Human Face Detection and Identification using Deep Metric Learning

Snehal S. Sapkal[1], Alka P. Mengane[2] , Tejashri A. Kalbhor[3], Nitashree S. Ohol[4]

[1]Students of Computer Engineering. PDEA's College of Engineering, Pune, India
[2]Students of Computer Engineering. PDEA's College of Engineering, Pune, India
[3]Students of Computer Engineering. PDEA's College of Engineering, Pune, India
[4]Students of Computer Engineering. PDEA's College of Engineering, Pune, India

Abstract— Human Face Detection and Identification using Deep Metric Learning. In Our proposed project we are using a new technique of Face detection with Human object detection the technique is call as deep metric learning.

Keywords— deep metric, deep learning, machine learning, face detection, recognition, identification, NN, ML

I. INTRODUCTION

In Our proposed project we are using a new technique of Face detection with Human object detection the technique is call as deep metric learning.

We use real-time images or stream videos from CCTV or any other video capturing device or user can put pre recorded or captured video for analysis.

Our system is boosted with widely used classification methods to detect and identify faces even from a blur images or scrappy videos.

II. WIDE DEPENDENCIES WE USED

A. *Open Computer version (OpenCV):*

Efficiently working computer vision repository which is highly efficient and smooths real-time image processing.

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library.

OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

B. *Dlib:*

dlib is a intelligent implementation of the machine learning vectors and deep learning areas to train machine in various scenario like complicated facial recognition.

Artificial Intelligence Behavior Tree Library DiLIB is a behavior tree library with tools to implement it over user interface aimed for C++ programmers.

C. *scikit-learn and scikit-image:*

To create deep learning network to understand inputted data and processes it with compiled version of our code with real time evaluation and high accuracy

Simple and efficient tools for data mining and data analysis, Accessible to everybody, and reusable in various contexts, Built on NumPy, SciPy, and matplotlib, Open source, commercially usable - BSD license

D. *Keras:*

High-level neural networks API. Makes coding, training, and deploying neural networks

Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, Theano, or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible.

E. *Mxnet:*

A scalable deep learning framework. Which is Extremely fast and efficient Capable of scaling across multiple GPUs and multiple machines.

Apache MXNet is an open-source deep learning software framework, used to train, and deploy deep neural networks.

III. OVERVIEW

In Our proposed project we are using a new technique of Face detection with Human object detection the technique is call as deep metric learning.

As we understand in deep learning we typically train a network to:

- Accept a single input image
- And output a classification/label for that image

But deep metric learning is different. We are not trying to output a single label or image or coordinates box of objects in an image, we are directly outputting a real-valued vector.

To perform this type of real time data vector and facial recognition in provided image or video we used some external factors to contribute in our detection. These factors are the external dependencies which we are using to build a complete vector graph and plotting facial vectors in order to valued it with inputted video or image to identify the suspect or our target from source image or video..

A. Motivation

In today's automation era, every information is being processed by the machine with artificial intelligence and used in many sophisticated applications. Even though many agencies have developed the state-of-the-art security systems, recent terrorist attacks exposed serious weaknesses of sophisticated security systems. Hence various agencies are more serious and motivated to improve security data systems based on body or behavioral characteristics, often called biometrics

[1]. Biometric-based technologies include the identification based on physiological characteristics: face, fingerprints, finger geometry, hand geometry, hand veins, palm, iris, retina, ear, voice and behavioral traits such as gait, signature and keystroke dynamics

[2]. Almost all the biometric technologies require some voluntary action by the user, i.e. the user needs to place his hand on a hand-rest for finger printing or hand geometry detection and has to

stand in a fixed position in front of a camera for iris or retina identification. However, face recognition can be done passively without any definite action or participation on the 2 part of the user, since face images can be acquired from a distance by a camera, and hence the face recognition system is more appropriate for security and surveillance purposes. Further, data acquisition in general is fraught with problems for other biometric techniques that rely on hands and fingers. These can be rendered useless if the epidermis tissues are damaged in some way (i.e., bruised or cracked). Expensive equipments are required for the iris and retina identification, and these methods are more sensitive to any body motion. Voice recognition is susceptible to background noises in public places and auditory fluctuations on a phone line or tape recording. Signatures can be modified or forged. However, facial images can be easily obtained with a couple of inexpensive fixed cameras. They cannot be modified or forged, and they are not affected by background sound noise. Face recognition algorithms with appropriate preprocessing of the images may compensate for noise, slight variations in orientation, scale and illumination. Although there are advantages of face recognition over other biometric techniques, existing face recognition technology is not able to satisfy the needs. Several challenges are there in developing face recognition systems, which are:

Illumination Variations: The direction of illumination in the image, greatly affects the face recognition performance. The variations between the same face images due to illumination are always greater than variations in the image due to change in face identity.

Frontal vs. Profile: In a surveillance system, people are not always facing the cameras. The faces are viewed by some angle. The angle with which the photo of the individual was taken with respect to the camera affects the face recognition performance drastically.

Expression Variations: Expression variations in face images affect the performance of face recognition. A smiling face, a laughing face, a crying face, a sad face, a fac with closed eyes, even a small distinction in the facial expression can influence facial recognition system greatly.

Aging: Face images of the same individual of 1 year and 15 years are difficult to recognize since face appearance changes rapidly. Images taken by varying a time from 5 minutes to 5 years change the system accuracy seriously.

Occlusions: It is very difficult to recognize the faces when they are partially occluded. Face images in real-world applications may occlude due to use of things, such as sunglasses, scarf, hands on the face portion, the objects which persons carry, and external sources that occlude the camera view partially.

The ultimate goal of researchers in this area is to develop the sophisticated face recognition to emulate the human vision system. Many researchers have proposed and developed the machine-based face recognition algorithms. Research has been conducted vigorously in the face recognition area for the past four decades, and enormous progress has been made. Still, there is scope for improvement. Encouraging results have been obtained, and current face recognition systems have achieved a specific degree of maturity when operating under various constrained conditions. However, they are far from achieving the ideal of being able to perform adequately in all the various situations that are commonly encountered by applications, utilizing available techniques in practical life.

B. Problem Definition and Objectives

Biometric identification is the technique of automatically verifying or identifying a person by a personal trait or physical characteristic. The term "automatically" means the biometric system must identify or verify a human characteristic or trait promptly with little or no intervention of the user. Biometric technology was developed for its use in high-level security systems and law enforcement markets. The key element of biometric technology is its ability to identify a human being and impose security. The aim of this thesis is to develop an automatic face recognition system, which will improve recognition rates for normal images and for images with illumination and expression variations.

In Our proposed project we are using a new technique of Face detection with Human object detection the technique is call as deep metric learning.

As we understand in deep learning we typically train a network to:

- Accept a single input image
- And output a classification/label for that image

C. Methodologies of Problem solving

We use real-time images or stream videos from CCTV or any other video capturing device or user can put pre recorded or captured video for analysis.

Our system is boosted with widely used classification methods to detect and identify faces even from a blur images or scrappy videos.

In order to make it functional on a regular computational CPU we made some functional changes in the dlib facial recognition network, now we can output the feature vector with 128-d i.e., a list of 128 real-valued numbers that our code will used to quantify the face. While we training network, we use triplets:

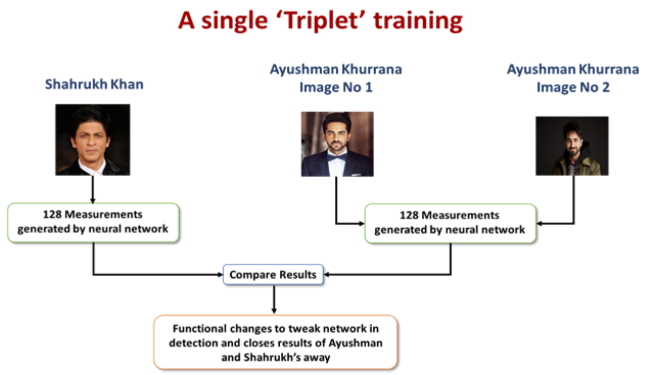

A single 'Triplet' training
Shahrukh Khan
Ayushman Khurrana Image No 1
Ayushman Khurrana Image No 2
128 Measurements generated by neural network
128 Measurements generated by neural network
Compare Results
Functional changes to tweak network in detection and closes results of Ayushman and Shahrukh's away

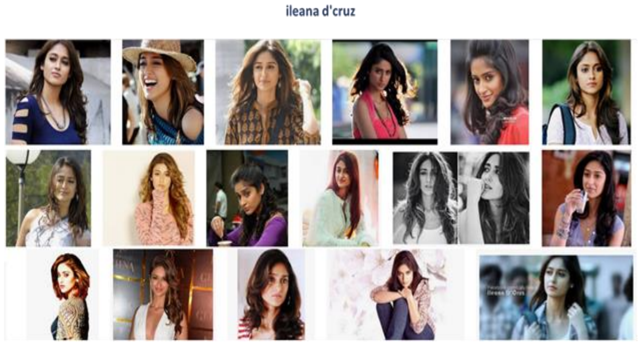
Figure 1: Block diagram of unique deep learning "triplet training."

In this demonstration we used three unique faces of popular Cinema Actor. The same functional triplet training consists of the dataset of unique images to train from. With the tradition we used SVM to generate 128-d vector for all images in dataset.

- In this we asked SVM to calibrate 128-d for above 3 images where the images of Ayushman are more than 2, hence defined algorithm sense's it as 2 similar faces in dataset.
- Although Shaharukh's image is a random pick from our data set and as not the same as Ayushman
- Our optimized network quantifies the faces and constructs the 128-d embeddings or quantification for each of the images in process.

IV. DATASETS WE USED TO DO FACE RECOGNITION

In this demonstration we used three unique faces of popular Cinema Actor. The same functional triplet training consists of the dataset of unique images to train from. With the tradition we used SVM to generate 128-d vector for all images in dataset.



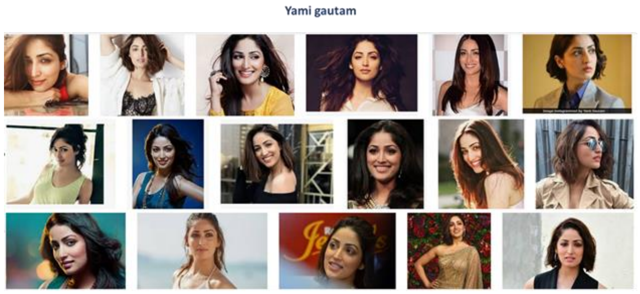


Figure2: Popular faces inspired dataset used in demonstration

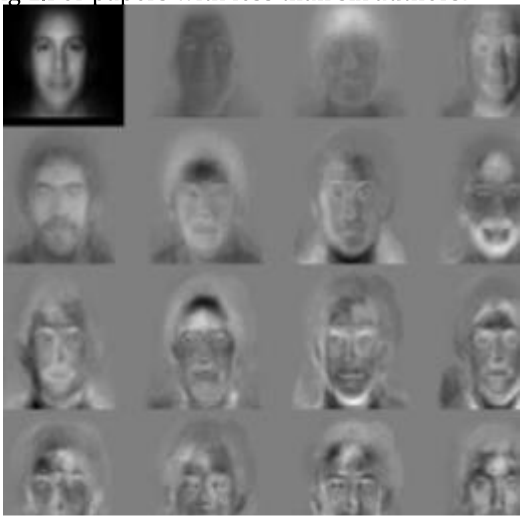
- Shahrukh Khan (16 images)
- Ayushman Khurrana (16 images)
- Ileana d'cruz (18 images)
- Yami Gautam (17 images)

The dataset we used and images we inputted into that are very random and having variety in color, shape, size and even in quality of pixels. This kind of variation is kept to ensure the quality and accuracy in face detection in various condition.

A. *Face Recognition Techniques- Appearance Based Approaches*

The Eigen face Method Firstly Kirby and Sirvoich demonstrated Eigenfaces method for recognition. Pentland and Turk made improvements on this research by employing Eigenfaces method based on Principle Component Analysis for the same reason

PCA is a Karhumen-Loeve transformation. PCA is a realized linear dimensionality reduction method used to determine a set of mutually orthogonal basis functions and as shown in fig 1.For papers with less than six authors:



maximum variance in g dimensional space and g is too big according to h. Subtracting the normalized training images from the calculated mean images thus mean centered images are calculated. If w is mean centered training image matrix Wi(i=1,2,........,L) and l is the number of training images, matrix d is calculated from as in equation 1

D = WWT

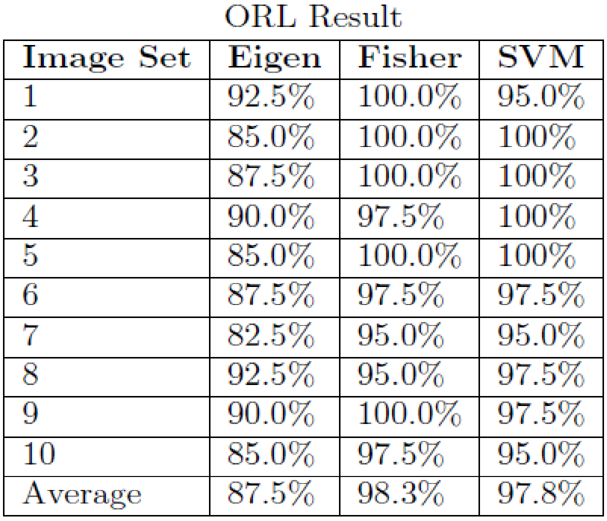
To reduce the size of covariance matrix D, we can use D = WTW instead. Eigenvectors ei and eigen values _i are obtained from covariance matrix.

Zi = ETwi(i = 1, 2, ...,L)

In the equation 2, Zi represents the new feature vector of lower dimensional space. Negative aspect of this method, it tries to max inter and intra class scattering. Inter class scattering is good for classification where intra scattering is not. If there is variance illumination, increases intra class scattering very high, even classes seems stained.

The Fisher face Method Belhumeur introduced the Fisher Face method in 1997, a derivative of Fishers Linear Discriminant (FLD) which has linear discriminant analysis (LDA) to gain the vast discriminant structures. Both PCA and LDA which are used to produce a subspace projection matrix is similar to eigen face and Fisher face methods. LDA describe a pair of projection vectors which form the maximum between-class scatter and minimum in the class scatter matrix concurrently

Example of Six Classes Using LDA



ORL Result

Image Set	Eigen	Fisher	SVM
1	92.5%	100.0%	95.0%
2	85.0%	100.0%	100%
3	87.5%	100.0%	100%
4	90.0%	97.5%	100%
5	85.0%	100.0%	100%
6	87.5%	97.5%	97.5%
7	82.5%	95.0%	95.0%
8	92.5%	95.0%	97.5%
9	90.0%	100.0%	97.5%
10	85.0%	97.5%	95.0%
Average	87.5%	98.3%	97.8%

Yale Result

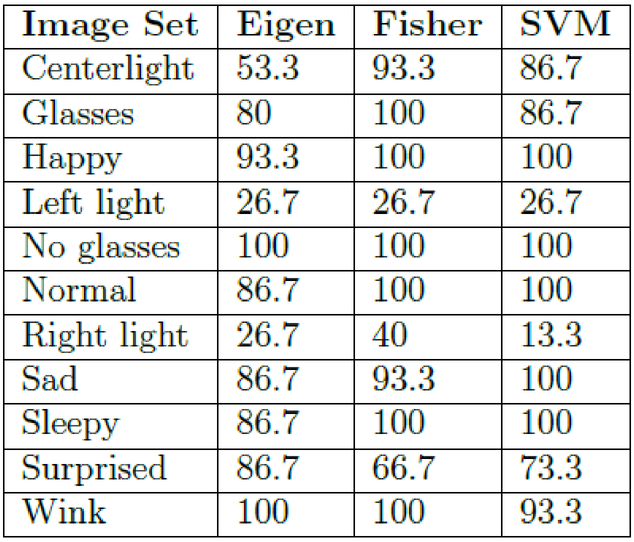
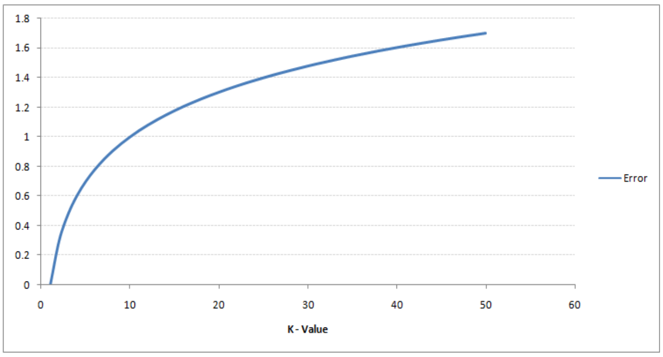


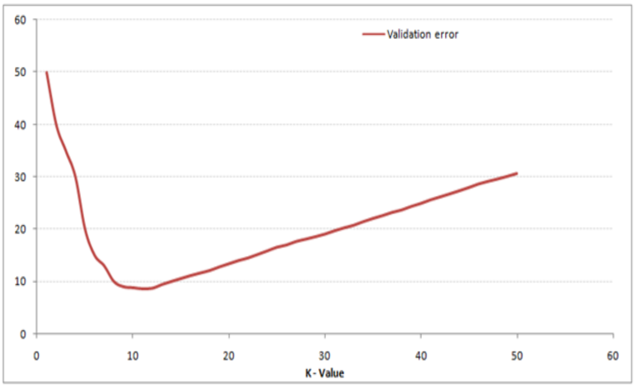
Image Set	Eigen	Fisher	SVM
Centerlight	53.3	93.3	86.7
Glasses	80	100	86.7
Happy	93.3	100	100
Left light	26.7	26.7	26.7
No glasses	100	100	100
Normal	86.7	100	100
Right light	26.7	40	13.3
Sad	86.7	93.3	100
Sleepy	86.7	100	100
Surprised	86.7	66.7	73.3
Wink	100	100	93.3

Following is the curve for the training error rate with varying value of K:



you can see, the error rate at K=1 is always zero for the training sample. This is because the closest point to any training data point is itself.Hence the prediction is always accurate with K=1. If validation error curve would have been similar, our choice of K would have been 1.

Following is the validation error curve with varying value of K:

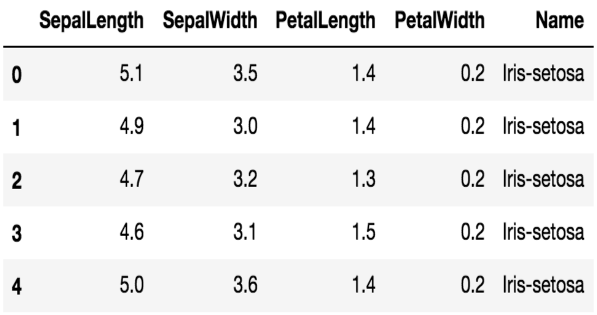


This makes the story more clear. At K=1, we were overfitting the boundaries. Hence, error rate initially decreases and reaches a minima. After the minima point, it then increase with increasing K. To get the optimal value of K, you can segregate the training and validation from the initial dataset. Now plot the validation error curve to get the optimal value of K. This value of K should be used for all predictions.

Breaking it Down – Pseudo Code of KNN

We can implement a KNN model by following the below steps:

```
1. Load the data
2. Initialise the value of k
3. For getting the predicted class, iterate from 1 to
   total number of training data points
    1. Calculate the distance between test data
       and each row of training data. Here we
       will use Euclidean distance as our
       distance metric since it's the most popular
       method. The other metrics that can be
       used are Chebyshev, cosine, etc.
    2. Sort the calculated distances in ascending
       order based on distance values
    3. Get top k rows from the sorted array
    4. Get the most frequent class of these rows
    5. Return the predicted class
```



	SepalLength	SepalWidth	PetalLength	PetalWidth	Name
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

Fisher face or Linear Discriminant Analysis (LDA) aims to increase inter class differences and are not used to increase data representation

$$S_w = \sum_{j=1}^{R} \sum_{i=1}^{M} (x_i^j - \mu_j)(x_i^j - \mu_j)^T$$

$$S_b = \sum_{j=1}^{R} (\mu_j - \mu)(\mu_j - \mu)^T$$

IV. EXPERIMENTAL RESULT

Tree Structure of the proposed system

```
$ tree --filelimit 10 --dirsfirst
.
├── dataset
│   ├── Shahrukh Khan (16 images)
│   ├── Ayushman Khurrana (16 images)
│   ├── ileana d'cruz (18 images)
│   ├── Yami Gautam (17 images)
│   ├── sample_1 [36 entries]
│   └── sample_2 [35 entries]
├── examples
│   ├── example_01.png
│   ├── example_02.png
│   └── example_03.png
├── output
│   └── lunch_scene_output.avi
├── videos
│   └── lunch_scene.mp4
├── search_bing_api.py
├── encode_faces.py
├── recognize_faces_image.py
├── recognize_faces_video.py
├── recognize_faces_video_file.py
└── encodings.pickle

10 directories, 11 files
```

In this project we enlist four top folders layers where we traverse all data and get orientation of the entire project

- **dataset/ :** This is the top most folder serve as a main training set for the flow and detection mechanism, Contains face images of subjects which we wanted to match
- **examples/ :** These are the images of unknown object to test by our trained network to assure detection and accuracy for the same these images are that are **not** in the dataset.
- **output/ :** our system is a Input based centralized system where we generate our output in a specific location this directory serve as a **Output Path** for our investigated contain like images or videos which are rendered by project.
- **videos/ :** As imported the Deep Learning in our project as a part of that Our project can detect images form video and the same will be rendered and stored as an Input video in this folder

  We have our worker files which will do all hard work for us

encode_faces.py: Encodings (128-d vectors) for faces
recognize_faces_image.py: Recognize faces
recognize_faces_video.py: Recognize faces in a live video stream
recognize_faces_video_file.py: Recognize faces in a video file from save video file
encodings.pickle : Facial recognitions encodings to form vector from dataset

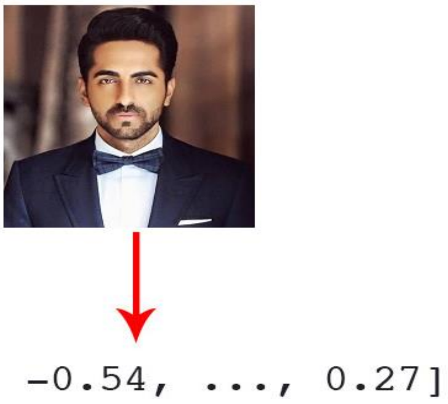


Figure 3: The hard work of Facial recognition using the module

As the critical process we must do some background work in order to streamline our activities and loads up images and videos for face detection and recognition.

As it is a input based activity first we need to enlist our source images or videos before we can recognize faces from images and videos which will be as a mode of operandi.

In order to established a connection between or dataset and network we must quantify the faces in training set. Right at this movement we are not training our module or network but we are making sure it has been done already to create 128-d embeddings

V. CONCLUSION

As the critical process we must do some background work in order to streamline our activities and loads up images and videos for face detection and recognition.

As it is a input based activity first we need to enlist our source images or videos before we can recognize faces from images and videos which will be as a mode of operandi.

In order to established a connection between or dataset and network we must quantify the faces in training set. Right at this movement we are not training our module or network but we are making sure it has been done already to create 128-d embeddings.

As our system is now detecting images successfully and we have created our 128-d face embeddings for each image in our dataset, Now We attempt to match each face in the input image (encoding) to our known encodings dataset (held in data["encodings"])

This function returns a list of events which are having True / False values, one by one for each image in our dataset On other hand our pre-processors are Internally are computing the Euclidean distance between the candidate embedding and all faces in our dataset using the compare_faces function

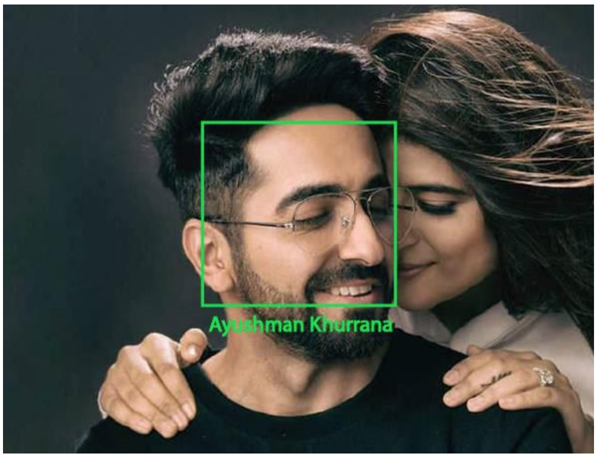


Figure 4: System has successfully recognized Ayushman Khurrana's face

Embedding an all faces in our dataset using the compare_faces function

- The purpose of this comparison is to detect the distance which is below our tolerance ratio which is auto set by CPU or GPU clock rate returning the True value which indicates the faces matched.
- And if the distance is above the tolerance threshold it will return False values which means we have **not match.**
- Along side of this we also able to show name of the face as it given in dataset The name variable will eventually hold the name string of the person
- This name variable is decided on the votes system will receive from pre processors while computing all datasets for operation the number of True values associated with each name which will eventually tally up and select the person's name which is having the most corresponding votes.

VI. REFERENCE

[1] Ahmet zdil Metin Mete zbilen A Survey on Comparison of Face Recognition Algorithms IEEE 2015

[2] G. Ramkumar, M. Manikandan - Face Recognition Survey International Journal of Advances in Science and Technology (IJAST) 2014 ISSN 2348-5426

[3] Sharkas, M. Abou Elenien, Eigenfaces vs. Fisherfaces vs. ICA for Face Recognition; A Comparative Study, 9th International Conference on Signal Processing, 2008, ICSP 2008., 2008, pp. 914919

[4] K. Kim, Intelligent Immigration Control System by Using Passport Recognition and Face Verification, in International Symposium on Neural Networks Chongqing, China, 2005, pp.147-156.

[5] J. N. K. Liu, M. Wang, and B. Feng, iBotGuard: an Internetbased intelligent robot security system using invariant face recognition against intruder, IEEE Transactions on Systems Man And Cybernetics Part C-Applications And Reviews, Vol.35, pp.97-105, 2005.

[6] H. Moon, Biometrics Person Authentication Using Projection- Based Face Recognition System in Verification Scenario, in International Conference on Bioinformatics and its Applications. Hong Kong, China, 2004, pp.207-213.

[7] D. McCullagh, Call It Super Bowl Face Scan 1, in Wired Magazine, 2001.

[8] CNN, Education School face scanner to search for sex offenders. Phoenix, Arizona: The Associated Press, 2003.

[9] P. J. Phillips, H. Moon, P. J. Rauss, and S. A. Rizvi, The FERET Evaluation Methodology for Face Recognition Algorithms, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.22, pp.1090-1104, 2000.

10] T. Choudhry, B. Clarkson, T. Jebara, and A. Pentland, Multimodal person recognition using unconstrained audio and video, in Proceedings, International Conference on Audio and Video- Based Person Authentication, 1999, pp.176-181.

[11] S. L. Wijaya, M. Savvides, and B. V. K. V. Kumar, Illumination-tolerant face verification of low-bit-rate JPEG2000 wavelet images with advanced correlation filters for handheld devices, Applied Optics, Vol.44, pp.655-665, 2005.

[12] E. Acosta, L. Torres, A. Albiol, and E. J. Delp, An automatic face detection and recognition system for video indexing applications, in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Vol.4. Orlando, Florida, 2002, pp.3644-3647.

[13] J.-H. Lee and W.-Y. Kim, Video Summarization and Retrieval System Using Face Recognition and MPEG-7 Descriptors, in Image and Video Retrieval, Vol.3115.

[14] Lecture Notes in Computer Science : Springer Berlin / Heidelberg,2004, pp.179-188.

[15] C. G. Tredoux, Y. Rosenthal, L. d. Costa, and D. Nunez, Face reconstruction using a configural, eigenface-based composite system, in 3rd Biennial Meeting of the Society for Applied Research in Memory and Cognition (SARMAC). Boulder, Colorado, USA, 1999.

[16] K. Balci and V. Atalay, PCA for Gender Estimation: Which Eigenvectors Contribute? In Proceedings of Sixteenth International Conference on Pattern Recognition, Vol.3. Quebec City, Canada, 2002, pp. 363-366.

[17] B. Moghaddam and M. H. Yang, Learning Gender with Support Faces, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.24, pp.707-711, 2002.

[18] R. Brunelli and T. Poggio, HyperBF Networks for Gender Classification, Proceedings of DARPA Image Understanding Workshop, pp.311-314, 1992.

[19] A. Colmenarez, B. J. Frey, and T. S. Huang, A probabilistic framework for embedded face and facial expression recognition, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vol.1. Ft. Collins, CO, USA, 1999, pp. 1592-1597.

[20] Y. Shinohara and N. Otsu, Facial Expression Recognition Using Fisher Weight Maps, in Sixth IEEE International Conference on Automatic Face and Gesture Recognition, Vol.100, 2004, pp.499-504.

[21] F. Bourel, C. C. Chibelushi, and A. A. Low, Robust Facial Feature Tracking, in British Machine Vision Conference. Bristol, 2000, pp.232-241.

[22] K. Morik, P. Brockhausen, and T. Joachims, Combining statistical learning with a knowledge based approach – A case study in intensive care monitoring, in 16th International Conference on Machine Learning (ICML-99). San Francisco, CA, USA: Morgan Kaufmann, 1999, pp.268-277.

[23] S. Singh and N. Papanikolopoulos, Vision-based detection of driver fatigue, Department of Computer Science, University of Minnesota, Technical report 1997.

[24] D. N. Metaxas, S. Venkataraman, and C. Vogler, Image-Based Stress Recognition Using a Model-Based Dynamic Face Tracking System, International Conference on Computational Science , pp.813-821, 2004.

[25] M. M. Rahman, R. Hartley, and S. Ishikawa, A Passive And Multimodal Biometric System for Personal Identification, in International Conference on Visualization, Imaging and Image Processing Spain, 2005, pp.89-92.

[26] R. Brunelli and D. Falavigna, Person identification using multiple cues, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol.17, pp.955-966, 1995.

[27] M. Viswanathan, H. S. M. Beigi, A. Tritschler, and F. Maali, Information access using speech, speaker and face recognition.