International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 05 | May 2022 www.irjet.net p-ISSN: 2395-0072

Nivritti : web-based Real time face recognition

Sujay Ingle[1], Tejas Khangal[2], Yash Bhuse[3], Nitish Das[4]

[1,2,3]Student, Dept. of CSE, MIT SOE, Pune, Maharashtra
[4]Professor, Dept. of CSE, MIT SOE, Pune, Maharashtra

Abstract - *Nivritti is a system capable of real-time face recognition. To collect pension, pension-seeker must get their jeevan-praman verified i.e giving a proof of being alive and proof of been the same person to whom pension is intended. This process of collecting pension from government office is a tedious one and also physical exhausting and unsafe for elderly people. This entire process can be easily carried on by a software using face recognition .As long as a software application verifies pension-seeker's identity from they face each time when they request for they pension, it can serve as they jeevan-praman and pension can be directly transferred to they account ,thus avoiding all the above mentioned hassle .Hence , we developed a system to ease out this process. The system should be such that it must be easily accessible by elderly , compatible with most of the local hardware available to the public and capable of performing face recognition with minimum time and memory requirements. Hence this entire system was developed as a website in Django, making it accessible from any web-browser with ease.I n our proposed work, various face-recognition algorithms and their implementations were compared. Along with this, Dlib was used to improve overall performance of models for face and landmark detection. After analyzing and testing the model, face-recognition + dlib was found to be the best option , given hardware specs and memory consumption. It gave an accuracy of 99.38% ,that too in real-time , which is crucial for our system.*

Key Words: Django, Sqlite, DbBrowser, Python, HTML, CSS, HOG, CNN, Dlib, Face-recognition.

1.INTRODUCTION

Pension-seekers usually include elderly people that once worked in government jobs whether it was an office job or defense one. As a reward for their lifetime service to the nation, the government awards them with a monthly pension after their retirement. To avail for this , elderly have to first visit government office to get the documents verified, Then every month in order to collect they pension they have to visit the government office to give they jeevan-praman i.e just a term to verify if the pension-seeker is alive or not and whether he/she is the person who was supposed to receive the pension. Our aim was to simply, digitize this process.

Jeeven-praman basically constitutes two things , identity and proof of being alive. Now as long as user uploads a new photo of themselves every month, the photo of them verifies the identity and a different photo captured by our system every month acts as a proof of them been alive. So entire process boils down to the fact that can are system recognize the user correctly in real-time with maximum accuracy possible.

Hence, we carried out a survey of all the major face-recognition algorithms that are available to compare out the best one for our system such that it's able to give out prediction in real time, compatible with almost all the local hardware devices available in market and requires least possible memory along with been easy to use. All the technologies that were used were chosen in the view of being most suitable ones for real-time. Also modifications were carried out on the face-recognition algorithms to improve their accuracy and make them more compatible with real time.

1.1 Need for Nivritti

Nivritti is an initiative to ease out struggle of many elderly. Hence the reason that lead to this system also revolves around them:

Elderly , have to visit government offices every month to collect their pension. For someone at that age any task that requires physical effort is a challenge , and this continues to become more and more challenging as they grow old.

After reaching the allocated government center, the task doesn't seem to get easier as they have to stand in queues for hours to get their pension. This situation is even worse for elderly who have difficulty in walking or even in simply standing up.

Government center , from where pension is collected is well known to the common public, hence there is always a threat of theft . There have been incidents in the past where elderly pension-seekers lost their pension to thieves on the way back home.

Sometimes due to lack of coordination , from the officials responsible for carrying out this process there are delays in receiving the pensions causing pension-seekers to visit

government centers multiple times a month and going through all this hassle all over again.

All of this can be avoided by simply digitizing this process and making sure the face-recognition system is sufficiently accurate and capable to work real-time.

1.2 Existing Systems

Telangana government is the only one who has taken an initiative in this area. They developed a similar system for jeevan-praman verification called "T-App". This was also an online face-recognition platform. However , it was later shut down due to its lack of accuracy and capability to provide predictions real-time.

Rest of the system basically includes algorithms(mentioned in section of existing face-recognition techniques)that are used for face-recognition in general and not specifically for the purpose of jeevan-praman verification.

2. Working of System

For using the system users need to get them registered at one of the government facilities. This is where all these documents will be verified and their images for face recognition will be captured. This visit is crucial because for security reasons it's necessary to verify the seekers identity and also since our system requires images in a proper format of face orientation ,is would be better if a trained person at government center captures then for the user .Then each user will be allocated a Pension ID. During this phase , bank details of the users will be linked with their login ID. Entire process of document verification only needs to be done once. Later on only once a year on a given date users need to visit government offices to update their existing image dataset. This increases the accuracy of the system as the face of elderly people changes a lot as they age. Hence to maintain accuracy and security it's better to update the entire image dataset once a year.

Once a profile is created the user needs to move login from their local device. Our website will have a link to a short youtube video explaining how to use our system. After logging in, users need to visit service webpages and click on jeevan praman link. This guide the user to a page where they need to enter their pension id.

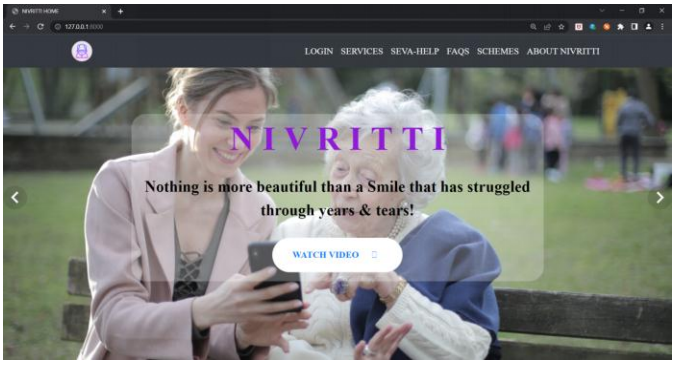

Fig-1: First page of Nivritti

Once id is entered ,our user only need to look at the camera and rest of the procedure will be carried out in background .Our system will first make sure if there is only one person in frame. If it detect multiple faces , it will notify user to isolate themselves while image is been captured. This is done for security purpose. Only one face means more accurate detection. After making sure of single face presence in frame , we move to the next step.

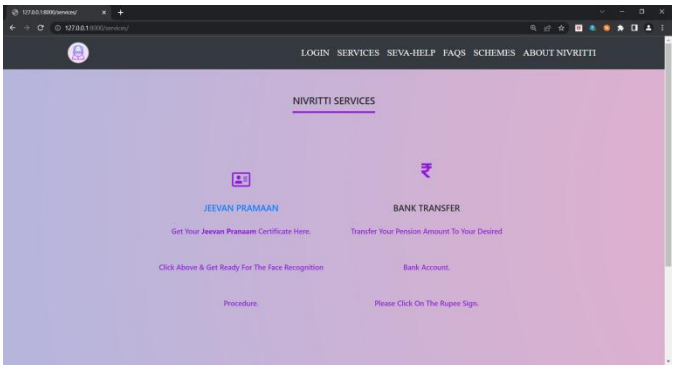

Fig-2: Services provided by the application

Now , our system with the help of a webcam will capture users' images so as to run a face recognition model on it. If for that particular pension id which the user entered before, our system identifies the user based on pre-stored images , the pension of the user will be directly sent to their account without any hassle.

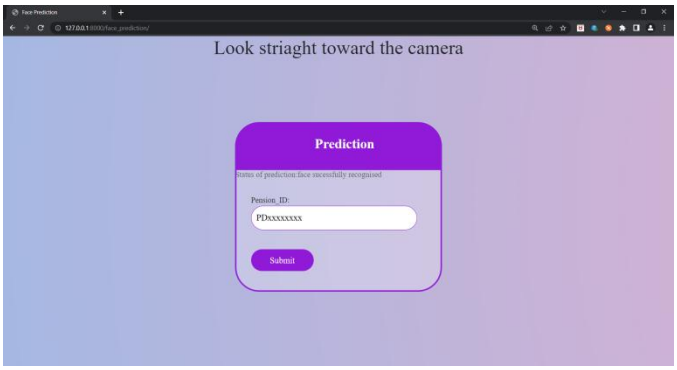

Fig-3: Result display of face recognition algorithm

Apart from our main face recognition system we also have an advertisement page that based on our users pension ,will recommend insurances ,offers on some assets, mediclaims, etc.

If our users are facing any issues with the system , there is also our seva feature that will allow users to send mail to our servers. To make it more convenient for our users ,we will also have a voice based assist ,all our users need to do is speak and their words will be written down into the mail.

3. Flow chart of System

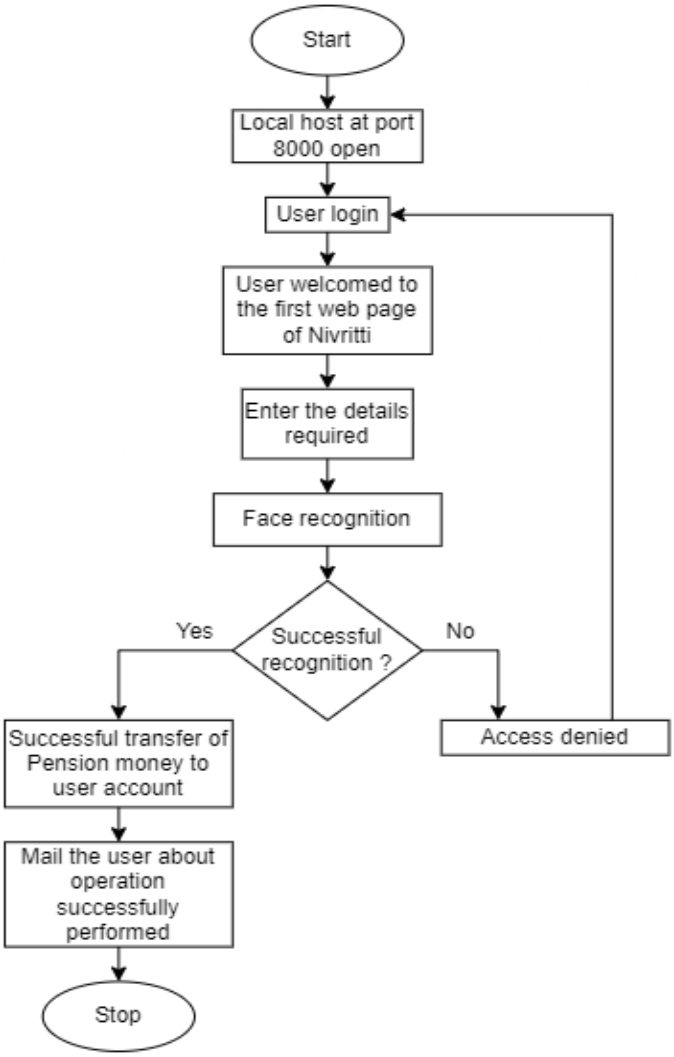

Fig 1: Flow chart of entire system.

Here local host at port 8000 refers to the default port on which Django local host server will be activated.

Entire system is web-based ,hence users will be easily able to login from their local devices.

The following flow chart dives deep into how our face-recognition process will be carried out.

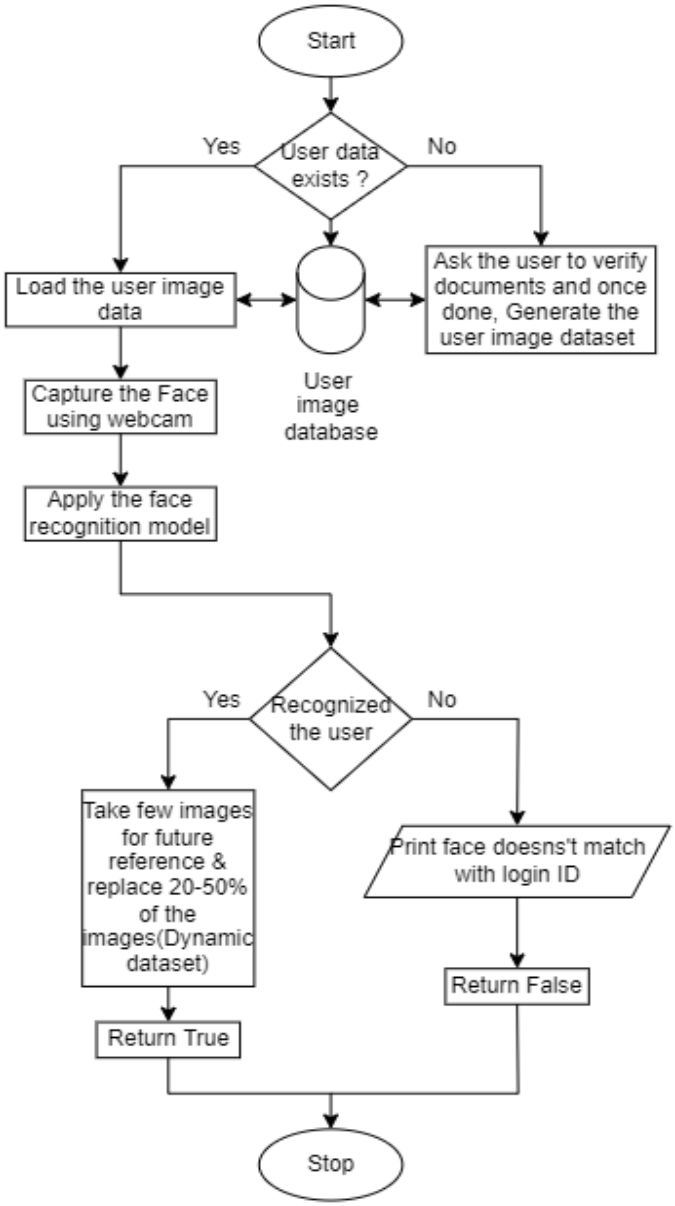

Fig 2: Workflow of backend system.

4. Technologies used

Django:-is a platform that assists the developers for rapid web developments.Along side its predefined methods for implementing many of the feature in a website, what makes it stand out is it capability to display dynamic data in real time onto static pages. Principle behinds it's working is MVT(Model-view-template).Here view refers the part of django where entire logic behind a certain website or backend code is written. It also responsible for rendering our systems html pages along with navigation between them and also data or parameter passing.

Templates refers to the section responsible for the front end. It's where our html pages are stored. This section is responsible for binding our HTML code with CSS and Javascript along with the static images used for presentation purposes.

Last and also another important section is model. Django allows the developers to create database tables in a very easy and reliable manner ,that is by treating tables as classes and its attributes as columns within the table. From table creation to handling error and query requests is done by the models section.

Sqlite:-is the default database system used by django. Being an sql based system it is efficient in handling structured data. It was used to store the profile of our users along with their banking details. It was also used in the login system to verify credentials.

Db Browser:- basically used to view contents of the database tables. Gives a view of how tables actually look with datasets and how values are stored in tuples. This is helpfuls to development purposes to verify query and form propers logic.

Dlib:- Dlib is a modern C++ toolkit containing machine learning algorithms and tools for creating complex software in C++ to solve real world problems.

Face_recognition:- Detect and manipulate faces from the command line in Python or using the world's simplest facial recognition library. Built with dlib's state-of-the-art facial recognition built with deep learning. The model accuracy is 99.38% on the Wild benchmark Labeled Faces.

5. Existing face recognition techniques

Existing implementation of face recognition techniques uses predefined weights that are about half a gigabytes in size, work better over good hardware resources available or relia on huge datasets. But considering the facts, for a real-time execution of the application the face recognition technique must be fast, require small dataset, and is easy to work on any designated hardware.

Table -1: Comparative study of existing models

Various models and there accuracy(The highest accuracy achieved by the implementation)			
Models	**Accuracy**	**Models**	**Accuracy**
ArcFace	98.02	OpenFace	97.3
DeepFace	95.92 ± 0.0029	FaceReco + Dlib	99.38
FAREC	96	SVM with HOG	97.78

After studying existing techniques ,it was concluded that pre-compiled model was the best pick for our system due to following reasons:

The Importance of using the pre-compiled model for easy use was chosen for better performance, easy implementation and updation & less memory as well as hardware requirement.

Better augmentation of image is done with dlib and differentiate easily over contrast changes, brightness, etc.

Face_recognition is better in performance than its peers on any device based on hardware and storage requirements.

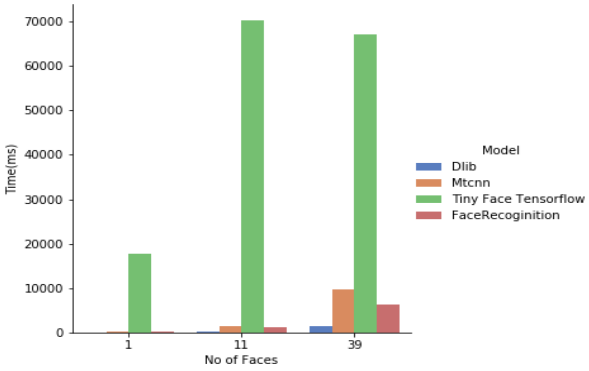

Chart-1: Execution time vs No. of faces to be predicted

Been pre-compiled and written in C++, and uses Dlib for Contour & face detection, the total performance of the model is faster compared to any other implementation given the hardware it's been used on.

Conventional ways of designing the model written in python provide default settings for the implementation of the model, which reduces performance, increases the amount of data required to train the model, and can even deteriorate the hardware over long use.

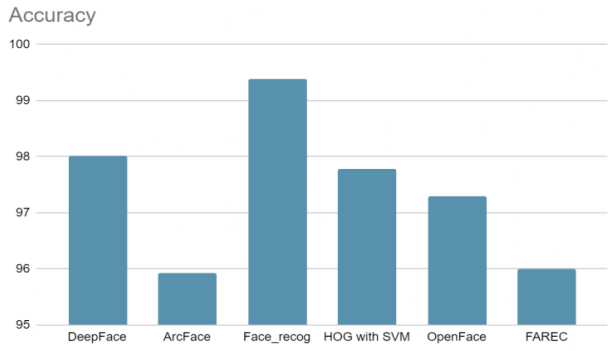

Chart-2: Different face recognition techniques with there accuracy

Various Other models do exist in the category which are more either hardware or data dependent.

Some models promise even better confidence rates than Face_recognition but lack over the hardware that they're been used on.

The basic CNN implementation of models written in C++ is ready to go any time the process of recognition is called, which becomes a bottleneck in others due to on spot compilation, hardware-limitation, interlanguage latency, and complex implementation for feature extraction.

The powerful backed support for a fast memory mapping and performing language reduce the execution time by a lot.

In the dlib face recognition network, the vector output feature is 128-d (i.e., a list of 128 real value numbers) used for face measurement. Network training is done using triplets.

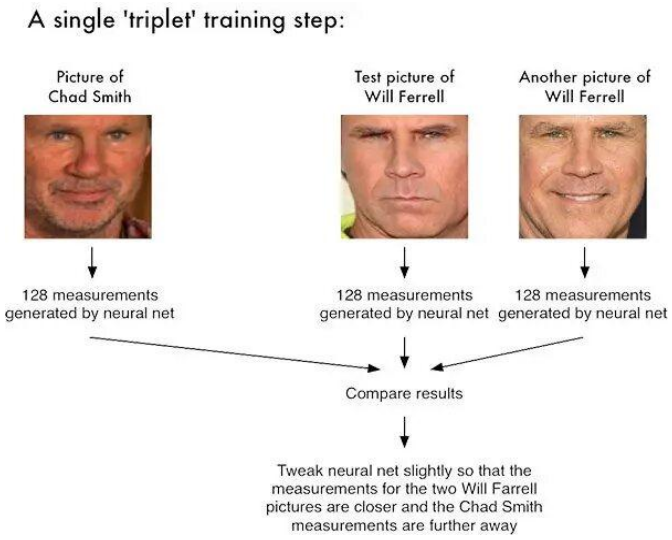

Fig-3: Working of Triplet training

The network itself was trained by Davis King on a ≈3 million photo database. In the Face Extended Face Database (LFW) network compared to other modern methods, it achieves 99.38% accuracy.

6. Modification

Use of Dlib: An excellent choice for face and landmark detection primarily written in C++, gives an upper hand over the performance boost given by the choice of language, and has good and easy to use python bindings.

CNN & HOG: The pre-compiled version is compatible with both HOG(Histogram of Oriented Gradients) and CNN(Convolutional Neural Network), based on the system hardware specification the implementation can change with no further dependencies or data required.

Pre-compiled Models: Rather than using weights that have size about half a gigabyte, the pre-compiled versions are the training model written in C/C++ with more optimization than what conventional python would provide.

Dynamic Dataset: Rather than increasing the data size for the training model, about 20 - 50% of the image would be replaced after successful recognition to retain both the features of the old and new image data since the working doesn't require a large dataset, and would provide excellent performance with no compromise in accuracy.

7. About the dataset

LFW(Labeled Faces in the Wild), dataset was used for comparing accuracies of face-recognition algorithms. It is a dataset of face photographs and usually used to perform face recognition in an unconstrained manner. It consists of 13,233 images of 5,749 different individuals. Format of images is .jpg with dimensions as 250 x 250.

https://www.kaggle.com/datasets/jessicali9530/lfw-dataset

3. CONCLUSIONS

With all the comparison over the ways of implementing a face recognition, the use of face_recognition + dlib is the best and the fastest option to operate the implementation for any device as the model is pre-compiled, less hardware dependent, local dependencies and the memory & data consumption is less compared to other models.

For the implementation of Nivritti, these criteria are very crucial for developing a real time face-recognition model that is reliable and is easy to implement.

REFERENCES

Jiankang Deng, Jia Guo, Niannan Xue, Stefanos Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition", in Cornell University, 2019.

Jiankang Deng, Jia Guo, Niannan Xue and Stefanos Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition", in IEEE, 2019..

Angeliki Fydanaki, Zeno Geradts, "Evaluating OpenFace: an open-source automatic facial comparison algorithm for forensics",FORENSIC SCIENCES RESEARCH, 2018.

Ali Elmahmudi, Hassan Ugail, "Deep face recognition using imperfect facial data", Future Generation Computer Systems, Oct. 2019.

S. Sharma, Karthikeyan Shanmugasundaram, Sathees Kumar Ramasamy, "FAREC — CNN based efficient face recognition technique using Dlib", IEEE, Jan. 2017.

Tadas Baltrusaitis, Peter Robinson and Louis-Philippe Morency, "OpenFace: an open source facial behavior analysis toolkit", in IEEE, 2016.

Ivan William, De Rosal Ignatius Moses Setiadi, Eko Hari Rachmawanto, Heru Agus Santoso and Christy Atika Sari, "Face Recognition using FaceNet (Survey, Performance Test, and Comparison)", in IEEE, 2019.

Q. Cao, L. Shen, W. Xie, O. M. Parkhi and A. Zisserman, "VGGFace2: A dataset for recognising faces across pose and age", International Conference on Automatic Face and Gesture Recognition, 2018.

X. Chen, L. Song and C. Qiu, "Face Recognition by Feature Extraction and Classification", 2018 12th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID), pp. 43-46, 2018.

Tony Gwyn, Kaushik Roy and Mustafa Atay, " Face Recognition Using Popular Deep Net Architectures: A Brief Comparative Study", in MDPI, 2021.

Hongzhi Zhou, Gan Yu, "Research on pedestrian detection technology based on the SVM classifier trained by HOG and LTP features", Future Generation Computer Systems, Dec. 2021.

Vikrant Khadatkar, Shubhangi Bagadkar, Prof. Nutan Dhande, "Face Recognition Based Attendance Management System Using HOG Feature Extraction and SVM Classifier", Ijraset Journal For Research in Applied Science and Engineering Technology, Mar. 2022.

Shashank Reddy Boyapally, "Facial Recognition and Attendance System Using Dlib and Face_Recognition Libraries",SSRN, 2021.

Jeff Forcier, Paul Bissex, Wesley J Chun Python Web Development with Django

William S. Vincent Django for APIs: Build Web APIs with Python and Django

BIOGRAPHIES

	**Sujay Kumar Ingle** , is currently a student at MIT ADT , pursuing B.tech in CSE. He has an interest in machine learning ,deep learning and computer vision.
	**Tejas Khanderao Khangal,** is currently a student at MIT ADT , pusing B.tech in CSE. He has an interest in machine learning and software development.
	**Yash Narendra Bhuse,**is currently a student at MIT ADT , pusing B.tech in CSE. He has interest in data analytics and Business Intelligence.
	**Dr. Nitish Das ,** is assistant professor in department of CSE at MIT ADT. He holds a PHD in electronics and communication engineering.