ISSN 2278-3091
Volume 10, No.2, March - April 2021
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse341022021.pdf
https://doi.org/10.30534/ijatcse/2021/341022021

Face Recognition Techniques: A Survey

Yasir Niaz Khan[1], Ali Ahmad[2], Muhammad Usama[3]
[1]The University of Lahore, Lahore, Pakistan, yasir.niaz@gmail.com
[2]The University of Lahore, Lahore, Pakistan,alijakhar129@gmail.com
[3]The University of Lahore, Lahore, Pakistan, usamabhatti4061@gmail.com

ABSTRACT

A lot of research work and development is taking place in being carried in field of face recognition, now a days. A face recognition process has two pillars: face detection and face recognition. A number of techniques are being used these purposes. The accuracy of all those techniques vary and separate techniques for detection and recognition are in practice at present. In this paper we will give an insight to accuracies of different face detection and recognition techniques which are being widely used by researchers and developers.

Key words: Face Recognition. Python, Image Processing, Face Detection, Deep Learning.

1. INTRODUCTION

Face recognition is a rapidly developing and widely applied aspect of biometric technologies. Its applications are broad, ranging from law enforcement to consumer applications, and industry efficiency and monitoring solutions. Facial attendance system is being widely used. Due to COVID threat, biometric attendance has been banned and Face recognition system have emerged as alternate. A number of researches have been carried out in past few years. Long ago, researchers used to face issue of having low computational power. With the advancement in technologies and availability of high processing power, dealing with huge datasets and trying different approaches became very easy. Google and Amazon have made cheap virtual systems available online. This led to a revolution in research of face recognition technologies. A number of face detection and recognition techniques have been published in different research papers over the past few years. We went through many of those research papers and explored those state-of-the-art techniques. We got an idea about some efficient ways to detect and identify the faces in an image. This paper is intended to let the user have an idea of how accurate these techniques are on a specific dataset. Face recognition has never been an easy task to accomplish. Like many other machine learning projects, availability of dataset is also a challenge in these cases. Good and labeled faces datasets are hard to come by.

2. DATASETS

All facial recognition and detection systems require face datasets for training and testing purposes. It plays a pivotal role in overall performance of the system. In our paper, we used two datasets so that we can check the variation in accuracy and draw any conclusion. **F**irst dataset was of our fellow student. We recorded short 180 Degree videos of our classmates. We extracted frame from those videos and then compiled a dataset. **S**econd dataset was of Indian celebrities. We downloaded this dataset from an online source. This dataset comprised of 229 persons. We selected this dataset keeping in view the ease with which images of Indian celebrities are available online.

No.	Dataset	Total subjects	Total images
1	Bollywood celebrity faces	229	9,329
2	Student faces	43	1,021

Table1

Fig. 1: Student and celebrity datasets

3. FACE DETECTION USING HAAR CASCADE:

Object detection using Haar-Feature based cascade classifier is an effective method proposed by Viola and Michael Jones in 2001. It is a machine learning based approach in which a cascade function trained from lot of positive and negative images [1]. We used frontal face cascade classifier to detect faces. This cascade classifier was trained to detect the frontal face only so it failed terribly to detect face region when there was even a slight change in posture of face. And below are the results:

Fig. 2: Haar-cascade face detection sample

Dataset	Total detected Faces	Correctly detected faces	Not detected faces	False detected Faces
Bollywood celebrity faces	8,222	7,851	1,107	371
Student faces	903	892	118	11

Table 2: Haar-cascade face detection result

Bollywood celebrity faces		Student faces	
TP=7851	FN=1107	TP=892	FN=105
FP=371	TN=0	FP=11	TN=13
Accuracy=84.2%		Accuracy=88.6%	

Table 3: Haar-cascade confusion matrix

4. FACE DETECTION USING FACE-RECOGNITION.LOCATION:

Face-recognition package includes a location function which detects the face region in an image. There are two methods to detect face using Face-recognition package's function.location. First method is Hogg method and second one is CNN method. We have to specify method in parameter and by default, it is set to Hogg Method. We detected using Hogg method because CNN method requires high processing power. CNN is accurate but slow and HOG is fast but less accurate as compared to CNN.

Fig. 3: Face-recognition.location correctly face detection sample

Fig. 4: Face-recognition.location false face detection sample

Dataset	Total detected Faces	Correctly detected faces	Not detected faces	False detected Faces
Bollywood celebrity faces	9,302	9,252	27	50
Student faces	987	987	34	0

Table 4: Face-recognition.location detection result

Bollywood celebrity faces		Student faces	
TP=9252	FN=27	TP=987	FN=21
FP=50	TN=0	FP=0	TN=13
Accuracy=99.2%		Accuracy=97.9%	

Table 5: Face-recognition.location confusion matrix

5. FACE DETECTION USING MTCNN:

MTCNN is a python (pip) library written by Github user Ipacz which implement the paper Zhang, Kaipeng et al. "Joint Face Detection and Alignment Using Multitask Cascade Convolution Networks". IEEE Single Processing Letters 23.10 (2016) 1499:1503 Crossref Web MTCNN runs fast on CPU [2].

Fig. 5: MTCNN correctly face detection sample

Fig. 6: MTCNN false face detection sample

Dataset	Total detected Faces	Correctly detected faces	Not detected faces	False detected Faces
Bollywood celebrity faces	8,830	8,195	499	635
Student faces	963	938	25	58

Table 6: MTCNN face detection result

Bollywood celebrity faces		Student faces	
TP=8195	FN=499	TP=938	FN=45
FP=635	TN=0	FP=25	TN=13
Accuracy=87.8%		Accuracy=93.1%	

Table 7: MTCNN confusion matrix

6. FACE RECOGNITION USING LBPH:

Local Binary Pattern (LBPH) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number. It was first described in 1994 (LBP) and has since been found to be a powerful feature for texture classification. It has further been determined that when LBP is combined with histograms of oriented gradients (HOG) descriptor, it improves the detection

performance considerably on some datasets [5]. This algorithm gives terrible results with the increment in dataset. The Moment we increase the dataset, its accuracy declines significantly. Difference in light intensity and posture of test and training image is also an aspect which effects its accuracy. This algorithm performs very well on small dataset. In our observation we have noticed it accuracy declines after number of persons increase from 5.A research paper based on LPBH has termed it a highly accurate algorithm but that researched had tested only 4-person dataset [6]. A few research papers have used LPBH to recognize slightly more than 5 persons [7] but in that case, the dataset of each person was extremely large. Training such a huge dataset is not feasible. On the other hand, other techniques give higher accuracy with very small dataset of each person.

7. FACE RECOGNITION USING FACE-RECOGNITION MODULE:

In this technique, we used face locations function to locate the face region and face recognition function to identify/recognize the face. Both of these functions are from Face-recognition python package [3]. We used Bollywood celebrity's dataset in this method. While implementing this technique, we had set face distance threshold at less than 50 [4]. After implemented this technique, we got following results:

Bollywood celebrity faces	Student faces
TP=2747 FN=11 FP=63 TN=11	TP=20 FN=0 FP=2 TN=28
Accuracy=97.4%	Accuracy=96%
Error-Rate=2.6%	Error-Rate=4%
Specificity= 14.9%	Specificity=93.3%
FP-Rate=85.1%	FP-Rate=6.67%
Recall=99.6%	Recall=100%
Precision =97.8%	Precision =90.9%

Table 8: Results recognition with face-recognition. distance less than 50

Bollywood celebrity faces	Student faces
TP=2685 FN=86 FP=31 TN=30	TP=19 FN=2 FP=1 TN=28
Accuracy=95.9%	Accuracy=94%
Error-Rate=4.1%	Error-Rate=6%
Specificity= 49.1%	Specificity=96.5%
FP-Rate=50.8%	FP-Rate=3.4%
Recall=96.9%	Recall=90.5%
Precision =98.9%	Precision =95%

Table 9: Results recognition with face-recognition. distance less than 45

8. FACE RECOGNITION USING FACE-RECOGNITION MODULE AND DETECTION USING MTCNN:

Bollywood celebrity faces	Student faces
TP=2116 FN=63 FP=12 TN=20	TP=18 FN=2 FP=1 TN=25
Accuracy=96.6%	Accuracy=93.5%
Error-Rate=3.3%	Error-Rate=6.5%
Specificity= 68.8%	Specificity=96.2%
FP-Rate=37.5%	FP-Rate=3.8%
Recall=97.1%	Recall=90%
Precision=99.4%	Precision =94.7%

Table 10: Results recognition with face-recognition. distance less than 45

9. CONCLUSION

No model is 100% accurate in the world of image processing and machine learning. We always tend to gain maximum accuracy by making changes in our model. Errors are of two type: False Positive and False Negative. Now it varies from system to system which category of error can be more serious for that system. For example, even if my system labels recognized persons as unknown ones, but never lets a unknown person labeled as a known person, then I can compromise FN to some extent. In previous case, FN was only 11 and, in this case, FN raised to 86. But at the same time, there is a significant decline in number of FP cases and improvement in TN. To let you have a better understanding of it, here is the elaboration of it: TP: Known Person labeled as known TN: Unknown Person labeled as Unknown FP: Unknown Person labeled as Known FN: Known Person labeled as Unknown In our case, FP cannot be tolerated as if system keeps on labeling the unknown persons as known ones then the purpose of system fails. So we decided to afford a rise in FN but we had to bring down FN. So, we changed the distance threshold from 50 to 45. By changing the threshold, we got some decline in accuracy but it helped us to deal with FP's. Face recognition module works by matching encoding scheme. It matches the encoding o the test image with encoding of all the images in record and labels the test image as that image of record whose encodings are closest to that image. This gives us a good accuracy but at the same time, this phenomenon leads to a high FP rate. While working on face recognition systems, we tend to have Low FP rate and high TN rate. As we have gone through many techniques, we have concluded that best combination of detection and recognition techniques is face detection using MTCNN and recognition using Face-recognition module.

10. ACKNOWLEDGEMENT

We acknowledge the contribution made by all the researchers who contributed by publishing research papers on which these widely used techniques are based.

REFERENCES

[1] https://docs.opencv.org/3.4/db/d28/tutorial_casca %20de_classifier.html
[2] https://towardsdatascience.com/face-detection-% 20using-mtcnn-a-guide-for-face-extraction-with-a-%20focus-on-speed-c6d59f82d49
[3] https://github.com/ageitgey/face_recognition
[4] https://face-recognition.readthedocs.io/en/latest/face_recogni tion.html
[5] https://towardsdatascience.com/face-recognition-%20how-lbph-works-90ec258c3d6b
[6] https://journals.sagepub.com/doi/full/10.1177/00 %2020294020932344s
[7] https://www.ijrte.org/wp-%20content/uploads/papers/v8i2/B2015078219.pdf