

Online Credit System using Face Recognition

Vaibhav Ambasta

Student, Department of Computer
Science and Engineering
B.M.S. College of Engineering
Bangalore, India

Rakshith S. M.

Student, Department of Computer
Science and Engineering
B.M.S. College of Engineering
Bangalore, India

Tenzin Kunsang

Student, Department of Computer
Science and Engineering
B.M.S. College of Engineering
Bangalore, India

Aashish Badami

Student, Department of Computer
Science and Engineering
B.M.S. College of Engineering
Bangalore, India

Umadevi V.

Associate Professor, Department
of Computer Science and
Engineering
B.M.S. College of Engineering
Bangalore, India

Muzammil Hussain, PhD

Founder,
Tika Data Services Pvt. Ltd.
Bangalore, India

ABSTRACT

This paper proposes a credit system employing face recognition that will enable payment transaction process faster than existing systems. Supermarkets are often slow when processing payments. For example, the customer will have to provide his/her card and then enter the pin number details to the card swiping machine. The work described in this paper aims to simplify the payment processing interface, with minimum user interference, and ensuring a fast payment checkout. Deep learning techniques have been deployed in the proposed architecture, largely composed of convolutional neural networks. All that is required to process a payment transaction is the customer's face image, and nothing else. The customer's face, once recognized, will enable the system to fetch his bank account details and allow the transaction to proceed by debiting the required amount from his/her bank account linked to the payment system. The face recognition model described in this paper has a 100% accuracy in predicting the correct output. The focus of this work is to recognize customer's face for online payment.

General Terms

Face Recognition, Deep learning, Model, Dataset, Online Payment System, Cost Function, Neural Networks

Keywords

Anchor image, Convolutional neural networks, Face recognition, Facenet model, Inception networks, MTCNN model, Triplet loss function

1. INTRODUCTION

Computer vision technologies have taken the world by storm. They have been versatile and have been used in areas like-medical diagnosis, satellite imaging, face recognition tasks, etc. There have been regular advancements in this -field and the better results have led to their applications across a diverse set of fields. The proposed online credit system described in this paper can be used by market vendors, as a payment checkout interface. This system needs minimum customer involvement and just requires the face image of the customer. After the customer's face is recognized, the bill amount gets automatically deducted from the customer's bank account that is linked to the payment system. The system's main component is face recognition. In this paper, details as to how the task of implementing the online credit system using face

recognition is achieved have been described. An image is first passed to our face recognition model. The model outputs a 128 length vector encoding. This encoding is then compared to the embedding of each person's image stored in the database.

An overview of the rest of the paper is as follows: Section 2 – Review of the literature relating to this work, Section 3 - Describes the proposed architecture design of the system, Section 4 – Discusses on different methods that were used to train our model, Section 5 - Deals with the dataset that was used to train our model, and finally, Section 6 - Results of the proposed our model will be presented..

2. RELATED WORKS

There has been a lot of work on face recognition related tasks. Some of the most popular works are-

DeepFace model[1] by Facebook, Facenet model [2] by Google, MTCNN model (Multi-task Cascaded Convolutional Network) [3], etc. These models have high accuracy in recognizing face image, the DeepFace model [1] has 97.35% on the LFW dataset and Facenet model [2] has up to 95.12% on the YouTube Faces Database. These models have many common components in their architecture. These models are largely composed of deep Convolutional Neural Networks with varying sizes of filters. A repetitive architecture is observed in these models[2], there have been variations with the inclusion of Inception Networks [2]. The MTCNN model[3] focuses on extracting the facial region in an image. A multi-task cascaded CNN framework is employed for joint face detection and alignment. Training the model is based on 3 stages- face classification, bounding box regression, and facial landmark localization. Learning was based on back-propagation. The Facenet model [2] consists of deep convolutional networks with inception networks, which help in cases of different feature sizes. The learning phase is based on a triplet loss function, which involves the anchor, positive and negative images. The model tries to maximize the difference between the anchor image and the negative image, while reducing the difference between the anchor image and the positive image. It achieves a classification accuracy of 98.87% on the LFW dataset. DeepFace model [1] highly stresses on face alignment tasks, based on a Siamese Network model. A multi-class network is trained to perform face recognition task on over 4000 identities. Weights are learned

based on the standard back propagation technique. This model is efficient in terms of computational complexity- it runs at 0.33 seconds per image, which involves image decoding, face detection and alignment, feed forward network and the final classification output.

The FaceNet model is shown to outperform several other state of the art models. The results have been clearly portrayed by the deformable FaceNet model [4], which gives a higher accuracy rate compared to the CosFace model and the ResNet 152 model. Face Recognition based Surveillance System Using FaceNet and MTCNN on Jetson TX2 [5] was successfully implemented with multiple cameras. This standalone system detects the person using their face image and an embedding being created was successfully detected with an accuracy of 97%.

3. PROPOSED DESIGN OF THE SYSTEM

The problem statement can be well summarized by the diagram shown in Figure 1. In a broader sense, the high-level design of the entire work has three main modules:

1. MTCNN model
2. Face Recognition model (FaceNet)
3. Comparison of image embedding

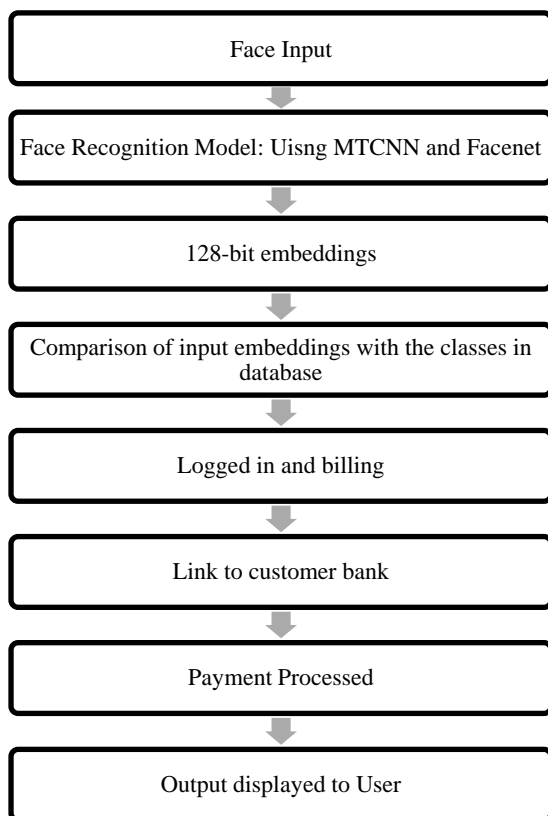


Fig 1: Face Recognition using FaceNet Model and MTCNN model

The MTCNN model is used to provide the facial region of any given input image. The output is then passed on to the face recognition model. A third party source code implementation [6] of the FaceNet model [2] has been used. The FaceNet model outputs a 128 length vector embedding. The final task involves the comparison of this embedding to the embedding of images of different people stored in the database. The

proposed credit system employs a face recognition model based on the FaceNet model [2] by Google. The model was trained on the CASIA WebFace dataset [7]. The model's architecture is based on convolutional neural networks, with inception networks also being a part of it in some layers. The extensive application of the inception modules [8] is a key component in the model. Inception modules make use of filters with multiple sizes that operate on same level. This is in contrast to plain convolutional networks, where all filters operating on any level have the same dimensions. The advantage of having filters of different sizes operate on the same level is that the model is better at identifying salient features in the image, which may have large variations in size. Inception networks have been the go to solution in instances where it has been hard to choose the right filter size.

The detailed architecture for the MTCNN model and FaceNet model are shown in Table I and Table II respectively. The model takes input as an image and the output generated by the model is a 128 dimensional vector. Euclidean distance is used to check for similarity among the vectors. The database has embedding of face image of different people. When a new embedding is received, we classify the embedding as belonging to the person with whom the Euclidean distance is the least, and is below a certain threshold. If there is no person for whom the Euclidean distance is below the threshold, the person's image will not be validated.

Table 1. Architecture of Mtcnn Model

layer	kernel	filters	input	output	Parameters
Conv1	3x3	24	32x32x3	30x30x24	648
Prelu1			30x30x24	30x30x24	24
Conv2	4x4	24	30x03x24	14x14x24	9216
Prelu2			14x14x24	14x14x24	24
Conv3	4x4	32	14x14x24	11x11x32	12288
Prelu3			11x11x32	11x11x32	32
Conv4	4x4	48	11x11x32	8x8x48	24576
Prelu4			8x8x48	8x8x48	48
Conv5	4x4	32	8x8x48	5x5x32	24576
Prelu5			5x5x32	5x5x32	32
Conv6	3x3	16	5x5x32	3x3x16	4608
Prelu6			3x3x16	3x3x16	16
Conv7	3x3	2	3x3x16	1x1x2	288
Prelu7					

4. METHODOLOGY

There are several methods that are employed to train the face recognition model. The cost function (error loss function) that is used is the triplet loss function [2]. Based on this loss function, the filter weights in the model are changed. Back-propagation is used for the purpose of changing the weight values. The final step is to classify a given image, to one of the similar class among several classes that are present in the database. The concept of Euclidean distance helps with this task.

Table 2. Architecture of FaceNet model by Google

Number of kernels/layer type	Kernel size
32	3*3
32	3*3
64	3*3
Max-pooling	3*3
86	1*1
192	3*3
256	3*3
Inception-net (depth=5)	1*1(96),3*3(96)
Inception-net	1*1(192),3*3(578)
Inception-net (depth=10)	1*1(256),1*7(128),7*1(128)
Inception-net	1*1(768),3*3(1154)
Inception-net (depth=6)	1*1(384),1*3(192),3*1(192)
Average pooling	3*3
Fully-connected	Output-128*1
Fully-connected	Output-128*1

The following steps have been used:

4.1 Facial Region Extraction

The first step before training is to convert the image to a form that includes only the facial region. Thus, for all images in the data set, there will be a new set of images, which include only the facial region. These new sets of images would be used for training the model.

The process of facial region extraction is done using the MTCNN [3] framework. This model was developed to focus on areas like face classification, bounding box regression, and facial landmark localization. All images, after having been fed through the MTCNN model, are resized to 160*160*3, which will be the input to the face recognition model. The MTCNN framework [3] has outperformed other state of the art model on several challenging benchmarks while keeping real-time performance. Therefore, using it would add robustness to the face recognition model.

4.2 Triplet Loss Function

The learning phase in the face recognition model involves the use of the triplet loss function [2]. This function aims to ensure that an anchor image of a person is closer to the positive images and farther away from negative images.

Thus, if

- α denotes the margin that is enforced between positive and negative pairs,
- x^a denote the anchor image,
- x^p denote the positive image,
- x^n denote the negative image,
- \forall denotes for all.

Then the following equation sums up the requirement:

$$(x^a - x^p)^2 + \alpha < (x^a - x^n)^2 \forall (x^a, x^p, x^n) \quad - (1)$$

When training, the triplets were selected that violate the triplet constraint, resulting in faster learning of the system. This is the main loss function of the face-recognition model. The triplet loss function is summarized by the Figure 2, which shows that the function learns to minimize distance between positive and anchor image, and tries to maximize the distance between the anchor and negative images.

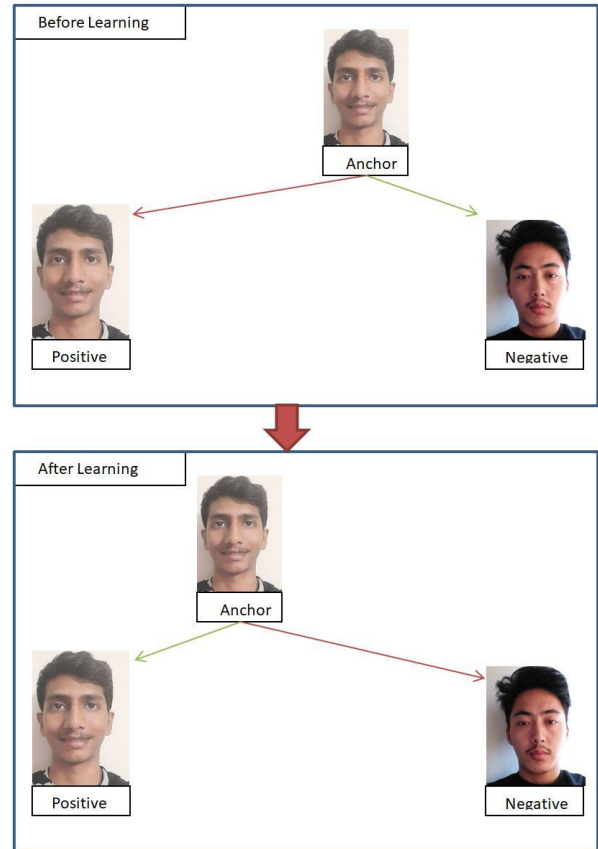


Fig 2: Application of Triplet Loss Function to recognise similar user's face and comparison of results before and after learning both columns

4.3 Back-propagation

The triplet loss function is used as the cost function. The method employed by the model to learn the weights was standard back-propagation, which uses gradient descent. In this case, the filter parameters are randomly initialized at the beginning. They weights are modified, based on the partial derivative of the cost function with respect to the filter parameters. This process is done for all the parameters and several iterations may be required in order to achieve a high degree of performance.

4.4 Comparison of Image Embedding

The final stage in the model is to classify the new test image as belonging to a particular class. Euclidean distance is used in the model to determine which class a given embedding belongs to. The model's database has an embedding stored for each person. When a new embedding is received, it is compared to every other embedding in the database. A certain threshold used in determining the output class, failing which the input image will not be validated.

5. DATASETS

The dataset used to train the model was the CASIA Web Face [7]. This dataset consists of about 500K images spread across 10,575 subjects.



6. RESULT

The model is deployed on the server using Flask framework. The server has provisions for live detection, as well as to add new users and predict the class for a new image. The model's final classification task achieves an accuracy of 100%. We have deployed one-shot learning [5] wherein only one training image per customer is needed to train the model. One-shot learning proves beneficial on a large scale, not putting load on the system and facilitating fast response time as well.

Table 3 shows the recognition of an image with high accuracy, that was already trained, and Figure 3, shows that system cannot recognize an image, which is not trained.

The different models and their results can be compared, as shown in Table IV. The CASIA Webface Dataset was used as the training set for the model, and for testing the performance a different dataset, the Youtube Faces DB[9] was used.

Table 3. High Accuracy is achieved by Successfully detecting the Customer's Identities

Input image	Predicted Output	Actual Output	Accuracy
	Rakshith	Rakshith	100%
	Tenzin	Tenzin	100%

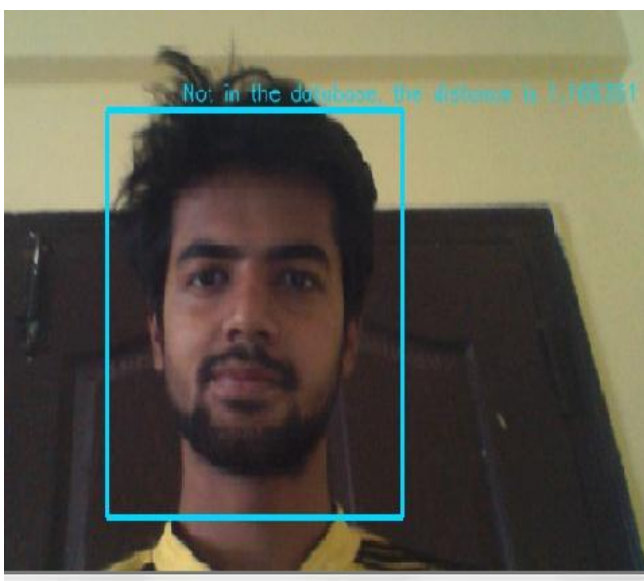


Fig. 3: System cannot recognize an image, which is not trained.

Table 4 Comparison of Facenet with other models

Dataset	Percentage of Accuracy		
	Facenet Model[2]	Deepface Model[1]	deepID2 Model[10]
Youtube FacesDB[9]	95.12%	91.4%	93.2%

7. CONCLUSION AND FUTURE SCOPE

The proposed face recognition model in this work has nearly 100% accuracy. The inclusion of MTCNN framework allowed the model to focus more on the subtle features in people's faces, and inception blocks allowed to capture variance in feature sizes. This could be a great replacement for card payment by customers and vendors, takes less time and also avoids the possibility of forgery. However, the model lacks liveliness detection, and in some cases can be fooled by being shown an image of a person. To counter this in future, the model could be trained so that it is able to detect live frames.

Deploying one-shot learning on a large scale is challenging, as there is only one image per customer. In such a scenario, a Siamese network that is trained on thousands of images would be beneficial. Moreover, tests should be conducted over images captured by different cameras.

8. REFERENCES

- [1] Taigman, Yaniv, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. "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 Location: Columbus, Ohio (USA) Date: June 2014.
- [2] Schroff, Florian, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015. Location: Boston, Massachusetts (USA) Date: June 2015 Version: 3rd version
- [3] Xiang, Jia, and Gengming Zhu. "Joint Face Detection and Facial Expression Recognition with MTCNN." 2017 4th International Conference on Information Science and Control Engineering (ICISCE). IEE 2017. Location: Changsha, China Date: July 2017
- [4] He, Mingjie, Jie Zhang, Shiguang Shan, Meina Kan, and Xilin Chen. "Deformable Face Net: Learning Pose Invariant Feature with Pose Aware Feature Alignment for Face Recognition." In 2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019), pp. 1-8. IEEE, 2019. Location: Lille, France Date: May 2019
- [5] Jose, Edwin, M. Greeshma, Mithun Haridas TP, and M. H. Supriya. "Face recognition based surveillance system using facenet and mtcnn on jetson tx2." In 2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS), pp. 608-613. IEEE,

2019. Location: Coimbatore, India Date: March 2019
- [6] Github, URL: <https://github.com/davidsandberg/facenet> [Last accessed: Apr 2018]
- [7] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," 2014, arXiv:1411.7923. [Online]. Available: <http://arxiv.org/abs/1411.7923> Date: November 2014 Version: 1st Version
- [8] Szegedy, Christian, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions." In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1-9. 2015. Location: Boston, Massachusetts (USA) Date: June 2015
- [9] L. Wolf, T. Hassner, and I. Maoz. Face recognition in unconstrained videos with matched background similarity. In IEEE Conf. on CVPR, 2011. 5 Location: Colorado Spring, Colorado (USA) Date: June 2011
- [10] Sun, Y., Chen, Y., Wang, X., & Tang, X. (2014). Deep learning face representation by joint identification-verification. In Advances in neural information processing systems (pp. 1988-1996). Date: June 2014 Version: 1st