IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

# EVALUATING FACE RECOGNITION, THE CORRECT WAY

1. Mohit Gaikwad, 2. Abhilash Muneshwar , 3. Omprakash Todkar 4. Kunal Dhote

Prof, Rajini Kumari

DEPARTMENT OF ELECTRICAL ENGINEERING

K.J. EDUCATIONAL INSTITUTE TRINITY COLLEGE OF ENGINEERING & RESEARCH, PUNE,

MAHARASHTRA, INDIA

***Abstract:*** In the past few years, face detection and recognition has been the hot research topic. Previously, low-level computer vision techniques were used for detection and recognition, such as HAAR Cascade and HOG, etc, but since the availability of hardware cheaply and efficiently has paved the way for extensive research in this field, also a wide range of applications in real world ranging from biometric identification and surveillance systems to emotion analysis and augmented reality have increased the attention towards research in this field.

The emergence of deep learning techniques, particularly convolutional neural networks (CNNs), has revolutionized face detection and recognition results. Modern-day CNNs take full advantage of algorithms developed in the field of machine learning, deep learning and computer vision specifically. The availability of large-scale annotated datasets, such as the Labeled Faces in the Wild (LFW) and the CelebA dataset, has also fueled the development of more sophisticated algorithms.

Taking advantage of the currently available face detectors and feature extractors, we propose a method that uses evaluation metrics such as precision, recall and F1-score for evaluating the approach. We would say this is a traditional multi-class classification problem. We also use an in-house dataset for the evaluation process, which poses more challenges for the detector to extract faces and learn the feature representations.

We tested our proposed approach on the standard LFW dataset which performs with Precision: 90.33%, Recall: 85.26%, also the approach on the in-house dataset performs with Precision: 80.95 %, Recall: 84.19 %.

***Index Terms- MTCNN, ResNet-50, Recall, Precision, F1-score***

## 1. INTRODUCTION

Face detection and recognition are essential to many day-to-day applications, such as social media posts and filters, face unlocks features and facial expression analysis. For these systems to perform excellently and efficiently, deep learning, particularly CNN's is preferred and is supposed to work the best. Many academic and industrial researchers have centred their focus on facial recognition and facial detection. The benchmark face detection datasets commonly preferred are [1, 2, 3]. There are already many different algorithms for face detection, recognition and classification. The research in the domain has mainly evolved around CNN architectures, loss function, adding variation to datasets and training strategies.

Along with the use of CNN architectures, loss function, adding variation to datasets and best training strategies, evaluation metrics are important to correctly evaluate the model. Most face recognition approaches prefer accuracy as the evaluation metrics for their approach [4]. But for classification problems, using accuracy only is not the best approach.

For the first time, the English court announced the attempt to identify criminals by comparing images. Facial recognition has become a law enforcement technology that involves detecting video or image crime. Facial recognition tests to match the suspect's facial image are performed by forensic experts. Automatic Face Recognition Technology Increases Operator Performance and Improves Business Processes.

In our approach, we evaluate the face recognition architecture using mAP, Recall and F1-score, which are the preferred evaluation metrics for classification tasks, the latter two of which are not been correctly used in most of the cases[9], [4]. [5] used the same evaluation metrics for the face detection and recognition task. For the sake of this paper, we use MTCNN [6] and dlib library as the face detectors and ResNet50 as the feature extractor. We evaluate the performances of both detectors with features extracted by ResNet50 and further use SVM as the classifier, which classifies the given input images into their respective classes. We have gathered a custom in-house dataset for evaluation which has images from multiple views, which poses some difficulty during the feature-matching task.

Our main contributions are summarized as follows,

- We gather an in-house dataset which poses some challenges for the model to detect faces and extract features.
- We perform a series of evaluations using various face detectors and then evaluate them based on mAP, Recall and F1-score.
- We evaluate our classifier on the LFW dataset. Also, we use one in-house dataset for evaluation purposes, which is a more challenging dataset for the model.

## *2.* RELATED WORK

### *2.1 Face Detection*

Generally, face detectors work the same as object detectors. Face detection in real-world scenarios has to deal with various variation problems, including occlusion, expression, makeup, scale, pose, illumination, blur, etc.

A lot of face detection methods are proposed been proposed by researchers from top companies and individual researchers to deal with these problems, in particular, to detect small faces that vary a lot in scale, context, and anchor in order. These methods include MTCNN [6], RetinaFace [10], RefineFace [11], and the most recent ASFD [12], MaskFace [13].

### *2.2 Face Recognition*

There are n-number of face verification and recognition works proposed using lower as well as higher level computer vision. In this paper, we briefly try to discuss the most relevant recent work. The works of [14, 15, 16] all of which employ a complex system over multiple stages, which uses the output of a deep convolutional network with PCA for dimensionality reduction and an SVM for classification purposes.

FaceNet[4] developed by Google researchers, utilizes machine learning to improve facial recognition. FaceNet is designed to directly train facial models using Euclidean space, which measures the similarities between different faces as distances. This approach helps to improve the accuracy of facial recognition.

## 3.1 WORKING PROCEDURE

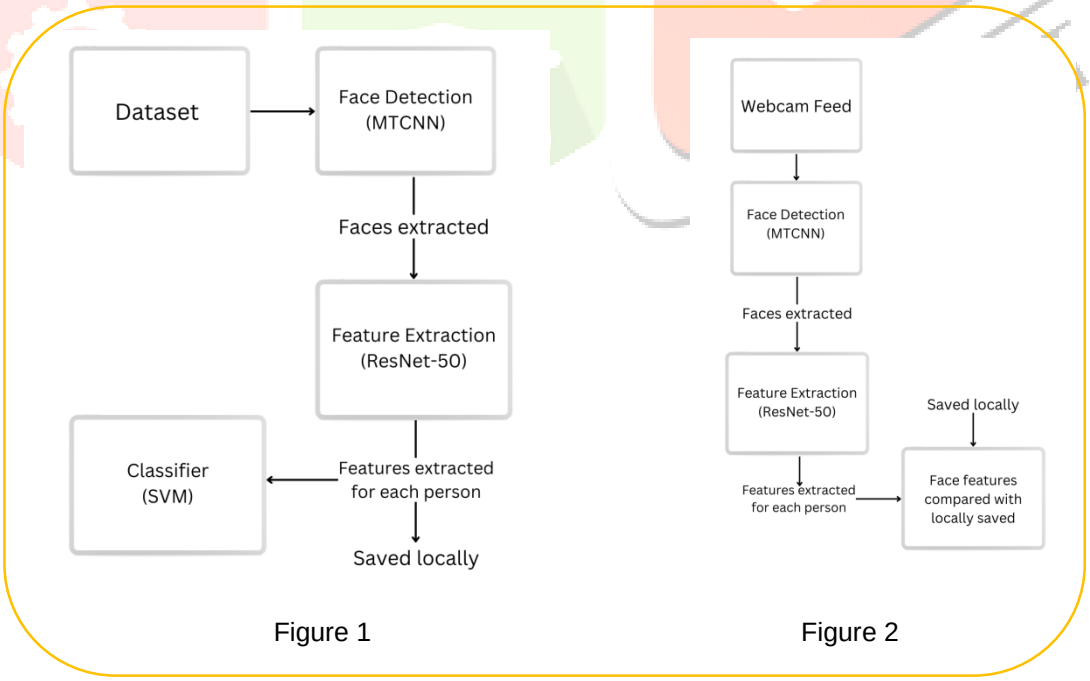


Figure 1 Figure 2

### 3.1 Model experimentation:

The proposed methodology is as mentioned in Figure 1. For experimentation purposes, we started with three face detectors i.e Yolo5 Face [5], MTCNN [6] and DLIB, and for the sake of this research documentation, we move forward with MTCNN and DLIB which were the best-performing ones. ResNet-50 is used as a feature extractor which has repeatedly proven to perform well in general cases. We have used an SVM classifier to classify the various classes/ persons and evaluate the model performance on the LFW and in-house datasets.

In the case of real-time implementation, we save the features extracted during the SVM classifier training process along with their class ids. These saved features are used for comparison with the newly extracted features during real-time implementation.

### 3.2 Real-time Implementation

For the sake of implementation, we use a slightly different approach for classifying or recognizing faces.

At first, we take a live feed from the webcam, in our case, Logitech c925e. The MTCNN face detector detects and extracts faces from the incoming video stream.

**MTCNN**

The sequential workflow of MTCNN involves face detection, followed by facial landmark localization and face alignment.

1. **Face Detection**: The face detection stage uses a CNN called Proposal Network (P-Net) that generates candidate face regions (bounding boxes) in an image. This is done by sliding a small window across the image, Each window classifies whether a face is present or not in the window patch.

2. **Facial Landmark Localization:** The second stage i.e. facial landmark localization, uses an even deeper CNN compared to P-Net called the Refine Network (R-Net) that refines and filters the candidate face regions generated by the P-Net. This helps in predicting more accurate bounding box coordinates and also estimates the facial landmarks, such as the positions of eyes, nose, and mouth.

3. **Face Alignment:** In the final stage, face alignment is performed by the third CNN called the Output Network (O-Net), this network further refines the bounding box coordinates and facial landmark positions. The network also computes an affine transformation to align the face, which makes it suitable for recognition tasks.

The extracted faces are used for extracting features that are ideally supposed to be unique for different faces, but the uniqueness and quality of features extracted from images are highly dependent upon lighting conditions, background noise, etc. In this work we ResNet50 as the feature extractor, which has proven to generalize well on unseen data. These features are 512-dimensional arrays that represent the details of the face.

**ResNet-50**

ResNet-50 convolutional neural network architecture belongs to the ResNet (Residual Network) family. The ‘50’ in its name refers to the number of layers in the network, which includes convolutional layers, pooling layers, fully connected layers, and shortcut connections. The bottleneck block is the basic building block of ResNet-50, which comprises three convolutional layers: 1x1, 3x3, and 1x1. This combination helps in reducing the computational cost while maintaining high accuracy.

The features extracted with ResNet-50 from faces extracted with MTCNN are compared with the features that were saved from multiple images for each person/ class. By comparison, we mean calculating the cosine distance between the features from the image captured from the webcam and the previously saved features. The distance should ideally be less for similar faces and greater for different faces.

**Cosine Distance**

Cosine Distance measures the similarity or dissimilarity between the two given vectors in a high-dimensional space. It calculates the angle between the vectors instead of their absolute magnitudes.

$$cosine\ similarity = (A \,.\, B) / (||A|| * ||B||) \quad (1)$$

Where A and B are two vectors, the cosine distance is calculated considering the cosine of the angle between the two vectors and ‘.’ means dot product.

$$cosine\_distance = 1 - cosine\_similarity \quad (2)$$

The cosine similarity ranges from -1 to 1. A value near 1 signifies significant similarity, while a value nearing -1 suggests considerable dissimilarity. A cosine similarity of 0 means the vectors are orthogonal or completely dissimilar.

## 4. DATASET

We have evaluated our method with two datasets and i.e. Labelled Faces in the Wild and in-house dataset, we evaluate our method for the face recognition task.
The LFW dataset contains 5425 images from 311 classes, which is a standard benchmark dataset. The in-house dataset contains 120 images from 12 classes. The LFW dataset is a comparatively more challenging dataset that helps us understand the robustness of the model and also helps us understand how well the model has generalized.

## 5. EXPERIMENTS

Employing a face detector on every image generates tightly fitting bounding boxes around the facial regions. For experimentation purposes, DLib and MTCNN detectors were preferred over other existing face detectors. Resnet50 model is used for feature extraction purposes, which leverages the weights that are already trained with vgg face2 dataset on almost 3.31 million images from about 9131 different identities.

We use an SVM classifier to classify the images into their respective classes in the model experimentation part and calculate the cosine distance between the features during the real-time implementation. We have used precision, recall, and F1-score to evaluate the SVM classifier performance that takes feature embedding from Resnet50 as inputs.

For the sake of experimentation, we have experimented with three face detectors, out of which finally two detectors were further considered based on their performances for final experiments.

### 5.1 Evaluation with DLib as face detector

DLib is a face detection framework that has proven to perform well in various conditions. Dlib includes two face detection methods built into the framework:

**HOG + Linear SVM face detector:** which is accurate and can be preferred in applications that need the overall system to be computationally efficient.

· **Max-Margin (MMOD) face detector**: due to its robustness and high accuracy, detecting faces from varying viewing angles, under different lighting conditions, and occlusion are some more things the detector is capable of.

| Dataset | Precision | Recall | F1-score | Accuracy |
|---|---|---|---|---|
| LFW Dataset | 81.73 % | 75.6 % | 76.55 % | 83 % |
| In-House Dataset | 80.75 % | 79.13 % | 77.14 % | 82 % |

Table 1

### 5.2 Evaluation on MTCNN as face detector

MTCNN[6] is a cascaded structure with three stages of deep convolutional networks that is designed to predict the face bounding boxes and landmark location in a image. Leveraging the advantages of multi-task learning, MTCNN has proved through authors experiments and our experiments that it does prove to predict highly accurate bounding boxes.

| Dataset | Precision | Recall | F1-score | Accuracy |
|---|---|---|---|---|
| LFW Dataset | 90.33 % | 85.26 % | 86.47 % | 90 % |
| In-House Dataset | 80.95 % | 84.19 % | 82.24 % | 89 % |

Table 2


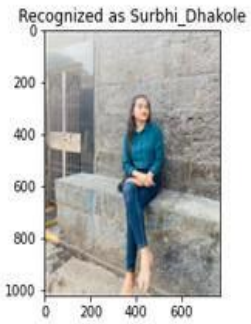


Image 1



Image 2

Image 1 and Image 2 show the classification of faces into their true classes using DLIB and MTCNN detectors resp. This gives a gist of how well the face detectors and feature extractors perform even when the proportion of faces in the image is low as compared to the background and other body parts.

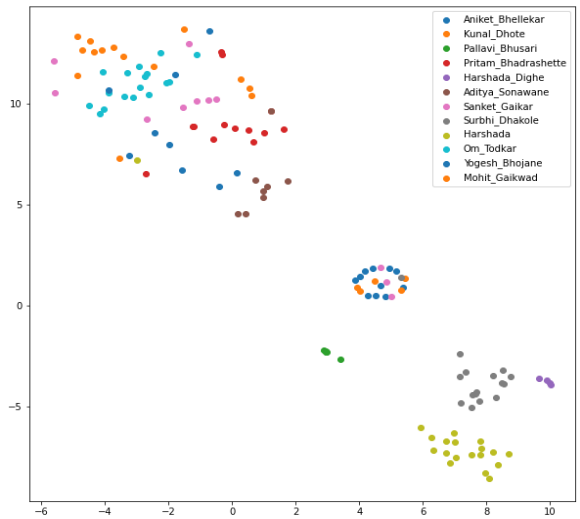


Image 3

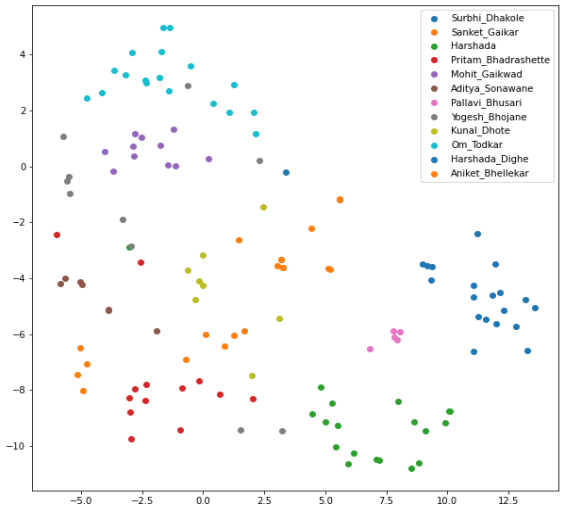


Image 4

Image 3 and Image 4 show the plot of feature embedding for our in-house dataset. Image 3 shows the plot for feature embedding for faces detected with the DLib detector and Image 4 for faces detected with the MTCNN detector. This also proves that MTCNN outputs better faces bounding boxes that ultimately help in better feature extraction. This plot is helpful in understanding the performance of the face detector as the feature extractor is the same in both cases.

## 6. CONCLUSION

The proposed method performs well up to the mark, mainly the MTCNN face detectors help in achieving good feature representations. This proposed method performs well in real-time as well. This approach can be implemented for various applications in the real world and is expected to give good results as the approach has already been tested on the in-house dataset, which is already a challenging dataset.

## 7. REFERENCES

[1] Z. Zhu, P. Luo, X. Wang, and X. Tang. Recover canonical view faces in the wild with deep neural networks. *CoRR*, abs/1404.3543, 2014.
[2] S. Yang, P. Luo, C. C. Loy, and X. Tang, "Wider face: A face detection benchmark," *http://shuoyang1213.me/WIDERFACE/index.html*.
[3] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labelled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007. 5
[4] Schroff, Florian, et al. "FaceNet: A Unified Embedding for Face Recognition and Clustering." *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–23. *arXiv.org*, https://doi.org/10.1109/CVPR.2015.7298682..
[5] Qi, Delong, et al. *YOLO5Face: Why Reinventing a Face Detector*. arXiv, 27 Jan. 2022. *arXiv.org*, http://arxiv.org/abs/2105.12931.
[6] Zhang, Kaipeng, et al. "Joint Face Detection and Alignment Using Multi-Task Cascaded Convolutional Networks." *IEEE Signal Processing Letters*, vol. 23, no. 10, Oct. 2016, pp. 1499–503. *arXiv.org*, https://doi.org/10.1109/LSP.2016.2603342.
[7] He, Kaiming, et al. *Deep Residual Learning for Image Recognition*. arXiv, 10 Dec. 2015. *arXiv.org*, http://arxiv.org/abs/1512.03385.
[8] King, Davis E. *Max-Margin Object Detection*. arXiv, 30 Jan. 2015. *arXiv.org*, http://arxiv.org/abs/1502.00046.
[9] J. Deng, J. Guo, Y. Zhou, J. Yu, I. Kotsia, and S. Zafeiriou, "Retinaface: Single-stage dense face localisation in the wild," *CVPR*, 2020.
[10] S. Zhang, C. Chi, Z. Lei, and S.Z. Li, "Refineface: Refinement neural network for high performance face detection," *ArXiv preprint 1909.04376*, 2019.
[11] B. Zhang, J. Li and Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, Y. Xia, W. Pei, and R. Ji, "Automatic and scalable face detector," *ArXiv preprint 2003.11228*, 2020.
[12] [13] D. Yashunin, T. Baydasov, and R. Vlasov, "Maskface: multi-task face and landmark detector," *ArXiv preprint 2005.09412*, 2020.
[13] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. *CoRR*, abs/1412.1265, 2014.
[14] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *IEEE Conf. on CVPR*, 2014