, Issue 6, pp. 1137-1149, 2015.

[4] W. Liu and D. Anguelov, "SSD: Single Shot Multi-Box Detector," European Conf. on Computer Vision (ECCV 2016), Vol. 99, Issue 5, pp. 21-37, 2016.

[5] K. Zhang and Zh. Zhang, "Joint Face Detection and Alignment using Multitask Cascade Convolutional Networks," IEEE Signal Processing Letters, Vol. 23, Issue 10, pp. 1499-1503, 2016.

[6] R. Ranjan and V. Patel, "Hyperface: A Deep Multi-task Learning Framework for Face Detection, Landmark localization, Pose Estimation and Gender recognition," IEEE trans. on Pattern Analysis and Machine Intelligence, Vol. PP, Issue 99, pp.1-1, 2017.

[7] S. Karahan, M. K. Yildirim, K. Kirtac, F. S. Rende, G. Butun, and H. K. Ekenel, "How Image Degradations Affect Deep CNN-Based Face Recognition," 2016 International Conference of the Biometrics Special Interest Group (BIOSIG), Germany, 2016.

[8] A. Krizhevsky, I. Sutskever, and J. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," Int. Conf. on Neural Information Processing Systems, Vol. 1, pp. 1097-1105, 2012.

[9] K. Simonyan and A. Zisserman, "Very Deep Convolutional networks For Large-Scale Image Recognition," Int. Conf. on Learning Representations, San Diego, USA, 2015.

[10] Ch. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going Deeper with Convolutions," IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2015), 2015.

[11] K. Grm, V. Struc, A. Artiges, M. Caron, and H. K. Ekenel "Strength and Weaknesses of Deep Learning Models for Face Recognition against Image Degradations," IET Biometrics, Vol. 7, Issue 1, pp. 81-89, 2018.

[12] F. Iandola, S. Han, M Moskevicz, K. Ashraf, W. J. Dalley, and K. Keutzer, "SqueezeNet: AlexNet-Level Accuracy with 50x Fewer Parameters and < 0.5MB Model Size," arXive preprint arXive:1602.07360, 2016.

[13] M. Ulicny, J. Lundstorm, and S. Bayttner, "Robustness of Deep Convolutional Neural Networks for Image Recognition," International Symposium on Intelligent Computing Systems, pp. 16-30, 2016.

[14] S. Dodge and L. Karam, "Undrestanding how image quality affects deep neural networks, " Eight Int. Conference on quality of Multimedia Experience, Lisbon, Portugal, 2016.

[15] G. Goswami, N. Ratha, A. Agarwal, R. Singh, and M. Vatsa, "Unravelling Robustness of Deep Learning based Face Recognition Against Adversarial Attacks," arXive preprint arXive:1803.00401, 2018.

[16] Y. Zhuo, D. Liu and T. Huang, "Survey of face detection on low quality images," 13th Int. Conference on Automatic Face and Gesture Recgnition, Xi'an, China, 2018.

[17] S. Zhang, X. Zhu, Zh. Lei, H. Shi, X. Wang, and S. Z. Li, "Single Shot Scale-Invariant Face Detector," IEEE Int. Conference on Computer Vision, Venice, Italy, 2017.

[18] A. G. Howard, "Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications," arXiv preprint arXiv:1704.04861, 2017.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2016), 2016.

[20] J. Huang, et. al., "Speed/Accuracy trade-offs for modern convolutional object detectors," IEEE Conference on Computer Vision and Pattern Recognition, Honlulu, Hi, USA, 2017.

[21] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, "Selective search for object recognition," Int. Journal of Computer Vision (IJCV), Vol. 104, Issue 2, pp. 154-171, 2013.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial payramid pooling in deep convolutional networks for visual recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol. 37, Issue 9, 1904-1916, 2015.

[23] S. Yang, P. Luo, C. C. Loy, and X Tang, "WIDER FACE: A Face Detection Benchmark," IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016), 2016.