International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 06 | June 2020 www.irjet.net p-ISSN: 2395-0072

ID-SELFIE MATCHING

Reema Malve[1], Ruchita Motwani[2], Hrutuja Sansare[3], Naveen Vaswani[4]

[1]Student, Dept. Of Computer Engineering, WIEECT College, Maharashtra, India
[2]Student, Dept. Of Computer Engineering, WIEECT College, Maharashtra, India
[3]Student, Dept. Of Computer Engineering, WIEECT College, Maharashtra, India
[4]Asst. Professor, Dept. Of Computer Engineering, WIEECT College, Maharashtra, India

---

Abstract - *Technology has been growing at a very fast pace as the world is moving towards a digital world. Identity verification plays a very important role whether you want to open a bank account or want to travel from one place to other. The aim of this paper is to develop a system which is going to verify a person using face recognition techniques. The system would be using MTCNN technique for image alignment, face detection and feature extraction. It would be trained using python libraries such as tensorflow, keras with triplet loss as an error function. The verification steps are 1) Take scanned image of ID document(pan card) and a live selfie of the person as input 2) Do preprocessing and generate embeddings which are compared using cosine similarity, 3) Display the verification status as the output. This system can overcome the flaws of manual process for identity verification as such as time consuming, unreliable, etc.*

Key Words: **MTCNN, Embeddings, Tensorflow, Keras, Cosine similarity, Triplet loss function**

1. INTRODUCTION

The traditional approach for verifying person's identity involves manual checking of ID cards. In this case it is possible that the person with similar looks can bypass the verification system. This may lead to many frauds since unauthorized user will get access to the system. When it comes to banking sector it is very important that the accurate verification should be done in order to protect users' data and privacy to avoid any fraud. We have developed a system which will accurately identify the person using his/her ID photo and the person's live photo. The system will use face recognition techniques to match the two pictures. At first face from live photo and ID face image will be detected and it will be pre-processed to extract some important face features which will be used for matching task. Matching would be done using Cosine Similarity and if this distance is less than the threshold (defined by the user) then matching would be successful (both pictures belong to same person) otherwise it would fail. This system is mainly useful for banking sector. They can use this system for verifying their customers and improve the security standards. This project has wide scope in many sectors like in civil applications and law enforcement, security application in electronic transactions to list a few.

2. LITERATURE REVIEW

Various methods are available for face detection and extraction. Hidden Markov Model (HMM)[2], Landmark localization[3], Geometric Approach[2] and variations of neural network[2][4][5][[6] are few of them. The Hidden Markov Model (HMM) produces individual models for each face in the dataset [2]. Hence we don't have to adjust the model for every new face. Although geometric approach is a historic way to recognize people, the image processing involved is very expensive [2]. Among all the available approaches for face detection Multi-Task Convolutional Neural Net- work (MTCNN) is very efficient. A deep cascaded multitask framework exploits the inherent correlation between face detection and alignment [6]. They [6] have used three convolutional networks namely P-net (Proposed network), R- net (Refinement Network) and O- net (Output Network) for face detection and alignment [6]. The output of this MTCNN is a dictionary with three keys bounding box, confidence and key- points. The key points are used for training the network along with the extracted features. Triplet Loss function is used as a loss function. Whereas for the final matching of Images Cosine Similarity can be used [8][1].

3. DESCRIPTION

Id-Selfie matching system is used to verify person's identity by using face recognition techniques. Here, a system is constructed which can efficiently authenticate the user. A live image (selfie) of a person is taken and it is compared with the photo on their ID, provided by them at the time of verification process. There are many issues involved in the process of face recognition.

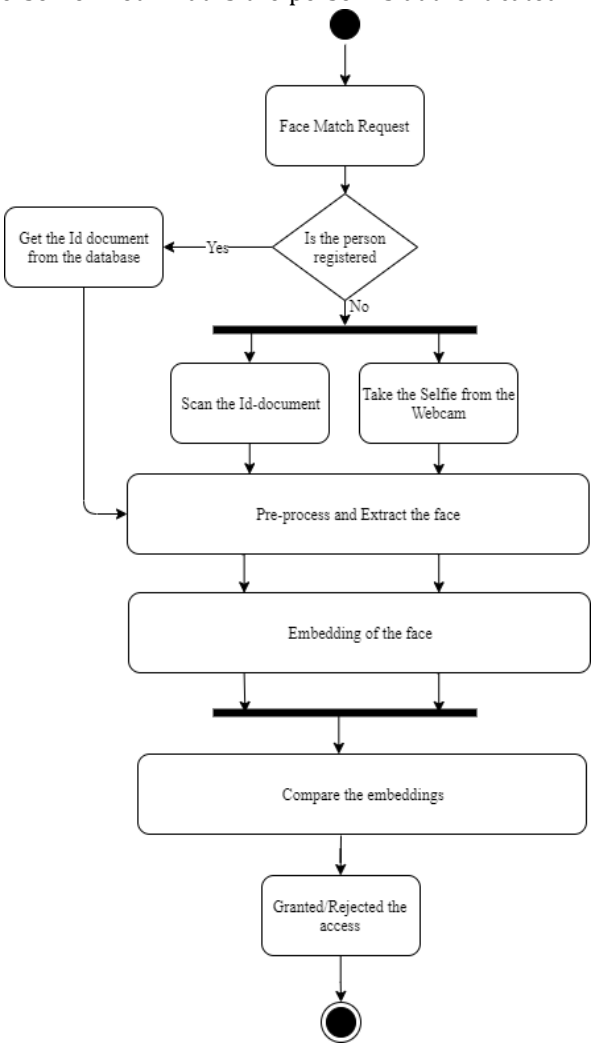
Factors Affecting Face Recognition:

- Facial Expression
- Partial Occlusion
- Pose Variation
- Facial Ageing

Steps involved in matching:

1. Take Images: The ID image and Selfie image are obtained from the user using appropriate platform (web cam and scanner). These images are sent to MTCNN for face extraction.
2. Pre-processing and Extraction of Face: In this step, image is aligned in preprocessing and face is detected and extracted using MTCNN. Output of MTCNN is face matrix of size 128 with the confidence and bounding box.

3. Getting The Embeddings: Here, we use weights obtained from previously trained model and CNN to get embeddings for the face matrix.
4. Comparing the Embeddings: In this step, embeddings obtained from the ID image and Selfie image are compared using cosine similarity method.
5. Grant Access: The results obtained from the comparison of embeddings will decide whether to grant access to the person or not. That is the person is authenticated. The



activity diagram of our model is shown in Fig -1.

Fig -1 Activity Diagram

4. IMPLEMENTATION

The proposed system is implemented using different techniques for extracting face, getting embeddings and comparing the final results to authenticate the users. The important techniques used here are discusses below:

1. Detection and Extraction of Face: As mentioned before we have used MTCNN to extract faces from both selfie image and ID image. The architecture of MTCNN is shown in Fig -2.

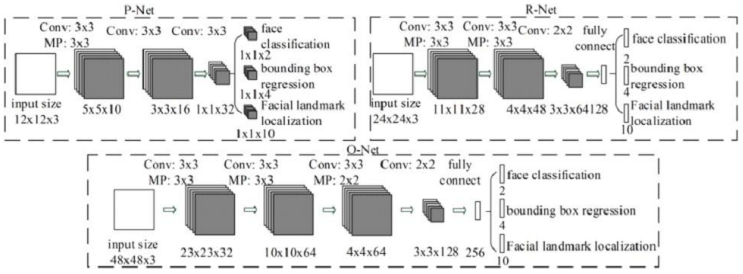


Fig -2 Architecture of MTCNN[6]

2. Training Model: Training plays very important role in improving the accuracy of the model. We have used datasets listed below to train and test the model. Model training is done using two methods: I) using facnet model. II) using CNN architecture. Model training sets the weights for neural network which are used further to obtain embeddings from face image.
3. Embeddings: In the context of neural networks, embeddings are low-dimensional, learned continuous vector representations of discrete variables [11]. They are useful because they can reduce the dimensionality of categorical variables and meaningfully represent categories in the transformed space. Here, we use the weights calculated during training for finding embeddings for the face which are further used for comparison.
4. Cosine Similarity: Cosine similarity is used forcomparing the embeddings obtained from both the images (ID and Selfie images).It is calculated using following formula: The similarity value obtained from above formula is compared with the threshold value and if the similarity is more than threshold, then two images are matched successfully and it authenticates the user.

4.1 Datasets

1. MS-Celeb-1M Dataset: The MS-Celeb-1M dataset [10] is a public domain face dataset facilitating training of deep networks for face recognition. It contains 8, 456, 240 images of 99, 892 subjects (mostly celebrities) downloaded from internet. However, the dataset is known to have many mislabels. We use a cleaned version of MS-Celeb-1M with 5, 041, 527 images of 98, 687.
2. Private ID-Selfie Dataset: It is a private dataset that consist of 500 Id data along with at least one selfie per person. Document used for verification here is Pancard which are scanned while the selfie are taken using the camera.

4.2 Training

The CNN architecture used here is variant of the NN4 architecture and identified as nn4.small2 model in the OpenFace project. This architecture consists of fully connected layer with 128 hidden units followed by an L2 normalization layer on top of the convolutional base. These

two top layers are referred to as the embedding layer from which the 128-dimensional embedding vectors can be obtained. This model learn an embedding f(x) of image x such that the squared L2 distance between all faces of the same identity is small and the distance between a pair of faces from different identities is large which is achieved using triplet loss.

Triplet Loss: It is minimum when the distance between an anchor image x^a and a positive image x^p (same identity) in embedding space is smaller than the distance between that anchor image and a negative image x^n (different identity) by at le Σ

$$L = [||f(x^a) - f(x^p)||^2 - ||f(x^a) - f(x^n)||^2 + \alpha] + [z]_+$$

Means max (z, 0) and m is the number of triplets in the training set [12].

4.3 Software and Hardware

Training of data for the Id-Selfie system would be done using TensorFlow which is the computational framework used to build machine learning models and Keras which is the open source neural network that works on the top of TensorFlow. Hardware required for this system would be a web camera for taking selfie.

Following are the requirements for a system to run this application:

Software requirements:

- Python
- Python Libraries: Keras, TensorFlow, numpy, OpenCV

Hardware requirements:

- Min processor: Intel i3
- Min RAM: 4GB
- OS: Windows/Macintosh

5. RESULT

We have trained our model using the databases mentioned in the previous section. After successful training we have tested the ID-Selfie model against 336 samples and we observed the accuracy of this model as 73.51%.

Fig -3 ID-selfie pair with 55.18%

Fig -4 Not-matched ID-selfie pair

6. CONCLUSIONS

In this paper, we have proposed an ID - Selfie matching system for authentication purpose. We have trained and tested the system using different methods as mentioned previously. After all the training and testing we have reached the accuracy of 73.52% and arrived at a conclusion that the system is able to match blur images and the rotated images more efficiently. Whereas it shows poor result for other factors such as facial expression, large age gap. More work need to be done for overcoming these factors.

REFERENCES

[1] Yichun Shi and Anil K. Jain, "DocFace+: IDDocument to Selfie Matching", IEEE Transactions on Biometrics, Behavior, and Identity Science, Vol. 1 , Issue. 1, pp. 56-67, Jan. 2019.

[2] Selvapriya. M and Dr. J. KomalaLakshmi, "Face Recognition Using Image Processing Techniques: A Survey", International Journal Of Engineering And Computer Science ISSN: 2319-7242, Volume 3 Issue 12 December 2014, Page No. 9704-9711.

[3] Prathap Nair, and Andrea Cavallaro, "3-D Face Detection, Landmark Localization, and Registration Using a Point Distribution Model", IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 11, NO. 4, JUNE 2009.

[4] Rakesh Rathi (Ph.D.), Manish Choudhary (M.Tech.) and Bhuwan Chandra (Ph.D.), "An Application of Face Recognition System using Image Processing and Neural Networks", JAN-FEB 2012, Vol 3 (1), 45-49, ISSN: 2229-6093.

[5] Patrik KAMENCAY, Miroslav BENCO, Tomas MIZ- DOS, Roman RADIL, "A New Method for Face Recognition Using Convolutional Neural Network", DIGITAL IMAGE PROCESSING AND COMPUTER GRAPHICS, VOLUME: 15 — NUMBER: 4 — 2017— SPECIAL ISSUE.

[6] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao, "Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks".

[7] Feng Wang, Weiyang Liu, Haijun Liu, Jian Cheng, "Additive Margin Softmax for Face Verification", arXiv:1801.05599v4 [cs.CV] 30 May 2018.

[8] Valery Starovoitov, Dmitry Samal and Bulent Sankur, "MATCHING OF FACES IN CAMERA IMAGES AND DOCUMENT PHOTOGRAPHS".

[9] L. L. Gayani Kumari, "Age progression for elderly people using image morphing", 2011 International Conference on Advances in ICT for Emerging Regions (ICTer), Sept. 2011.

[10] Yandong Guo and Lei Zhang and Yuxiao Hu and Xiaodong He and Jianfeng Gao, "MS-Celeb-1M: ADataset and Benchmark for Large-Scale Face Recognition", 2016.

[11] Koehrsen, Will. "Neural Network Embeddings Explained." *Towards Data Science,* 02 Oct. 2018, www.towardsdatascience.com/neural-network-embeddings-explained-4d028e6f0526. Accessed 23 June 2020.

[12] *Jupyter Notebook Viewer,* www.nbviewer.jupyter.org/github/krasserm/face-recognition/blob/master/face-recognition.ipynb?flush_cache=true. Accessed 23 June 2020.