ALI-GOMBE, A., ELYAN, E. and ZWIEGELAAR, J. 2020. Towards a reliable face recognition system. In Iliadis, L., Angelov, P.P., Jayne, C. and Pimenidis, E. (eds.) *Proceedings of the 21st Engineering applications of neural networks conference 2020 (EANN 2020); proceedings of the EANN 2020, 5-7 June 2020, Halkidiki, Greece*. Proceedings of the International Neural Networks Society, 2. Cham: Springer [online], pages 304-316. Available from: https://doi.org/10.1007/978-3-030-48791-1_23.

Towards a reliable face recognition system.

ALI-GOMBE, A., ELYAN, E. and ZWIEGELAAR, J.

2020

Towards a Reliable Face Recognition System[⋆]

Adamu Ali-Gombe[1], Eyad Elyan[1], and Johan Zwiegelaar[2]

[1] Robert Gordon University Aberdeen, United Kingdom
[2] Mintra Group Oslo, Norway
https://www.rgu.ac.uk, https://www.mintragroup.com
{a.ali-gombe,e.elyan}@rgu.ac.uk, johan.zwiegelaar@mintragroup.com

Abstract. Face Recognition (FR) is an important area in computer vision with many applications such as security and automated border controls. The recent advancements in this domain have pushed the performance of models to human-level accuracy. However, the varying conditions in the real-world expose more challenges for their adoption. In this paper, we investigate the performance of these models. We analyze the performance of a cross-section of face detection and recognition models. Experiments were carried out without any preprocessing on three state-of-the-art face detection methods namely HOG, YOLO and MTCNN, and three recognition models namely, VGGface2, FaceNet and Arcface. Our results indicated that there is a significant reliance by these methods on preprocessing for optimum performance.

Keywords: Face Detection · Face Recognition · Deep Learning · YOLO

1 Introduction

Face detection and recognition have numerous real-world applications such as person identification and tracking. The real-world environment is typically unconstrained and has been the attention of the computer vision community for some time now. Despite exceeding human performances on test data, FR models hardly meet the requirements in the real-world [28]. Thus, preprocessing steps such as pose augmentation and illumination normalization continue to be crucial especially in mismatched conditions [16]. However, extra preprocessing steps could add delays to real-time recognition.

Majority of the established deep learning face recognition systems consist of three modules namely, a detector module, a pre-processing module and a recognition module [23], [17], [19]. Established detections model such as Viola and Jones [25], Bob [3] and fiducial detectors [23] are employed to localize the required face area before a recognition model is used. This makes the process reliant on the accuracy of the detection model. The stand-out face recognition models that reported close to or better than human performances are Deepface [17], DeepID [21], VGGFace [5], SpereFace [15], ArcFace [7], CosFace [26] and FaceNet[23].

[⋆] Supported by InnovateUK, Mintra Group and Robert Gordon University.

Although some of the results reported are close to perfect, it was discovered when testing is done at scale, these models' performances degrade considerably [11]. Moreover, these tests were carried out in controlled environments and most of these datasets were carefully curated. Furthermore, the bias in data collection such as ethnicity and race creates skewed model performances[30] [2] [1]. Again, recognition across wide age gaps is still challenging even for state-of-the-art models with near-perfect results. Other challenges include disguise or individual appearance and variations such as beard, facial expression, and others. Pictorial conditions such as illumination, pose, occlusion due to dressing (wearing a cap or eyeglasses), image quality, etc. [16] and Face spoofing [4] are all considered challenging problems to state-of-the-art FR systems.

In this paper, we perform face detection and recognition using state-of-art models and demonstrate that despite the great successes, challenges still exist in deploying these models in the real-world. Our experiments highlight these challenges and we show that without preprocessing and post-processing such as alignment, illumination normalization and frontalization, models under-performs below the reported results.

The rest of the paper is organised as follows. In Section 2, related literature is reviewed and discussed. Section 3 presents the methods used in this work. Section 4 discusses in details experimental set-up and the datasets used. Findings are discussed in section 5. Finally, we conclude and suggest future directions in Section 6.

2 Related Works

2.1 Face Detection

While face detection can be achieved using a general detection framework such as Histogram of Oriented Gradients (HOG) [6], You Look Only Once (YOLO)[18], Single Shot Detector (SSD) [14], Region Convolution Neural Network (R-CNN) [9], Max Margin Object Detection (MMOD) [12]; there are specialized face detection frameworks like Multi-Task Cascade CNN (MTCNN) [31], retina face [8] and Face Attention Networks (FAN) [27] built specifically for this purpose. Both categories have merits and the choice of a detector will depend on the application or nature of the data available. That said, specialized detectors benefit from the inclusion of ad-hoc detection pipelines with little to no overhead such as facial landmark detection that could be beneficial in post-processing. Face detection techniques such as HOG, Haar cascade are considered traditional machine learning approaches. Recent face detection techniques such as YOLO, use deep learning model or a Convolutional Neural Network (CNN) as the backbone model. The shift in trend is that fact that traditional approaches require features to be extracted before a machine learning classifier such as an SVM could be trained. Thus, features engineering reduces the generalization of these approaches. Whereas deep learning approaches learn features directly from pixel values over many training iterations thereby, generalizing better to unseen samples.

Haar cascades method [25] is one of the early successes in face detection systems and remains a popular choice. This method introduces the concept of integral images which is calculated based on region neighborhood. Similarly, HOG divides the image into cells with discrete angular bins of gradient orientations. Both are effective and fast but are affected by pose and occlusion or partial face view. These techniques are best suited for frontal faces with fewer pose effects.

Cascade CNN [13] are quite efficient in detecting faces with high visual variation such as pose and facial expressions. This approach performs detection in three different stages at different scales. A combined six CNN are used with three CNNs to determine face candidates and the other three CNNs are for bounding box calibration. Multi-Task Cascade CNN (MTCNN) is an extension of cascade CNN. While both use a cascade of CNNs, MTCNN is much faster and more accurate than the former. RetinaFace [8] added a self-supervised signal using 3D dense face regression alongside identity classification, face and facial landmark regression. According to the authors, the intuition is that since mask prediction in Mask-RCNN improved localization, then additional supervisory signal will be just as important in face localization. RetinaFace is a one-stage detector i.e faces are detected in a single go with no branches or sub-networks. Face Attention Network (FAN) [27] adds attention mechanism using a RetinaNet structure with a novel anchor assignment strategy.

Apart from generalizing better, deep learning methods enhance performances through preprocesses such as augmentation, random cropping, hard mining of samples, negative detection and others. Günther *et al.* [10] observed that on open-set detection challenge using UCCS dataset, both TinyFaces, Cascade CNN, YOLO, LBF and LgfNet performed well on face detection. The models were able to detect at least 33000 of the 36153 labeled test faces. However, the authors observed this was at the expense of high false detections. Generally, there is a trade-off between speed and accuracy when choosing a detector. Deep learning-based detectors are more accurate but are slower than traditional approaches such as HOG, but traditional approaches are less accurate. The difference in prediction time could be negligible when experimenting with few images or locally but when providing services at scale or remotely, this may be a factor to consider.

2.2 Face Recognition

Face recognition is achieved using a machine learning model by training on either engineered features or raw pixel values. A face recognition model learns an embedding function that brings together similar identities closer in the embedding space irrespective of the image conditions. Deep models in FR share a lot of commonalities and mostly use standards CNNs (such as ResNet, VGG, SENet) as their backbone. Regardless of the model used, deep learning approaches use a classifiier [5] on identification task or a distance metric when verification is the task [19].

DeepFace recognition [23] presented an improved recognition approach using 3D face alignment and frontalization technique. The facial alignment was guided by 6 fiducial points and refined by a Support Vector Regressor (SVR). DeepFace achieved identification task using a softmax and the learned model was used as a Siamese network with a chi-squared (χ^2) distance metric as the objective in a verification task. An extension of DeepFace was presented in DeepFace2 [24] which extend the process with bootstrapping (semantic bootstrapping). similarly, VGGface [5] and VGGface2 [17] were trained using softmax.

Deep IDentity features (DeepID) [21] learned identity-related features in a multi-class identification task using multiple CNNs (60). DeepID features are 160-D each and were combined with features from other networks ($160 \times 2 \times 60$). Faces were detected using fiducial detectors and the CNNs were trained on multiple face region crops. DeepID features were found to generalize well to face verification even to unseen faces. This was extended to DeepID2 [20] and DeepID2+ [22] with better network architecture, bigger hidden representations and supervision in convolution layers.

FaceNet [19] used triplet loss with Euclidean distance to train an inception model in image recognition. The approach implemented a triplet batch of two matching pairs and a non-matching sample. To choose the right pairs, FaceNet developed a novel negative exemplar mining of the most difficult triplets during training. In the Euclidean space, identical faces were held at smaller margins while different faces were pushed apart. FaceNet turned out to be highly invariant to illumination and pose on test images.

Arcface [7] utilizes an additive angular margin in obtaining highly discriminative features in face recognition. Essentially, this approaches uses centers which are determined by employing the weights of the last fully connected layer and the embedding after normalization. Extensive experiments were performed on many public datasets and the results obtained showed better performances than other existing approaches. Closely related to this are Sphereface [15] and Cosface [26].

The recent deep models use similar backbones and what differentiates them most is the training protocol. Some employ a different training function such as a softmax or additive angular margin loss or even a distance measure. All these approaches present compelling evidence on the choices made. These choices in some literature show some dependency on the task, for instance, FaceNet employed a triplet loss on their verification task which is quite logical. However, VGGface2 was trained using softmax but the model also showed comparable results on verification when the model was used as a face features and a face similarity is evaluated.

3 Methods

Three detector models considered in this paper, these are; YOLO, MTCNN and HOG. The choice of these is to compare the performance of a general-purpose detector, a specialized detector, and a mix of deep learning model and traditional machine learning models. Three face recognition models were considered namely:

VGG2faces, Arcface and Facenet, All of which are deep models. Thus, this gives us a cross-section of loss flavors that is; a VGG2face trained using softmax, an Arcface model trained using additive angular margin and a Facenet trained on triplet loss.

The first detector considered is HOG. HOG is a general detector and relies on image structure to perform detection. HOG first divides the images into local regions/grids and evaluate the gradient and orientation of pixels within these regions. Then a histogram is generated from each region. Gradients are changes in intensities along the x and y directions both of which are evaluated to be the magnitude at that pixel. The orientation is the gradients angle. An image histogram is then generated from each region/grid using these two values. Gradient normalization is usually applied to minimize the effect of illumination in the process. Equations 1 2 3 4 shows how the total gradient and orientation angle is calculated.

$$g_{x_i} = x_{(i+1)} - x_{(i-1)} \tag{1}$$

$$g_{y_i} = y_{(i+1)} - y_{(i-1)} \tag{2}$$

$$G = \sqrt{g_{x_i}^2 + g_{y_i}^2} \tag{3}$$

$$\phi = \arctan(g_{y_i}/g_{x_i}) \tag{4}$$

The second detector is YOLO which uses a CNN backbone and detect/classify objects in a single pass. This feature improves the speed of detection in real-time application. YOLO performs detection by subdividing an image into grid cells. Each grid cell outputs a bounding box, a confidence score and a class. The confidence is a measure of how accurate the model thinks an object exists within the cell. The bounding box is center of the object with width and height relative to the entire image. The cumulative loss is calculated as shown in Equation 5.

$$\begin{aligned} L = \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} l_{ij}^{obj} [(x_i - \hat{x_i})^2 + (y_i - \hat{y_i})^2] \\ + \lambda_{coord} \sum_{i=0}^{s^2} \sum_{j=0}^{B} l_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w_i}})^2 + (\sqrt{h_i} - \sqrt{\hat{h_i}})^2] \\ + \sum_{i=0}^{s^2} \sum_{j=0}^{B} l_{ij}^{obj} (C_i - \hat{C_i})^2 \\ + \lambda_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^{B} l_{ij}^{noobj} (C_i - \hat{C_i})^2 \\ + \sum_{i=0}^{s^2} l_i^{obj} \sum_{c \in classes} (p_i(c) - \hat{p_i}(c))^2 \end{aligned} \tag{5}$$

Where l_i^{obj} denotes the presence of object in cell i, l_{ij}^{obj} the jth bounding box in cell i, C is a set of classes with $p(c)$ probability, B is the set of bounding boxes, S^2 is the grids and x, y, w, h are coordinates.

Our final detector is MTCNN. This method employs online hard mining of samples to improve detection. These samples are positive face samples, negative face samples and partial faces. Detection is achieved in three-stages with three different CNNs from a coarse to fine-grained detection (P-Net, R-Net and O-Net). The first stage, P-Net, proposes candidate faces which are graded using bounding box regression and Non-Maxima Suppression (NMS) to get the high likely face candidates. The second stage is used to isolate false candidates through NMS and bounding box regression. The final stage applies supervision in learning the correct face regions. The supervision signal is a face classification and the overall loss is the sum of the Equations 6 and 7.

$$L_i^{det} = -(y_i^{det} \log(p_i) + 1 - y_i^{det}(1 - \log(p_i))) \tag{6}$$

$$l_i^{box} = ||\hat{y}_i^{box} - y_i^{box}||_2^2 \tag{7}$$

where y_i are ground truths and $p_i, \hat{y}$ are the network outputs.

3.1 Face Recognition

Different loss functions are employed in this domain that captures the similarities between image pairs or sometimes the popular probabilistic based softmax functions. The basic idea in these losses is somewhat similar but newer losses provide better parameter handling and samples combination [28]. Losses may be task-dependent, that is whether the target is an open-set or a closed-set recognition.

VGGface2 relies on a simple softmax classifier to train a ResNet for face identification task. Because of the size of the network and dataset, VGGFace learns to separate samples of different identities and brings closet samples from the same identity in the embedding space.

FaceNet uses a triplet loss to achieve face verification. The triplet loss function makes use of an anchor image x^a, positive image x^p and a negative image x^n. The loss maximizes the distance between the anchor and a negative image while minimizing the distance between the anchor and the positive sample. However, the models require the right anchor, positive, negative batch combinations for best performance. Equation 8 shows how the triplet loss is evaluated.

$$L = ||f(x_i^a) - f(x_i^p)||_2^2 + \alpha < ||f(x_i^a) - f(x_i^n)||_2^2 \tag{8}$$

Where α is a margin hyper-parameter.

Arcface uses an additive angular margin to penalizes the loss based on a geodesic distance between samples in a hyper-sphere using an arc-cosine function. This is an extension of angular softmax. Angular softmax (A-softmax) [26] adds a constraint in the hypersphere to learn better discriminative features in face recognition. A-softmax is more efficient than traditional softmax because it adopts a different decision boundary for each class. Equation 9 shows how the additive angular margin is calculated.

$$L_{arcface} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))}+\sum_{i=i,j\neq y_i}^{n}e^{s\cos\theta_j}} \tag{9}$$

Where s is the scale of the embedding and m is the margin (kept at 0.5).

4 Experiment

4.1 Datasets

Experiments were carried on two datasets namely, Wider face [29] & VGG2 [5]. Wider face is a popular benchmark for face detection in an uncontrolled environment. It contains faces with high variations in scale, pose, occlusion and illumination. The choice of the dataset is because it captures all the ideal scenarios for a face detection task in the wild. Wider face contains 32,203 images with 393,703 labeled faces. The dataset is split into a train, validation and a test set (40-10-50 split). The train set was used to train detectors and the validation set was kept as a hold out for evaluation. Results were reported on the validation set because we do not have access to the test set ground truth.

VGG2 Dataset is a large scale face recognition dataset with about 3.3m images. Images are taken in a more controlled environment but some pictures contain multiple faces, occlusion and varying light conditions. VGG2 has many samples per identity. The dataset is split across 8631 identities in the training set and 500 identities in the test set. Both of these sets are disjoint, making the dataset ideal for facial verification task. For our recognition task, the test set is kept as a hold-out for evaluation.

4.2 Experimental set-up

The detector models (HOG, MTCNN, YOLO) were trained using wider face dataset. Our HOG detector is based on the implementation in Dlib library, details can be found here [3]. Wider Face annotations were converted to XML using a python script. For MTCNN, we used a pre-trained model available at [4] which was also trained on wider face. YOLO version 3 model was trained on Wider Face following the protocol specified in [5]. Annotations were first converted to YOLO standards then, new filters and anchor boxes were evaluated before training. We used a batch size of 64 and subdivision of 16, and training was stopped when the loss remained unchanged for many iterations. In all experiments, no further preprocessing was applied to data apart from augmentation and sampling/mining techniques peculiar to the models. The models were evaluated on the test set on the number of correctly detected faces and a positive detection is considered if the IOU is over 0.4.

[3] https://github.com/davisking/dlib

[4] https://pypi.org/project/mtcnn/

[5] https://github.com/pjreddie/darknet

The recognition models (Arcface, VGGFace and FaceNet) were trained on VGG2 dataset. Prior to training, the face area was cropped out from the images using the bounding box information provided. All models were trained using a ResNet-50 backbone. The Arcface model was obtained from the authors official GitHub repository[6]. No age prediction or LFW dataset verification was employed during training. We only used a validation set for verification after 2000 batches. The training was terminated when the error rate was less than zero when the validation and training accuracies are almost the same. We trained FaceNet model using the Arcface repository but changed the loss function to a triplet loss and all other settings remain thesame. We used a pre-trained VGGface model from [7] which was trained on thesame dataset and ResNet-50 model. These models were evaluated on face crops from the test data with no further facial alignment or augmentation done. This is to give us a better understanding of the actual performance or effect of the approaches used in training the models.

Testing was carried out by generating image pairs from the test set. Using ten folds, a total of 100k pairs were generated with 50% negative matches in the pairs. The models were evaluated by measuring the True Accept Rate (TAR), False Accept Rate (FAR) and False Reject Rate (FRR). These metrics were calculated using Equations 10, 11 and 12. At test time, the models were used to extract facial embeddings from pairs. A correct match is measured using cosine similarity between these facial embeddings. A threshold of 0.5 was chosen and all faces with similarity less than or equal to the threshold are considered a match. The threshold value was chosen from repeated experimentation.

$$TAR = \frac{matches}{samplesize} \tag{10}$$

$$FAR = \frac{falseacceptance}{samplesize} \tag{11}$$

$$FRR = \frac{falserejections}{samplesize} \tag{12}$$

5 Discussion

Table 1 shows the detection performances from each model. HOG detection had the lowest false detection rate of 1.32% with YOLO and MTCNN at 8.95% and 5.04% respectively. This is not surprising given the number of detected samples. HOG detector struggled to detect face because of the varying image conditions in the dataset. As seen in Figure 1, HOG detector is affected significantly by scale, pose and occlusion.

YOLO is a general detector but shows robustness in this challenging domain. YOLO performed significantly better than HOG. From the sample detection in

[6] https://github.com/deepinsight/insightface

[7] https://github.com/WeidiXie/Keras-VGGFace2-ResNet50

Figure 1, we can see that Partial face view or partial occlusion do not affect YOLO. However, it struggles with considerable occlusion. Also, it had the worst false detection rate among the models. This may indicate that it sometimes finds it difficult to distinguish the background from faces. YOLO re-scale images in training and this is meant to improve detection of smaller objects. But we discovered that some small and blurry faces were also missed.

MTCNN detected more faces than the other detectors in this experiment. It also had a low false detection rate which demonstrates the benefits of training on negative samples. The model is also not affected by scale or partial occlusion. However, we observed that there were instances when partial faces were missed.

Generally, all the models show good IOU on the detected faces. The high average IOU returned by these models suggest reliability in these challenging circumstances. That said, none of the detectors achieved over 50% detection with IOU threshold of over 0.4.

Table 1: *Face Detection performances*

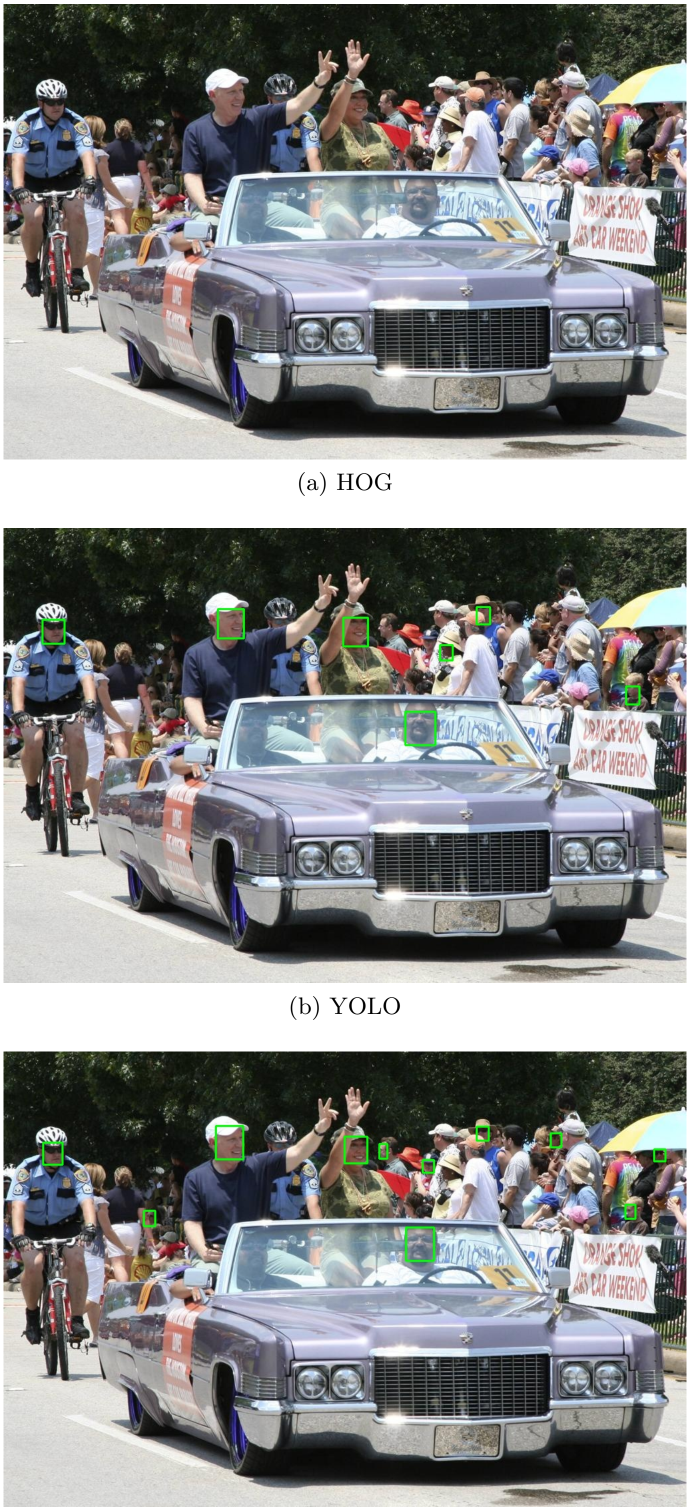
Model	Ground Truth	Detected Faces	False Detections	Average IOU
HOG	39708	5774	76	0.69
YOLO	39708	14846	1328	0.62
MTCNN	39708	17047	860	0.73

Tables 2 shows the performances of the face recognition models. All models had a very low false acceptance rate. This points to the facts that there was a clear separation of dissimilar samples by the models in the embedding space. However, the number of false rejections is significantly high. This is could be associated with the varying image conditions in the dataset used. We observed that some of the false rejection were due to pose angle and partial faces.

In this experiment, both VGGface2 and Arcface generated better embeddings than FaceNet. This shows that the two models trained using variants of softmax produced better facial features that the model trained on triplets. But this was at the expense of a slightly higher false acceptance rate. That said, the performances were generally below expectations and demonstrate the reliance of these model of preprocessing to achieve optimum performances.

Table 2: *Face recognition performances*

Model	TAR	FAR	FRR
VGGface2	86.27	0.15	13.58
FaceNet	84.97	0.13	14.90
Arcface	88.13	1.25	10.62



(a) HOG

(b) YOLO

(c) MTCNN

Fig. 1: *Sample face detection output from the three detection models*

Furthermore, one may argue that the metric or threshold value chosen could have played a part. However, when face alignment was introduced as a preprocessing step in a different experiment, the TAR increased by almost 9% across board. Thus, there is little connection between the threshold or metric and the performance. And this indicates that preprocessing continue to be significant in face recognition models.

6 Conclusion

In this paper, we analyze the performances of established face detection and recognition models. Experiments were conducted to compare models trained on a common dataset and the same recognition task. The performances of these models were evaluated using different metrics and the results indicated that optimum performance can be obtained only when extra preprocessing steps are carried out. These techniques are domain-specific and may create an overhead on the overall system and this may hinder their uses in real-time applications. This work opens a new research direction on the need for methods that rely less on preprocessing for optimum performances.

References

1. Ali-Gombe, A., Elyan, E.: Mfc-gan: class-imbalanced dataset classification using multiple fake class generative adversarial network. Neurocomputing (2019)
2. Ali-Gombe, A., Elyan, E., Jayne, C.: Multiple fake classes gan for data augmentation in face image dataset. In: 2019 International Joint Conference on Neural Networks (IJCNN)
3. Anjos, A., El-Shafey, L., Wallace, R., Günther, M., McCool, C., Marcel, S.: Bob: a free signal processing and machine learning toolbox for researchers. In: Proceedings of the 20th ACM international conference on Multimedia. ACM (2012)
4. Boulkenafet, Z., Komulainen, J., Hadid, A.: Face spoofing detection using colour texture analysis. IEEE Transactions on Information Forensics and Security (2016)
5. Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A.: Vggface2: A dataset for recognising faces across pose and age. In: 13th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2018) (2018)
6. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE computer society conference on computer vision & pattern recognition (2005)
7. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE Conference on CVPR (2019)
8. Deng, J., Guo, J., Zhou, Y., Yu, J., Kotsia, I., Zafeiriou, S.: Retinaface: Single-stage dense face localisation in the wild. arXiv preprint arXiv:1905.00641 (2019)
9. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Region-based convolutional networks for accurate object detection and segmentation. IEEE transactions on pattern analysis and machine intelligence (2015)
10. Günther, M., et al.: Unconstrained face detection and open-set face recognition challenge. In: IEEE International Joint Conference on Biometrics (IJCB) (2017)

11. Kemelmacher-Shlizerman, I., Seitz, S.M., Miller, D., Brossard, E.: The megaface benchmark: 1 million faces for recognition at scale. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
12. King, D.E.: Max-margin object detection. arXiv preprint arXiv:1502.00046 (2015)
13. Li, H., Lin, Z., Shen, X., Brandt, J., Hua, G.: A convolutional neural network cascade for face detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2015)
14. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision (2016)
15. Liu, W., Wen, Y., Yu, Z., Li, M., Raj, B., Song, L.: Sphereface: Deep hypersphere embedding for face recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2017)
16. Mehdipour Ghazi, M., Kemal Ekenel, H.: A comprehensive analysis of deep learning based representation for face recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 34–41 (2016)
17. Parkhi, O.M., Vedaldi, A., Zisserman, A., et al.: Deep face recognition. In: bmvc. vol. 1, p. 6 (2015)
18. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2016)
19. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 815–823 (2015)
20. Sun, Y., Chen, Y., Wang, X., Tang, X.: Deep learning face representation by joint identification-verification. In: Advances in neural information processing systems. pp. 1988–1996 (2014)
21. Sun, Y., Wang, X., Tang, X.: Deep learning face representation from predicting 10,000 classes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1891–1898 (2014)
22. Sun, Y., Wang, X., Tang, X.: Deeply learned face representations are sparse, selective, and robust. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2892–2900 (2015)
23. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: Closing the gap to human-level performance in face verification. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1701–1708 (2014)
24. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Web-scale training for face identification. In: Proceedings of the IEEE conference on CVPR (2015)
25. Viola, P., Jones, M.J.: Robust real-time face detection. International journal of computer vision **57**(2), 137–154 (2004)
26. Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., Liu, W.: Cosface: Large margin cosine loss for deep face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
27. Wang, J., Yuan, Y., Yu, G.: Face attention network: An effective face detector for the occluded faces. arXiv preprint arXiv:1711.07246 (2017)
28. Wang, M., Deng, W.: Deep face recognition: A survey. arXiv:1804.06655 (2018)
29. Yang, S., Luo, P., Loy, C.C., Tang, X.: Wider face: A face detection benchmark. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
30. Zeng, Y., Lu, E., Sun, Y., Tian, R.: Responsible facial recognition and beyond. arXiv preprint arXiv:1909.12935 (2019)
31. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. IEEE Signal Processing Letters (2016)