Deep Authentication model using TensorFlow

Manikartheek Kanasani
Department of Computer Science
New York Institute of Technology
Vancouver, Canada
mkanasan@nyit.edu

Hari Priya Aalla
Department of Computer Science
New York Institute of Technology
Vancouver, Canada
haalla@nyit.edu

Dhana Lakshmi Muddana
Department of Computer Science
New York Institute of Technology
Vancouver, Canada
dmuddana@nyit.edu

***Abstract*—Mobile phones have become an integral part of life. The number of mobile users is expected to hit 5 billion by 2020 [1]. The use of various mobile applications is increasing exponentially. Most biometric authentication models in mobile computing are based on classical models. Deep Convolutional Neural Networks (DeepCNN) outperforms these models in most of the cases. Deep Learning is a Machine Learning technique that learns features directly from training data. It has gotten so much hype mostly due to high computation power offered by GPUs and huge amounts of labeled data accessible in past few years. In this paper, the DeepCNN model [2] is implemented using TensorFlow framework and Keras API. Model uses DeepCNN for feature extraction and achieve high accuracy. This paper aims to build upon previous work by using additional datasets, different input sizes and analyze the results.**

Keywords—Mobile Computing, Biometric Authentication, Deep Learning, Convolutional Neural Networks, TensorFlow.

I. Introduction

Authentication is one of the most important challenges in Mobile Computing Security. There are three factors that can provide a form of authentication [3]. (i) What you know – password, PIN or codes; (ii) What you have – Authentication token, access cards; (iii) What you are (also called Biometric Authentication) – uses fingerprints, retina scans or voice. Biometric Authentication gives access only after verifying the unique biological features of individual. Since the passwords can be cracked and cards can be stolen, authenticating a person using unique features is better than the previous cases. With advances in Deep Learning, Biometric Authentication models can be trained faster and achieve results with high percentage of accuracy.

An Artificial Neural Network (ANN) is inspired by neurons in the brain. They have numerous applications like classification, prediction. Usually an ANN consists anywhere between hundreds and millions of neurons arranged in a series of layers. The layers receive input data (which the network wants to process and learn) and transforms it someway and sends to next layer. The layers after input layer are usually hidden except the final output layer. The output layer displays the results (for classifications and predictions. Google uses a 30-layered neural network to power Google Photos and Facebook uses artificial neural networks for its DeepFace algorithm, which can recognize specific faces with 97% accuracy [4]. TensorFlow is an opensource software library developed by Google. Main application of TensorFlow are Deep Convolutional Neural Networks. We are implementing a Biometric Authentication model using Deep Convolutional Neural Networks (DeepCNN). Deep usually refers to multiple hidden layers. DeepCNN provides automatic feature extraction and outperform classical models like Local Binary Pattern (LBP), Support Vector Machine (SVM) and Histogram of oriented gradient (HOG). This paper uses the model 'Deep Authentication in Mobile Cloud Computing' proposed in [2], to build a DeepCNN with different properties (e.g. number of feature maps, window size, number of neurons and so on), to analyse results on how accuracy varies with different input sizes other than 28x28 (Example: 14x14, 56x56). The dataset used in [2] is FEI Face Dataset [5] with 2800 images. We are implementing this model with three more datasets: Georgia Tech Face Dataset [6] with 750 images and Face Recognition Data [9] by University of Essex with 7900 images and our own custom dataset based on ethnicity (with Asian, African American and Indian).

The remaining sections are structured as follows: providing some related works of biometric authentication in section 2, in section 3 and 4 we present our Deep authentication model and methodology. The fifth section discusses experimentation and results; sixth section concludes this paper.

II. Related Works

As discussed earlier, biometric authentication is more secure than the other two. We can use facial features, fingerprints, iris, retina or anything unique to a person. Face recognition is one of the most prominent techniques of biometric authentication [8]. In Face recognition, the person is authenticated using his face. By using the front camera of the device as a source in [9]. The authors proposed real-time detection of users cropped faces for continues authentication.

D Crous et al., in [10] proposed and implemented face based continuous authentication system. They fused mobile device face capture with other sensors like gyroscope, accelerometer and magnetometer data to correct the camera orientation.

In [11] Lehad AL Rassan et al., proposed a model to prevent unauthorized access to the mobile cloud using fingerprint recognition. They used a fingertip image through a mobile phone camera to authenticate the user. Feature Extraction and image processing are done by Converting RGB to Grey-scale image, Normalization, Reduce the blur effect, Segmentation, Orientation Estimation, and Ridge Enhancement and then calculate a similarity score based on which the user is either given access or denied.

Abdelhalim Zeroval et al., in [2] implemented biometric authentication using Deep Convolutional Neural Networks for better feature extraction and Softmax classifier for classification. They outperform classical models in almost every case. The authors used a small data set of 2800 images and fixed input (28x28) for training and testing DeepCNN. The next section presents the model we are implementing. We are building upon their [2] work by implementing model on three different datasets and two different input sizes.

III. Proposed Model

The proposed authentication model in [2], uses DeepCNN for feature extraction and Softmax as classifier. The reason for choosing softmax classier is due to its ability to give probabilities of every available class. Primarily DeepCNN provides high accuracy but it also has other benefits as cited in [12]. They are:

1. Feature extraction is automatically obtained from training data.
2. The variety of weights to analyse is lots smaller than the quantity of weights for the usual neural network.
3. Every layer learns a greater specialized characteristic set on top of the preceding layer features.

The actual authentication process is depicted in Figure 1. It has three steps:

- The mobile user takes a picture of himself; resizes the image and sends it to the system for authentication.
- We implement Deep Face Recognition system based on DeepCNN for Authentication.
- Depends on the decision received from the system user is either given access or denied to the application.

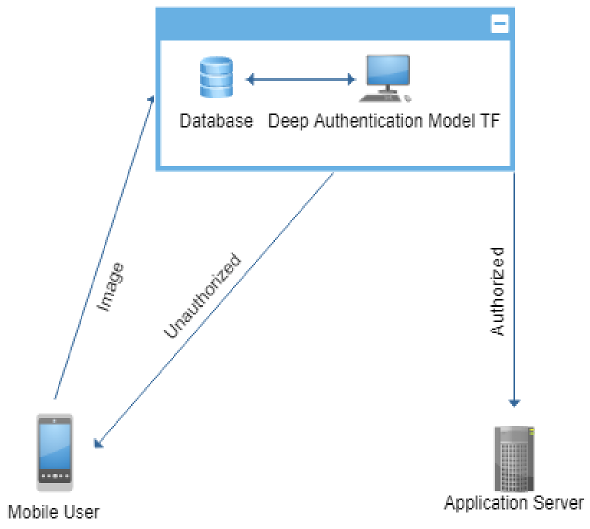


Fig. 1. Architecture for a deep authentication model

DeepCNN is used for training and testing images. The architecture of proposed Deep Convolutional Neural Network in [2] is: INPUT–(CONV/RELU)-POOL-(CONV/RELU)-POOL-FC. The architecture of DeepCNN is shown in Figure 2. The structure (properties) will be different for each dataset since output layer has neurons equal to number of classes. The fully connected hidden layer before final output layer has different number of neurons as well. The layers of DeepCNN will be presented in methodology section.

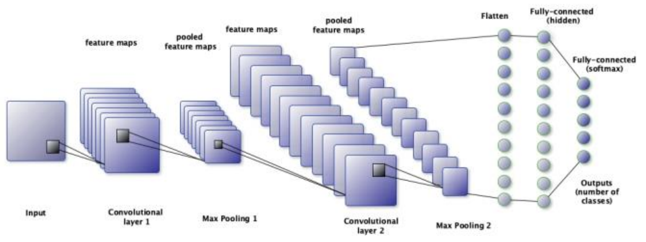


Fig. 2. Diagram of proposed DeepCNN (without properties)

The next section discusses the methodology, architecture of DeepCNN and how it is going to be different from previous work.

IV. Methodology

Deep Convolutional Neural Network Architrcture:

As shown in Figure 3, DeepCNN takes raw pixel data as input, and the input is sent to the convolutional layer.

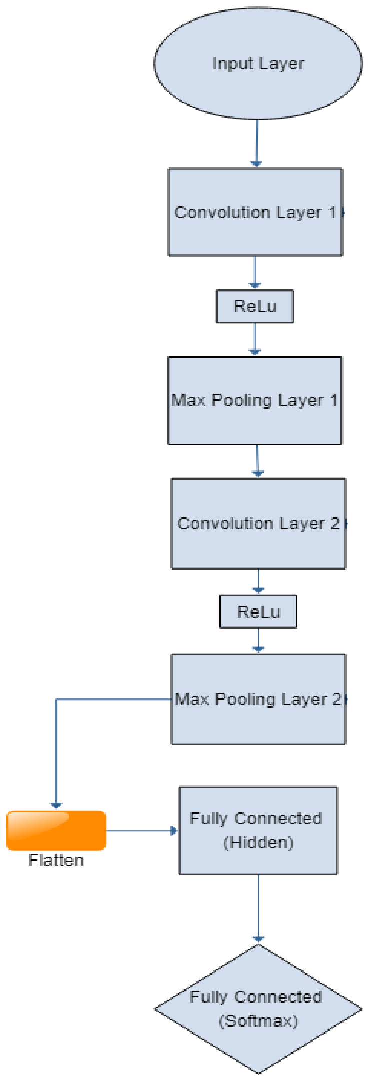


Fig. 3. Sequential model of DeepCNN

Figure 3 depicts the sequential model of DeepCNN, meaning the data flows sequentially from one layer to other. The data from input layer goes to convolutional layer (hidden layer).

A. *Convolutional Layer*

Idea of convolution is to find the useful features. It is a hidden layer which contains filters that detects patterns and images, the proposed model used 32 neurons and we used the same for our datasets. Then we have a convolutional window, which is going to attempt to simplify its findings to a value. It shifts over when it is done; and process is repeated till entire pixel data is covered. In case of window slide over the edge we use padding. In [2] researchers used 5x5 window and 3x3 window for first and second convolutional layers respectively. We used different parameters for different datasets.

B. *ReLU Activation Function*

After each convolutional operation, we use ReLU as an activation function. This activation function helps in mapping particular set of output to a particular set of inputs. Think of activation function as a graph. By using the summation of inputs multiplied by their respective weights the activation function determines whether neurons fire or not.

C. *Max Pooling Layer*

After Convolutional Layer it passes through Max Pooling layer which reduces the size of the input. When a (2,2) max pooling is applied largest values in 2x2 window are picked; thereby reducing the size of input.

Another round of Convolution Layer, ReLU activation function and Max Pooling layer is applied for efficiency and precision.

D. *Flatten*

The resultant matrix is full flattened and forwarded to the final layer. The data so far is two dimensional during convolutional and pooling operation; since the final Fully Connected Layer is one dimensional we flatten the matrix.

E. *Fully-Connected Layer*

The fully connected layer classifies data into various classes. There are two of these layers one hidden and other one is output layer (number of neurons equal to number of classes)

F. *Softmax Classifier*

Softmax Classifier is a type of activation function, used to impart probabilities. It gives the classification where the probability of a class is maximum. We use this to compute the class scores of each person. Softmax computes scores are classifies if the person is already in the database and the system grants or denies the access accordingly.

As discussed earlier, the few parameters in architecture of DeepCNN will be different for each dataset. The layers of DeepCNN is shown in Table 1. We added a garbage class to each dataset to classify ‘no_match’.

	FEI Face Dataset	Georgia Tech	Face Rec Data	Ethnicity
Convultion1	32 neurons 5x5 window	32 neurons 5x5 window	32 neurons 5x5 window	32 neurons 5x5 window
MaxPooling1	2,2	2,2	2,2	2,2
Convultion2	64 neurons 3x3 window	64 neurons 3x3 window	64 neurons 3x3 window	64 neurons 3x3 window
MaxPooling2	2,2	2,2	2,2	2,2
Fully Connected (Hidden)	512 neurons	2000 neurons	6000 neurons	1000 neurons
Output Layer	201 neurons	51 neurons	396 neurons	3 neurons

Tab. 1. Layers of DeepCNN

We used Python as our programming language and using the Keras [13] library for developing our proposed model of DeepCNN. Keras is a high-level API that sits on top of TensorFlow. We also installed TensorFlow-GPU [14] and its prerequisites CUDA9, CuDNN, Microsoft Visual studio. We also installed Jupyter notebook, and we added OpenCV, Sklearn, Matplotlib, Scipy, Pillow, Requests, h5py, plutil libraries for resizing images, splitting the dataset, plotting the results, etc.

Our machine is characterized by 8GB RAM, Intel Core i7-4710HQ CPU 2.50GHz, 1 TB of Hard Drive and 4GB NVIDIA GeForce GTX 860M Graphic card.

V. EXPERIMENTATION AND RESULTS

A. *Dataset*

In [2], authors used FEI face dataset [5] that contains 2800 images of 200 persons (100 male and 100 female). The size of each image is 640x480 pixels.

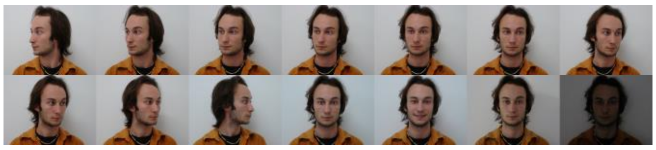


Fig. 4. Examples of images of FEI Face Dataset [5].

We’re using two more datasets:

- Face recognition data [5] in Figure 4 by the University of Essex that contains 7900 images of 395 persons (different genders and races). The size of each image is 180x200 pixels.
- Georgia Tech face database [6] in Figure 5 is another dataset with 800 images of 50 persons (different genders and races). The size of each image is 150x150 pixels.

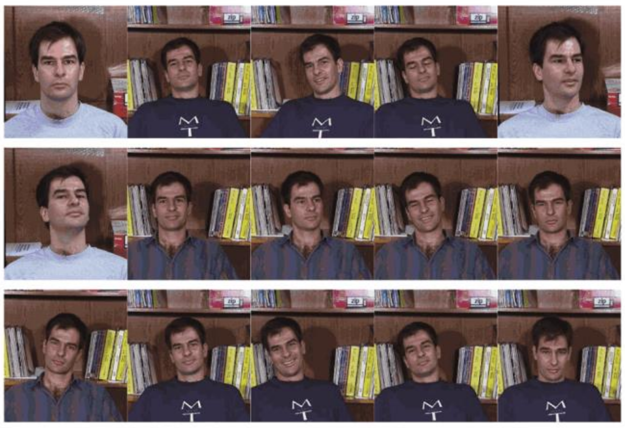


Fig. 5. Images of Georgia Tech face database [6].

The reason for selecting these two datasets is that, Georgia Tech Dataset has fewer number of people and Face Recognition Data has higher number of people compared to FEI. We want to see how model adapts with increasing number of users. Since we are using different datasets, different backgrounds of images might become an issue during the training process

The fourth dataset we implemented out model on, is custom made. We manually picked images based on ethnicity added few more from Google Images [12]. It had 1540 images of people belonging to African American, Asian and Indian ethnicity. We used 80% of the dataset to train the model and test with 20% of the dataset, for all four datasets.

B. Results

The results for each dataset are discussed in subsections below.

1) FEI Face Database:

With the parameters mentioned in Table 1, We trained the model with 80% of dataset (2380 images) and tested on 20% of the dataset (600 images) for three input sizes 14x14, 28x28, 56x56. We used 128 batch size and 50 epochs. Batch size is the number of samples that will be passed through the network at one time, an epoch is one single pass of all the data through the network. The results are shown in Table 2.

	14x14	28x28	56x56
Loss	0.0370	0.0265	0.0437
Accuracy	0.9955	0.9955	0.9901
Validation Loss	2.8109	1.3610	1.2536
Validation Accuracy	0.6007	0.8003	0.8284

Tab. 2. Results of FEI Face Dataset

The loss is rate of error the model is making when classifying the images. Accuracy is calculated based on rate of the model. Loss and Accuracy refers to images that model already seen during previous epoch; whereas the validation loss and validation accuracy refers to images the model did not see during training.

As we can see from Table 2, validation loss is gradually decreasing, and validation accuracy is improving when we increase the input size. The main observations are:

- 14x14 is an unreliable input size, since we are only getting 60% accuracy.
- Images were taken in an upright frontal position with profile rotation of up to about 180 degrees (Figure 4 depicts the images of this dataset), This might have affected the validation accuracy since the model cannot see full facial features in some images.
- Since the picture of the user can be taken in any angle, these are the results we can expect in real life scenario.

2) Georgia Tech Face Database:

This is the smallest dataset of four. With the parameters mentioned in Table 1, We trained the model with 80% of dataset (740 images) and tested on 20% of the dataset (180 images) for three input sizes 14x14, 28x28, 56x56. We used 32 batch size and 50 epochs. The results are shown in Table 3.

	14x14	28x28	56x56
Loss	0.1513	0.0030	0.0085
Accuracy	0.9948	1.000	0.9959
Validation Loss	0.8388	0.8160	0.7202
Validation Accuracy	0.7923	0.9016	0.9116

Tab. 3. Results of Georgia Tech Face Database

As expected, validation accuracy is improving with the input size. The main observations are:

- This dataset got best validation loss and accuracy, compared to other datasets. As shown in Figure 5, The pictures show frontal and tilted faces with different facial expressions.

3) Face Recognition Data:

This is the largest dataset that we used, with over 8000 images of 395 persons. We did the same 80% and 20% split for training and testing the model for input sizes 14x14, 28x28, 56x56. We used 256 batch size and 50 epochs.

	14x14	28x28	56x56
Loss	0.1897	0.0025	0.0021
Accuracy	0.9438	1.9987	1.0000
Validation Loss	0.9066	1.4241	0.9264
Validation Accuracy	0.8306	0.8534	0.8716

Tab. 3. Results of Face Recognition Data

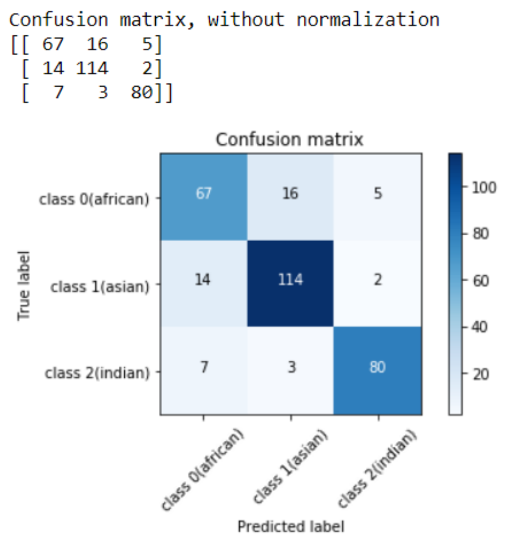
As shown in Table 3, the validation accuracy went up with increasing input size. The main observations are:

- This dataset had the best validation accuracy for 14x14 input size, compared to the other two datasets.

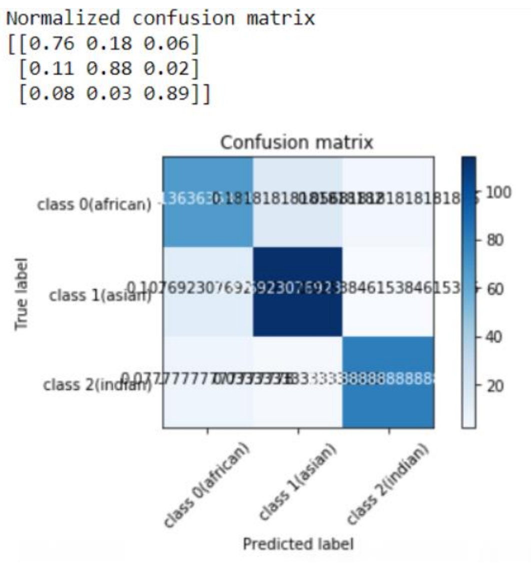
- Even though the images had random backgrounds, different scales and lighting conditions with high number of users: the model gave good validation accuracy.

4) Ethnicity:

As mentioned earlier this is a custom dataset with three groups: African American, Asian, Indian. We trained the model with input size of 56x56, and we got validation loss of 1.3009 and validation accuracy of 0.8312. The results are presented in the form of confusion matrix as shown in Figure 6. A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known [15].



```
Confusion matrix, without normalization
[[ 67  16   5]
 [ 14 114   2]
 [  7   3  80]]
```



```
Normalized confusion matrix
[[0.76 0.18 0.06]
 [0.11 0.88 0.02]
 [0.08 0.03 0.89]]
```

Fig. 6. Confusion matrix with and without normalization

The main observations are:

- Model classified Indian (89%) more accurately than other two classes, followed by Asian (88%).
- Most inaccurate classifications between Asian and African (11% & 18%), which is significantly high.
- Fewer inaccurate classifications between Asian and Indian (2% and 3%), comparatively low.
- The reason for the misclassifications between Asian and African might be the model has not taken all facial features (example skin tone) into account. There is no way of knowing in our model, since DeepCNN provides automatic feature selection.

VI. Conclusion

Our Deep authentication model got accuracy ranging from 79% to 91%, except in one case. The training time and validation accuracy is increasing with input size in all three datasets. Input size 14x14 is not reliable since the accuracy for one dataset is very low (60%). For this model, we think input size 28x28 is better than other two input sizes, keeping training time and accuracy in mind. Our model classified a person based on ethnicity with 83% accuracy. For future work, we suggest an authentication model, where the user is classified based on gender, age, ethnicity and face (with different scores for each section, face being highest); And given access only if the person meets a certain score.

References

[1] IHS Markit white paper. Retrieved from: https://technology.ihs.com/589158/artificial-intelligence-next-key-growth-area-for-smartphones-as-numbers-top-six-billion-by-2020-ihs-markit-says

[2] A. Zeroual, M. Amroune, M. Derdour, A. Meraoumia and A. Bentahar, "*Deep authenication model in Mobile Cloud Computing*" in International Conference on Pattern Analysis and Intelligent Systems (PAIS) on, 2018.

[3] Melissa Hedge, The 3 Forms of Authentication & How We Use Them. Retrieved from: https://www.contegix.com/the-3-forms-of-authentication-how-we-use-them/

[4] Bernard Marr, What Are Artificial Neural Networks - A Simple Explanation For Absolutely Anyone. Retrieved from: https://www.forbes.com/sites/bernardmarr/2018/09/24/what-are-artificial-neural-networks-a-simple-explanation-for-absolutely-anyone/#4f595ea41245

[5] C. E. Thomaz. (2005). FEI Face Database. Available: http://fei.edu.br/~cet/facedatabase.html

[6] Georgia Tech Face Database. Available: http://www.anefian.com/research/face_reco.htm

[7] Face Recognition Data. Available: https://cswww.essex.ac.uk/mv/allfaces/

[8] S. Alotaibi, S. Furnell, and N. Clarke, "*Transparent authentication systems for mobile device security: A review,*" in Internet Technology and Secured Transactions (ICITST), 2015 10th International Conference for, 2015, pp. 406-413.

[9] U. Mahbub, V. M. Patel, D. Chandra, B. Barbello, and R. Chellappa, "*Partial face detection for continuous authentication,*" in Image Processing (ICIP), 2016 IEEE International Conference on, 2016, pp. 2991-2995.

[10] I. Al Rassan and H. AlShaher, "*Securing mobile cloud computing using biometric authentication (SMCBA),*" in Computational Science and Computational Intelligence (CSCI), 2014 International Conference on, 2014, pp. 157-161.

[11] D. Crouse, H. Han, D. Chandra, B. Barbello, and A. K. Jain, "*Continuous authentication of mobile user: Fusion of face image and inertial measurement unit data,*" in Biometrics (ICB), 2015 International Conference on, 2015, pp. 135-142.

[12] Google Images. https://www.google.com/imghp?hl=en

[13] Keras. https://keras.io/

[14] TensorFlow-GPU. https://www.tensorflow.org/install/gpu

[15] Data School, Simple guide to confusion matrix terminology. Retrieved from: https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/

VII. Appendix

A. *Advisory comments*

Based on advisor's suggestions on our mid-term report, we made the following corrections:

- We corrected simple errors in text, mobile applications instead of applications, authentication instead of privacy, providing examples for classical machine learning models.
- Defined the following terms: Face Recognition, Biometric Authentication.
- We referenced and discussed all figures and tables in the text.
- We provided references and reasons for selecting our datasets.
- Suggested future work.
- Referenced all tools and what functions they provide.
- In methodology, we described the experiment process. Mentioned the parameters and range of values we used. We compared results based on loss and accuracy.
- We expanded Introduction and Methodology section
- In Introduction section, we added a paragraph that summarizes what each remaining section of the paper is about.

B. *Implementation*

We explained the Deep Authentication model and implemented that in our machine. In this section we show how that model comes in an actual authentication process in Figure 7. The user clicks a photo using his front camera and sends it to Deep Authentication model and based on the classification the user is either granted access or denied.

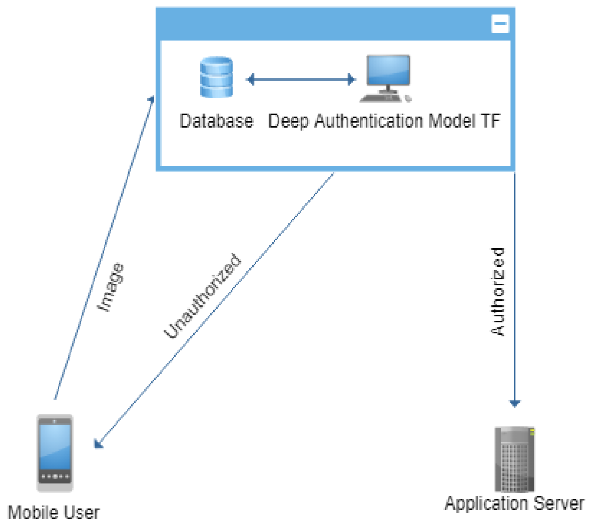


Fig. 7. Authentication Process