Suspect's Presence Identification using Convolution Neural Network

Amit Kumar[1], Revanth Ravadagundi[2], Pawan Kumar Jaiswal[3], Abha Kumari[4]

[1]Department of ECE, NIT Srinagar, Hazratbal-190006, India
[2]Department of ECE, BMS Institute of Technology and Management, Bengaluru-560064, India
[3]Department of EE, MIT, Muzaffarpur, Bihar-842003, India
[4]Department of ECE, MIT, Muzaffarpur, Bihar-842003, India
E-mail:[1]amit.kumarc210@gmail.com, [2]revanthravadagundi9@gmail.com, [3]pawankumar.jaiswal@gmail.com, [3]kmrabha@gmail.com.
.

ABSTRACT: Several studies have been reported on face recognition for the past two decades. Face recognition is widely used in suspected person tracking. Tracking a suspected person to identify terrorist activity is always important for protecting the world. The entire world is currently facing problems due to the coronavirus pandemic. To avoid the spread of suspected coronavirus-positive persons, tracking is an important process. The spread of the coronavirus is greater when a suspected or confirmed person attends large-crowd events. In such cases, analysis of the patent's presence in that event is important. And all the other persons presented thereafter are suspected persons. Therefore, in this paper, an attempt has been made to track the person's presence in an event by processing only one image using the Google method named "FACENET". Here, a convolutional neural network is used for the identification, along with 128 Euclidean distance measurements. The video is taken from YouTube. The accuracy of this method is 90% with a single image. The tabular results for the suspected presence are given.

INDEX TERMS COVID-19, CNN, Image processing, FACENET.

I. INTRODUCTION

Image processing is one of the most important and highly demanding areas of research for the last few decades. Artificial intelligence (AI) techniques and image-processing methods are widely used across sectors such as sign language recognition, face recognition, agriculture, and satellite engineering.[11]. A convolutional neural network (CNN) is one of the widely used methods in AI [28]. CNNs have been applied in several areas, including biomedical signal processing [7], biomedical image processing [13, 18], agricultural development [1], satellite image processing [27], and face recognition [3, 25]. The COVID-19 pandemic is one of the most recent global problems. Many studies have been conducted to address the pandemic [10]. These studies aim to generate data on the vaccine, COVID-19 symptoms, the effect of temperature on COVID-19 across different ages, the impact of lockdowns on COVID-19 spread and other health issues, post-COVID-19 symptoms, etc. [6]. Tracking COVID-19 patients is an important way to prevent symptoms. Tracking must be done to stop the spread of coronavirus, so it is always important to monitor the patient's attendance at events with large crowds. The lockdown imposed by many countries led to economic recession and job losses. So, imposing a long lockdown is not considered a good solution [18]. Person tracking can be performed using image processing, face recognition, and video processing. Face recognition can be performed using several techniques, such as Laplacian faces [22], CNN [13, 14, 24, 25], Face Recognition Technology (FERET) [9], LDA-based algorithms [8], etc. For the last decade, several studies have been done to identify the face. Face recognition using machine learning and deep learning

provides better results in comparison to several other methods [19]. In a study, the authors have performed face identification using continuous-density Hidden Markov Models (HMMs) [12]. Here, researchers have used the stochastic model to encode useful information. Flexible models are designed to learn the appearance of the human face through face identification, where the problem of determining whether two face images depict the same person remains persistent. Recognition of faces is a difficult task because different images of different persons vary in pose, background, scale, hairstyle, expression, and glasses [4]. To address these constraints, authors have introduced two new methods for learning robust distance measures. In the first method, they have used a logistic discriminant approach. This approach is used to learn the metric from a set of labeled image pairs (LDML). And in the second approach, the nearest neighbor approach (NNA), where probability computation is done using two images of the same class. The authors reported results of 79.3% and 87.5% correct in the restricted and unrestricted settings, respectively, significantly improving on the current state-of-the-art of 78.5%. Approaches based on nearest neighbors (NN) and deep learning require large datasets. In this work, a single image is needed to identify the suspect and determine the time of their presence [5]. The paper is structured as follows: Section 1 includes a brief literature review of the importance; in Section 2, the method, including CNN, Euclidean distance, and data set details, is described. Section 3 includes the methodology used in this work. In Section 4, the results and discussion have been presented. The study's conclusion is presented in Section 5.

2. Methods and material

In this study, 128-D Euclidean vectors and CNN have been used; brief details of both have been given below:

2.1 Convolution Neural Network

Convolutional neural networks are also called shift-invariant or space-invariant artificial neural networks (SIANN). CNNs are a class of deep neural networks (DNNs), used to analyze visual imagery. CNNs involve a shared weight architecture and translation invariance characteristics. CNNs are used in several research fields. In CNNs, multilayer perceptrons are used for regularization; therefore, these networks are fully connected. In CNNs, each neuron in a layer is connected to all neurons in the next layer. The main advantage of using CNNs is that they exploit hierarchical structure in data and assemble more complex patterns from smaller, simpler ones. Therefore, on the scale of connectedness and complexity, CNNs are on the lower extreme [5].

In Fig. 1, the neural network's layer structure is shown. Here, the layers, Convolution Layer, Activation Layer, Convolutional Layer, Pooling Layer, and Classification Layer are presented. The convolution layer accepts directly raw images as input, where a set of small filters is convolved over the image to produce one or more feature maps. Sliding filtering is performed by convolving the filter with the image, computing the dot product of the filter's elements with the image's elements [23]. This process extracts specific features from the image [24]. After that, an activation layer is used to take the convolutional image outputs. In most cases, the CNN uses the Rectified Linear Unit (ReLU), which converts negative values to 0.

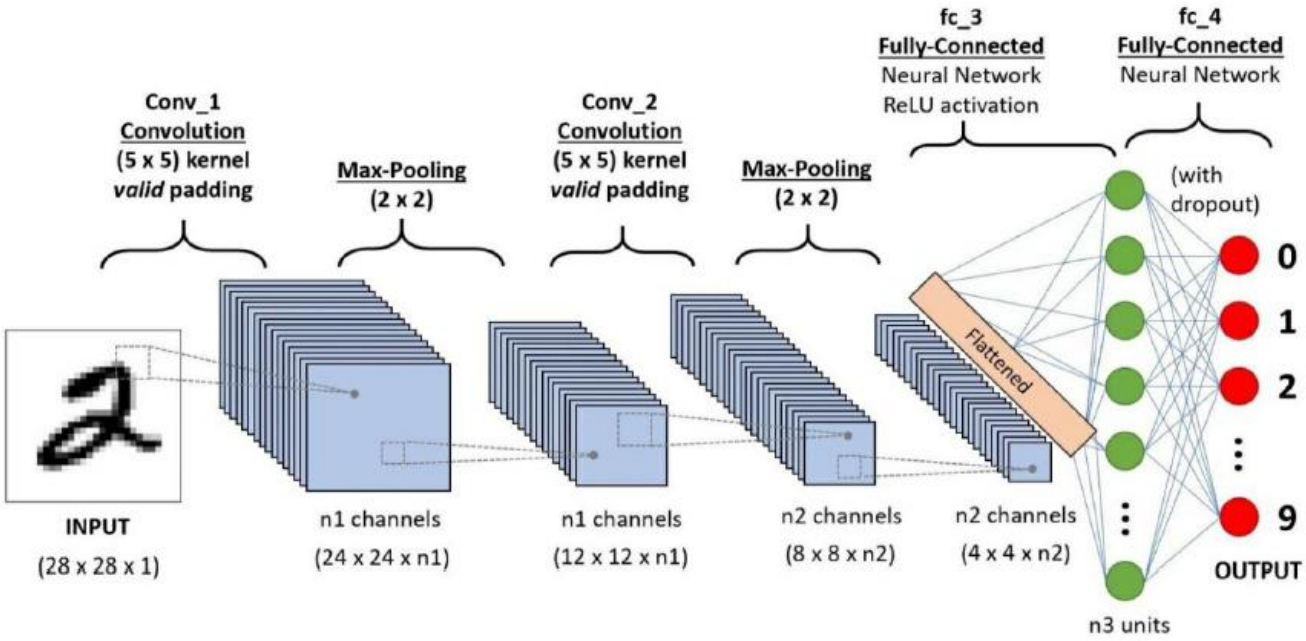

Fig. 1 Convolution Neural Network Layer [31]

To reduce the size of an image, a pooling layer is used. Here, the data is downsampled along each dimension [1]. The most popular types of pooling are average pooling and max pooling. After the pooling layer, a classification layer is used for classification [24, 27].

2.2 Histogram of oriented gradients (HOG):

Description of image features after amassing is a very important part of image analysis and is a feature descriptor used to detect objects in computer vision and image processing. Histograms of oriented gradients (HOG) can be used to describe image features. In this technique, counts of gradient orientation are used in localized portions of an image-detection window or region of interest (ROI) [29]. The following are the steps for implementing the HOG descriptor algorithm:

Step 1: Image segmentation:

Step 2: Image discretization: Discretize each cell into angular bins according to the gradient orientation.

Step 3: Pixel contribution: Each cell's pixel contributes a weighted gradient to its corresponding angular bin.

Step 4: Block formation:

Step 5: Histogram normalization:

2.3. Euclidean vectors

Euclidean/ geometric/ special vector is a vector (a geometric object that has magnitude/ length and direction) used in engineering, physics, and mathematics. It is similar to a vector in vector algebra; i.e., it can be added to other vectors. A line and arrow of direction can represent it. As indicated in Fig. 2, A and B are two points, and the blue lines connect them [2].

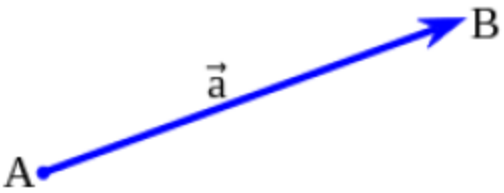

Fig. 2. Euclidean vector

2.4. Facial recognition

Previous face recognition approaches based on deep networks. Some of these then combine the output of a CNN with PCA for dimensionality reduction and SVM for classification. Approaches such as those of [30] and the DeepFace group at Facebook [17] first "warp" or "align" faces into a more amenable form (either 'canonical frontal view' or DeepFace's general 3D model) and then learn a CNN to classify each face as belonging to an identity. The architectures explored using FACENET are based on either the [26] model or the Inception [16] model (which won the ImageNet competition in 2014).

2.5. FACENET overview

FACENET uses a deep convolutional network to learn a Euclidean embedding for each image [32, 33, 34]. The network is trained so that face similarity is directly correlated with the squared L2 distances in the embedding space: faces of different persons have big distances, whereas faces of the same person have small distances. The aforementioned tasks become simple once this embedding is created: face verification requires thresholding the distance between the two embeddings; recognition becomes a k-NN classification problem; and clustering can be performed using commercially available methods such as k-means or agglomerative clustering. A classification layer [15, 17] trained over a set of known face identities is used in earlier deep network-based face recognition techniques. An intermediate bottleneck layer is then used as a representation to generalize recognition beyond the set of identities used in training.

The indirectness and inefficiency of this approach are its drawbacks: the representation size per face is typically very big (1000s of dimensions) when employing a bottleneck layer, and one must trust that the bottleneck representation generalizes effectively to new faces. PCA has been used in some recent work [15] to reduce dimensionality; it is a linear transformation that can be learned in a single network layer. Unlike these methods, FACENET uses a triplet-based loss function based on LMNN to train its output to directly produce a compact 128-D embedding.

The loss seeks to distinguish the positive pair from the negative pair by a distance margin. Our triplets are made up of two matching face thumbnails and one non-matching face thumbnail. Except for size and translation, the thumbnails are tight crops of the face region with no 2D or 3D alignment. Inspired by curriculum learning, we describe a unique online negative exemplar mining technique that ensures continually increasing triplet difficulty as the network trains. Selecting the right triplets turns out to be crucial for attaining good performance. We also investigate hard-positive mining methods that promote spherical clusters for a single person's embeddings to increase clustering accuracy.

2.6. Triplet loss

The triplet-based loss function used to learn the mapping is an adaptation of Kilian Weinberger's Large Margin Nearest Neighbor (LMNN) classifier [21] (which repeatedly pulls together images of the same person and simultaneously pushes images of any different person away) to deep neural networks. [15] Use ensembles of networks trained using a combination of classification and verification loss. The verification loss they use is similar to the triplet loss used to learn the mapping used by FACENET in that it minimizes squared L2 distances between images of faces from the same person and enforces a margin separating images of faces from a different person, but it's different in that only pairs of images are compared, whereas the triplet loss encourages a relative distance constraint by looking at three at a time. A loss similar to FACENET's triple loss was used by [20] to rank images based on semantic and visual similarity.

The embedding is represented by f(y) ∈ R^d. It embeds an image *y* into a d-dimensional Euclidean space. Additionally, we constrain this embedding to live on the d-dimensional hypersphere, i.e., ||f(y)||2 = 1. The nearest-neighbor classification serves as the motivation for this loss. In this case, we want to make sure that a picture y_i^a (anchor) of a particular individual is closer to all other images y_i^p (positive) of the same individual than it is to any image y_i^n (negative) of any other individual. Fig. 3 illustrates this. Therefore, we desire,

$$\left\|y_i^a - y_i^p\right\|_2^2 + \alpha < \left\|y_i^a - y_i^n\right\|_2^2, \forall \left(y_i^a, y_i^p, y_i^n\right) \in \mathrm{T} \quad (1)$$

where α is a margin that is enforced between positive and negative pairs. T is the set of all possible triplets in the training set and has cardinality N. The loss that is being minimized is then

$$\mathrm{L} = \sum_i^N \left[\left\|f(\mathrm{y}_i^{\mathrm{a}}) - f\left(y_i^p\right)\right\|_2^2 - \left\|f(\mathrm{y}_i^{\mathrm{a}}) - f(y_i^n)\right\|_2^2 + \alpha\right] \quad (2)$$

Many triplets that are easily satisfied (i.e., satisfy the constraint in equation (1)) would be produced if all feasible triplets were generated. Since these triplets would still be sent across the network, they would not aid in training and would hinder convergence. Choosing hard triplets that are active and can thus enhance the model is essential. The following section discusses the different approaches we use for triplet selection.

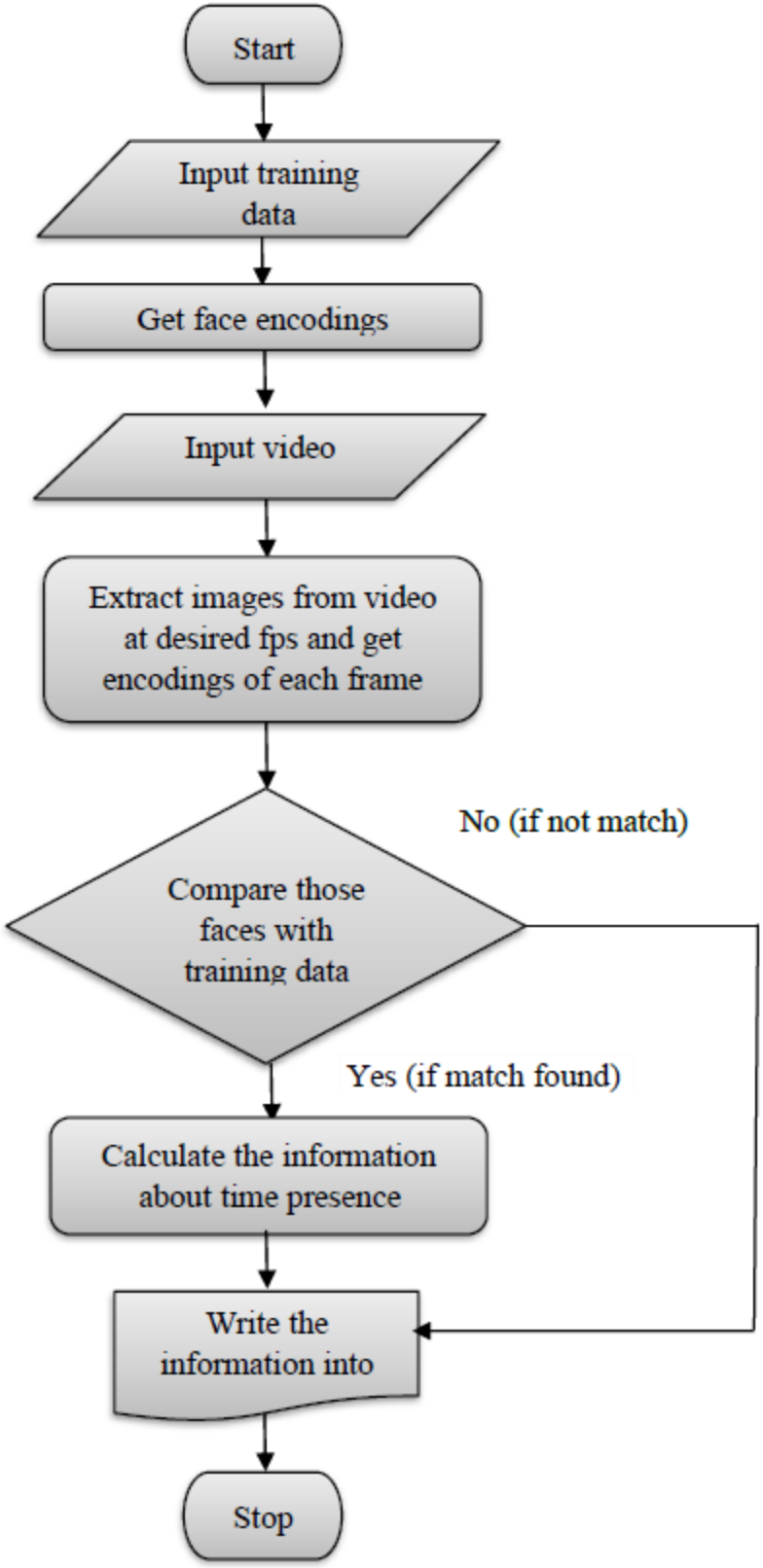

Fig. 3. Flow chart of the methodology

2.7. Triplet selection

To ensure a quick connection, it is important to select triplets that break the triplet limit in Eq. (1). This means that, given the $y_i^a yia$, we want to select the file for $y_i^p yip$ (hard positive) as argmax y_i^p || f(y_i^a) − f(y_i^p) || 2^2 similarly yin (hard negative) similar to argmin y_i^n || f(y_i^a) − f(y_i^p) || 2^2 . It is not possible to compute argmin and argmax over the entire training set. In addition, it can lead to negative training, as faces with the wrong words and negative images will dominate the good and the bad. Two obvious options to avoid this problem:

• Generate offline triplets every step in n steps, using the most recent network test site and using computer arcmin and argmax in the data set.

• Produce three triplets online. This can be done by selecting hard/encouraging hard models within the mini-batch.

Here, we simply count argmin and argmax within each mini-batch and concentrate on creating and using huge mini-batches of several thousand samples online. In order to obtain a realistic depiction of peer-to-peer distances, it is necessary to ensure that each subgroup contains only a very small number of single-identity models. In our experiment, we sampled training data to select about 40 faces per minibatch. Additionally, the sample back surface can be added to each mini-batch from time to time.

Instead of choosing the best one, we use every anchor-positive pair in a mini-batch to select the hardest one. We do not have a close comparison of anchor-positive pairs within a mini-batch, but we found that treating the entire anchor-positive set as a single batch made the method stable and yielded very good initial performance quickly.

We also tested the offline production of triplets in conjunction with online generation, which would allow the use of smaller batch sizes, but testing was incomplete. Selecting the worst items in practice can lead to local minima at the start of training; in particular, it can result in a falling model (e.g., f(y) = 0). To minimize this, it is helpful to select such y_i^n.

$$\left\|f(\mathrm{y}_i^{\mathrm{a}}) - f(y_i^p)\right\|_2^2 < \left\|f(\mathrm{y}_i^{\mathrm{a}}) - f(y_i^n)\right\|_2^2 \quad (3)$$

Since small mini-batches tend to improve convergence during Stochastic Gradient Descent (SGD), we would want to employ them [31].

3. Methodology

The methodology for determining a person's presence follows these steps.

Step 1: This step is all about collecting the training images.
In this work, the method is designed using a CNN to test it. Videos were taken from YouTube; the links to the videos are given in the table. (If the light and intensity are good, we have taken the video from the camera). After that, a single image has been used to track both time and the person's image.

Step 2: Encode the collected faces into 128-D Euclidean vectors.
After training, a face encoder is used to encode the image into 128-D Euclidean vectors.

Step 3: Selection of a video to test our model.
After training the model for video recognition, the video is fed to the model. Here, the video can be of any length.

Step 4: Break the video into images, name them with the person's name and the frame number with respect to. the video, and store them in a folder.

Step 5: Get the images from a folder in the order of their frame numbers, then encode them into 128-D Euclidean vectors and compare them with the already encoded faces. If a match is found, store the image name in memory.

Step 6: Now, perform a calculation about how many times a person's name is encountered. This gives the number of seconds the person is present in the video. These steps are also given in Fig. 3.

4. Results and Discussions

Table 1. Person's presence duration using brighter video

Sl. no	Video link	Name of person	Time duration of the presence
1	https://www.youtube.com/watch?v=3EXXkAA8vSk	Ivanka	Initially seen at 0 seconds
			Finally seen at 43 seconds.
			The total presence is 43 seconds.
2	https://youtu.be/_Qq6dQwLh1s	Chris Evans	Initially seen at 1 second
			Finally seen at 52 seconds.
			The total presence is 5.5 seconds.
3	https://youtu.be/_Qq6dQwLh1s	Jermy Renner	Initially seen at 0 seconds
			Finally seen at 36.5 seconds.
			The total presence is 4.5 seconds
4	https://youtu.be/_Qq6dQwLh1s	Mark Rufalo	Initially seen at 20.5 seconds.
			Finally seen at 58.5 seconds.
			The screen time is 7.5 seconds
5	https://youtu.be/_Qq6dQwLh1s	Robert Downey	Initially seen at 29 seconds.
			Finally seen at 59 seconds.
			Total presence is 3 seconds.
6	https://youtu.be/_Qq6dQwLh1s	Scarlett Johansson	Initially seen at 3 seconds
			Finally seen at 39.5 seconds.
			Total presence is 10.5 seconds.
7	https://www.youtube.com/watch?v=S_6vjb1cJkE (crowded video)	Will Smith	Initially seen at 8.5 seconds
			Finally seen at 3:15 seconds.
			Total presence is 1:17 seconds.
8	https://youtu.be/LdOM0x0XDMo (dark mode)	John David Washington	Initially seen at 23.5 seconds
			Finally seen at 1:55 seconds.
			Total presence is 9 seconds.

In this paper, four videos were used to test the method's performance. The presence of the suspect is given in Table 1. In this table, the suspect's screen time, as seen at the beginning and end of a video, has been provided. 12 images of John's absence have been shown in Fig. 4 from one of the YouTube videos.

For testing the method, brighter as well as darker videos are considered. Here, the accuracy for brighter videos is ~90%, and for darker videos, it is lower, around 85%. In this table, popular people, such as Ivanka, Chris Evans, Jeremy Renner, Mark Ruffalo, Robert Downey Jr., Scarlett Johansson, Will Smith, and John David Washington, are assumed to be suspects. Later, if we want, we can extract the frames where a particular person is present and form a video from them that shows the suspect. So, we can even get to know the people surrounding the suspect. This would help a lot in these pandemic days.

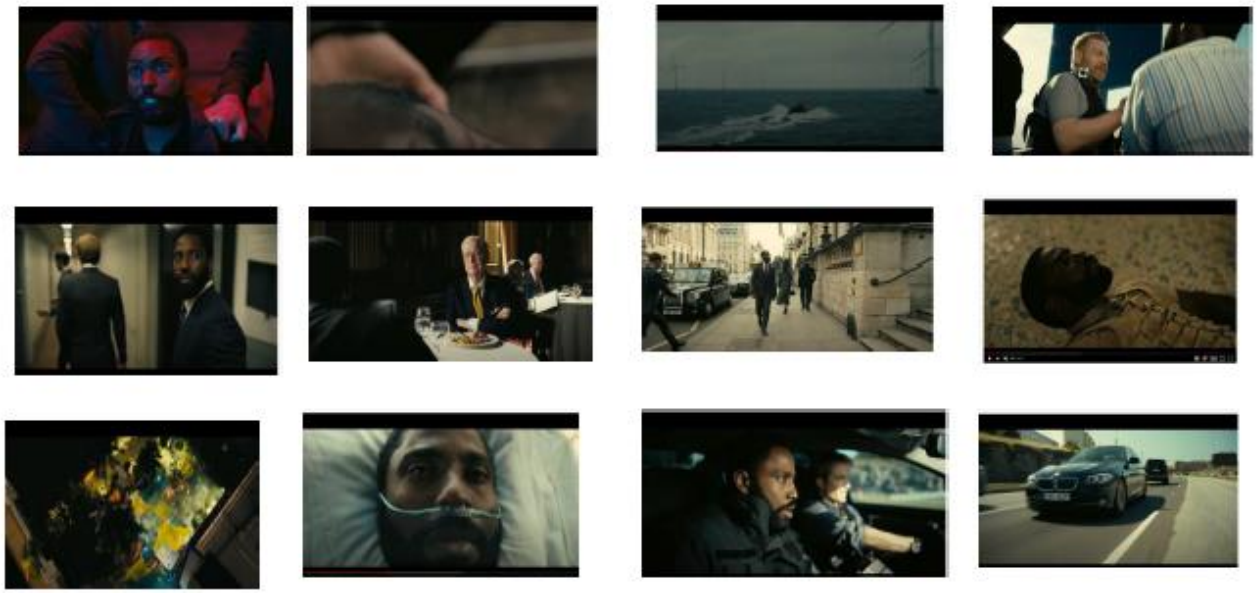

Fig. 4. 12 images of video given in https://youtu.be/LdOM0x0XDMo reflecting the presence/ absence of John

5. Conclusions

The performance is very good but not excellent. There are very few exceptions. While creating encodings, we can pass either "HOG" (Histogram of Gradients) or CNN ("Convolutional Neural Network") as an argument. CNNs are more accurate than HOGs, but are time-consuming and require a Graphical Processing Unit (GPU) to run. Generally, using a GPU is not preferred; therefore, it is better to stick with HOG. One of them, HOG, is available by default for encodings. From the tabular results, it can be observed that the proposed method can be used to measure the presence of any person. It can be applied for tracking the person present in the crowded area, which will help slow the spreading of the coronavirus, in terrorist activity by checking the suspect's connections and activity during the event, and one real-world application of this project can be found in Amazon Prime Video, where every actor in a scene is listed along with their images on the left side of the screen.

VI. REFERENCES

[1] Abdullahi, H.S. et al., "Convolution neural network in precision agriculture for plant image recognition and classification," In: 2017 Seventh International Conference on Innovative Computing Technology (INTECH). pp. 1–3 IEEE (2017). https://doi.org/10.1109/INTECH.2017.8102436.

[2] Cortés, V. et al., "Special geometry of euclidean supersymmetry 1. Vector multiplets," J. High Energy Phys. 8 (3) 593–665 (2004). https://doi.org/10.1088/1126-6708/2004/03/028.

[3] Eccv, A., "Leaving Some Stones Unturned : Dynamic Feature Prioritization for. Eur," Conf. Comput. Vis. 1, 1–7 (2016). https://doi.org/10.1007/978-3-319-46478-7.

[4] Gao, W. et al., "The CAS-PEAL large-scale chinese face database and baseline evaluations," IEEE Trans. Syst. Man, Cybern. Part ASystems Humans. 38 (1) 149–161 (2008). https://doi.org/10.1109/TSMCA.2007.909557.

[5] Huang, W., Yin, H., "Robust face recognition with structural binary gradient patterns. Pattern Recognit," 68, 126–140 (2017). https://doi.org/10.1016/j.patcog.2017.03.010.

[6] Jain, S., Sharma, T., "Social and travel lockdown impact considering coronavirus disease (Covid-19) on air quality in megacities of india: Present benefits, future challenges and way forward," Aerosol Air Qual. Res. 20 (6) 1222–1236 (2020). https://doi.org/10.4209/aaqr.2020.04.0171.

[7] Li, D. et al., "Classification of ECG signals based on 1D convolution neural network," 2017 IEEE 19th Int. Conf. e-Health Networking, Appl. Serv. Heal. 2017. 2017-Decem, 1–6 (2017). https://doi.org/10.1109/HealthCom.2017.8210784.

[8] Lu, J. et al., "Face recognition using LDA-based algorithms," IEEE Trans. Neural Networks. 14 (1) 195–200 (2003). https://doi.org/10.1109/TNN.2002.806647.

[9] Phillips, P.J. et al., "The FERET database and evaluation procedure for face-recognition algorithms," Image Vis. Comput. 16 (5) 295–306 (1998). https://doi.org/10.1016/s0262-8856(97)00070-x.

[10] Pierce, M. et al., "Mental health before and during the COVID-19 pandemic: a longitudinal probability sample survey of the UK population," The Lancet Psychiatry. 7 (10) 883–892 (2020). https://doi.org/10.1016/S2215-0366(20)30308-4.

[11] Robert J. Schalkoff: Digital image processing and computer vision. John Wiley & Sons, Inc., New York (1989).

[12] Samaria, F.S., Harter, A.C., "Parameterisation of a stochastic model for human face identification," IEEE Work. Appl. Comput. Vis. - Proc. 138–142 (1994). https://doi.org/10.1109/acv.1994.341300.

[13] Singh, P., Sehgal, P., "Numbering and Classification of Panoramic Dental Images Using 6-Layer Convolutional Neural Network," Pattern Recognit. Image Anal. 30 (1) 125–133 (2020). https://doi.org/10.1134/S1054661820010149.

[14] Staroletov, S.M. et al., "Development and Testing of Algorithms for Vehicle Type Recognition and Car Tracking with Photo and Video Traffic Enforcement Cameras," Pattern Recognit. Image Anal. 31 (2) 323–333 (2021). https://doi.org/10.1134/S1054661821020152.

[15] Sun, Y. et al., "Deeply learned face representations are sparse, selective, and robust," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 07-12-June, 2892–2900 (2015). https://doi.org/10.1109/CVPR.2015.7298907.

[16] Szegedy, C. et al., "Going deeper with convolutions. Proc," IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 07-12-June, 1–9 (2015). https://doi.org/10.1109/CVPR.2015.7298594.

[17] Taigman, Y. et al., "DeepFace: Closing the gap to human-level performance in face verification," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 1701–1708 (2014). https://doi.org/10.1109/CVPR.2014.220.

[18] Traore, B.B. et al., "Deep convolution neural network for image recognition," Ecol. Inform. 48 (2) 257–268 (2018). https://doi.org/10.1016/j.ecoinf.2018.10.002.

[19] Tripathi, B.K., "On the complex domain deep machine learning for face recognition," Appl. Intell. 47 (2) 382–396 (2017). https://doi.org/10.1007/s10489-017-0902-7.

[20] Wang, J. et al., "Learning fine-grained image similarity with deep ranking," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 1386–1393 (2014). https://doi.org/10.1109/CVPR.2014.180.

[21] Weinberger, K.Q., Saul, L.K., "Distance metric learning for large margin nearest neighbor classification," J. Mach. Learn. Res. 10, 207–244 (2009). https://doi.org/10.1145/1577069.1577078.

[22] Wu, Y., Gu, R.M., "A new subspace analysis approach based on laplacianfaces," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 4233 LNCS, 253–259 (2006). https://doi.org/10.1007/11893257_28.

[23] Xia, H., "Intelligence Science and Big Data Engineering. Image and Video Data Engineering," Springer International Publishing, Cham (2015). https://doi.org/10.1007/978-3-319-23989-7.

[24] Yang, Y.X. et al., "Face recognition using the SR-CNN model," Sensors (Switzerland) 18 (12) (2018). https://doi.org/10.3390/s18124237.

[25] Ye, S. et al., "Person Tracking and Reidentification for Multicamera Indoor Video Surveillance Systems," Pattern Recognit. Image Anal. 30 (4) 827–837 (2020). https://doi.org/10.1134/S1054661820040136.

[26] Zeiler, M.D., Fergus, R., "Visualizing and understanding convolutional networks," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 8689 LNCS, PART 1, 818–833 (2014). https://doi.org/10.1007/978-3-319-10590-1_53.

[27] Zhong, Y. et al., "SatCNN: satellite image dataset classification using agile convolutional neural networks," Remote Sens. Lett. 8 (2) 136–145 (2017). https://doi.org/10.1080/2150704X.2016.1235299.

[28] Zhou, X. et al., "An Efficient Compressive Convolutional Network for Unified Object Detection and Image Compression," Proc. AAAI Conf. Artif. Intell. 33, Romberg, 5949–5956 (2019). https://doi.org/10.1609/aaai.v33i01.33015949.

[29] Zhu, Q. et al., "Fast human detection using a cascade of histograms of oriented gradients," Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 2, 1491–1498 (2006). https://doi.org/10.1109/CVPR.2006.119.

[30] Zhu, Z. et al., "Recover Canonical-View Faces in the Wild with Deep Neural Networks," Comput. Vis. Pattern Recognit. 1–10 (2014).

[31] DataSciencepr, https://datasciencepr.com/convolutional-neural-network/.

[32] Schroff, F., Kalenichenko, D., & Philbin, J. (2015). FaceNet: A Unified Embedding for Face Recognition and Clustering. ArXiv. https://doi.org/10.1109/CVPR.2015.7298682.

[33] A. Autade et al., "Automated Multi-Face Recognition and Identification using Facenet and VGG-16 on Real-World Dataset for Attendance Monitoring System," 2023 7th International Conference On Computing, Communication,

[34] Control And Automation (ICCUBEA), Pune, India, 2023, pp. 1-5, doi: 10.1109/ICCUBEA58933.2023.10392198.

[35] Karamizadeh, S., Shojae Chaeikar, S., & Salarian, H. (2025). Combining MTCNN and Enhanced FaceNet with

[36] Adaptive Feature Fusion for Robust Face Recognition. Technologies, 13(10), 450.

[37] https://doi.org/10.3390/technologies13100450