Face Recognition based Surveillance System Using FaceNet and MTCNN on Jetson TX2

Edwin Jose
Department of Electronics
Cochin University of Science and Technology
Kochi, India
edwin.jose@cusat.ac.in

Greeshma M.
Department of Electronics
Cochin University of Science and Technology
Kochi, India
greeshma27@cusat.ac.in

Mithun Haridas T. P.
Department of Electronics
Cochin University of Science and Technology
Kochi, India
mithuntp@cusat.ac.in

Supriya M. H.
Department of Electronics
Cochin University of Science and Technology
Kochi, India
supriya@cusat.ac.in

***Abstract*— Surveillance systems, in spite of the recent advances, still poses many challenges, especially in the field of patrolling or tracking of subjects through CCTV footage or any other portable drone mechanisms. Real time monitoring of public places for possible suspects are still made through manual observations in many places. The manual labour involved and the human errors that can occur makes the system less efficient. Many research attempts were made for designing foolproof systems and still going on, understanding the importance of the problem. This paper presents the implementation of an intelligent multicamera Face Recognition based surveillance system using FaceNet and MTCNN algorithm on Jetson TX2. The proposed portable system tracks the subject or the suspect with the camera ID/location together with the timestamp and logs his presence in the database, using multiple camera installation.**

Keywords*— *FaceNet, Face Recognition, Convolutional Neural Network ,MTCNN, Jetson TX2, Surveillance system

I. INTRODUCTION

Ever since the advent of image sensory devices, cameras have dependably been the eyes of the security business, apart from the traditional motion detection sensors[1]. In most cases, video surveillance systems are fixed to infrastructure and usually specific for a location, but to implement a portable surveillance system [2] its required to have a highly accurate algorithm along with a great computational embedded device that can work with low power consumption[3]. With rapid technological developments in the past century there has been an explosive growth in the surveillance industry. This scenario while increasing the need for high-quality surveillance cameras reduced the cost of such surveillance devices[2]. Multi-camera surveillance is widely used in every possible scenario and is intended to reduce crime and improve public and private safety. The multi cameras work seamlessly leading to the capture of a stack of videos[1]. Monitoring and scrutinizing all sets of live cameras or footage by any human makes it inaccurate as well as inefficient[4]. The concept of deep learning techniques with its automated face recognition on Graphical Processing Unit (GPU) based processors becomes very relevant at this juncture. These algorithms with its quick response as well as computational intelligence overcomes the limitation of existing surveillance system.

II. RELATED WORKS

Surveillance systems have been implemented throughout the years on various devices and infrastructures, mostly being CPU and FPGA [5] based embedded systems. The more powerful and much efficient GPU[6] has been used to implement the principles of surveillance mechanism with the traditional algorithms of face recognition and object detection[3] with an accuracy which was nearly equal to 88 percentage[7]. But for the past few years, several deep learning architecture such as FaceNet [8] has shown a promising accuracy of nearly 99 percentage on the GPU based system. In this work, a face recognition-based surveillance system is proposed to be implemented on a powerful and efficient GPU based embedded device such as Jetson TX2 from Nvidia.

III. FACENET

FaceNet is a deep convolutional neural network developed by Google researchers and introduced around 2015 to effectively solve the hurdles in face detection and verification. The FaceNet algorithm transforms the face image into 128-dimensional Euclidean space similar to word embedding[9]. FaceNet model thus created is trained for triplet loss to capture the similarities and differences on the image dataset provided. The embeddings with 128-dimensions, created by the model could be used to cluster faces in, a much effective and precise manner. Using FaceNet embeddings as feature vectors, functionalities such as face recognition, and verification could be implemented after creating the vector space[10]. In short, the distances for the similar images would be much closer than the random non similar images. The overall block representation of the FaceNet approach of face recognition is mentioned in Fig.1.

Fig. 1. FaceNet Overall Architecture Block Diagram

The FaceNet network architecture consists of a batch input layer and a deep convolutional neural network which is then followed by L2 normalization, that provides the face

978-1-5386-9533-3/19/$31.00 ©2019 IEEE

embeddings[9]. This process is inturn followed by the triplet loss.

A. Triplet Loss

Using Triplet Loss process, the distance between the anchor and the sample is minimized if the sample is positive and signifies the same identity; also, the distance between the anchor and negative sample, which signifies a different identity is maximized. Thus, triplet loss is one of the best ways to learn good 128-dimensional embedding for each individual faces[11]. Here the anchor image refers to the reference image that we took from that dataset in order to calculate the triplet loss. For calculating a triplet loss, we need three images an anchor, a positive and a negative image. A visualization of the triplet loss algorithm is represented in the Fig.2.

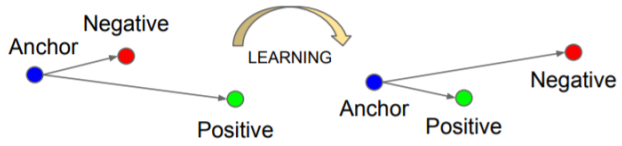


Fig. 2. Triplet Loss Training

Out of all triplet loss methods, online triplet mining method was used to generate triplet loss of non-matching face patches or roughly aligned matching face patches, to train the FaceNet. Representational efficiency can be improved through this approach, by using only 128-bytes per face.

B. MTCNN

The Multi-task Cascaded Convolutional Neural Networks (MTCNN) algorithm used to detect face and face landmarks, works in three steps and uses one neural network for each process. The initial part is a proposal network which will predict potential face positions and their bounding boxes [12] just like an attention network in Faster R-CNN. The result of this process is a large number of face detections and lots of false detections. The second part uses images and outputs of the first prediction, thus making a refinement of the result to eliminate most of the false detections and aggregate bounding boxes. The last part refines even much more the predictions and adds facial landmarks predictions in the original MTCNN implementation[13]. Experimental results had always been demonstrated that while keeping the reliability of real-time performance, this method consistently outperforms the sophisticated conventional methods across most of challenging benchmarks. Face detection and face alignment are analysed with respect to Face Detection Data Set and Benchmark (FDDB) and WIDER FACE benchmarks, and Annotated Facial Landmarks in the Wild (AFLW) benchmark respectively . This better performance for the real time , is of great importance in a surveillance system[13].

C. One Shot Learning

A large amount of dataset is required by conventional deep learning network, to predict the classification with high accuracy. Using minimum images or training samples, one shot learning tries to learn the features of the object classes. In one shot learning [14] fewer dataset of the subjects are used initially to train the model, but for the introduction of newer subjects the same model can be used without being retrained, with the help of addition of new face embeddings. A Siamese network is preferably used to implement a one shot learning.

Siamese networks are aimed at mapping the pair of input vectors into a maximum distance in the feature space if the inputs belongs to two different type (or class), or into a small or minimum distance in the feature space if they belong to the same relation type (class) [15]. From equation (1), an input embedding x, and the convolutional function alongwith max pooling operation fw(·) , gives a flattened vector output. This flattened output vector goes through the dense layer g(·) and calculates the distance layer dist(·) defined by the absolute distance given in the Equation (1).

The distance layer output is the absolute distance value obtained from the difference of the actiavation outputs g(.) for the a pair of input vector x. Thus a mapping is obtained in a feature space using metrical distance. The activation input is generated using the convolution function followed by max pooling fw(.). This output is then flattened and provide to the dense layer g(.) , as given in Equation (1).

$$dist(x_i, x_j) = |g(f_w(x_i)) - g(f_w(x_j))| \quad (1)$$

Optimization of this network is performed with respect to the loss at the output layer, obtained through equation (1). Similar images gives smaller loss where as the loss of non-similar images will be higher.

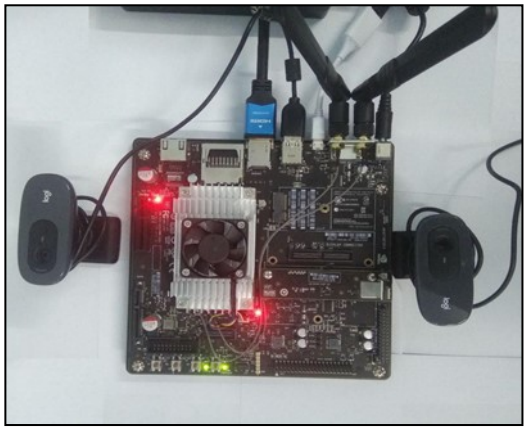


Fig. 3. Jetson TX2 with multiple cameras

IV. JETSON TX2

Jetson TX2 is one of the quickest, embedded AI computing device which is highly power-efficient. This supercomputing module helps in bringing genuine AI processing at the end devices, with a low power consumption of 7.5-watt. It is a GPU based board with Nvidia 256 core pascal architecture along with 64 bit hex core ARMv8 CPU , stacked with a memory of 8 gigabyte and 59.7 GB/s 128-bit interface of memory data transfer capacity. It includes a range of standard equipment interfaces like display, camera, GPIO etc ,thus enabling fast prototyping [16]. It also has a 1400 megapixels per second Image Signal Processor (ISP). Jetpack SDK, is used to automate the basic installations on Jetson TX2, which includes the Board Support Packages, libraries especially for deep learning and computer vision[17].

This is good hardware to run deep learning inferences [18]. This also enhances how fast the person could be

recognized with much higher accuracy. These features that the GPU based board has, helps us implement the multiple camera recognition systems. This helped to focus in implementation on the multi-camera surveillance system.

V. Dataset

The dataset for one shot learning and embedding creation, consists 5000 RGB images. The dataset consists of the face images of 10 students, being a one-shot model very less dataset of images are only required to detect a person, compared to other algorithms. We have also included a dataset of unknown faces which is a set of random images of the foreign members. This data [19] was accessed from the LFW dataset [20]. Fig 4 shows the sample dataset.

Fig. 4. Sample dataset of 10 students

VI. Summary Of Execution

Using Jetson Development Pack (JetPack) installer and ubuntu 16.04 computer, install JetPack into the Jetson TX2 with CUDA support option. Necessary test were ran to verify the installation and verify the version using the command: nvcc -V

Building and installing required packages for FACENET [21]. The packages required for FACENET are as follows TensorFlow==1.7, scipy, scikit-learn, OpenCV-python v3, h5py, matplotlib, Pillow, requests, psutil.To make the programs compatible with the GPU based board Editing the FACENET source code [21] was necessary thus making it compatible for Jetson TX2. The changes include: importing the correct files of sklearn. "class sklearn.cross_validation.KFold" instead of using "class sklearn.model_selection.KFold" this is necessary cause Jetson TX2 have limited support for the latest versions(v.20+) of sklearn. Initialize and verify the multi camera system using the programs build over OpenCV.

Finally access the dataset of Students-frontal face images and train with the dataset ,to create the embedding file , using the the pre trained FaceNet model [9]. FaceNet Model Name: 20180402-114759 was used to create the embedding.

Real-Time Face Detection and Recognition is performed, which include the following sub processes: reading frames with OpenCV, face detection using MTCNN, face Embedding with triplet loss and Siamese Network face Recognition using FaceNet algorithm, storing the recognized face for analysis and to monitor intruder alert. The identified face is exported into an Excel sheet for registering the suspects presence, along with the location ID of the place where the target person was detected. The targets location is bound to the camera ID or its location. The Fig.6. Illustrates the Block diagram of the implementation.

Surveillance system is an application of multiple camera face recognition and subject tracking. Live tracking of the subject candidates and logging of route data (consecutive locations) by analysis of the camera locations, in which the subjects are detected, are the key features of the system. The Excel sheet that displays the demo output of the surveillance is shown in Fig.5. and real time output in Fig.7.

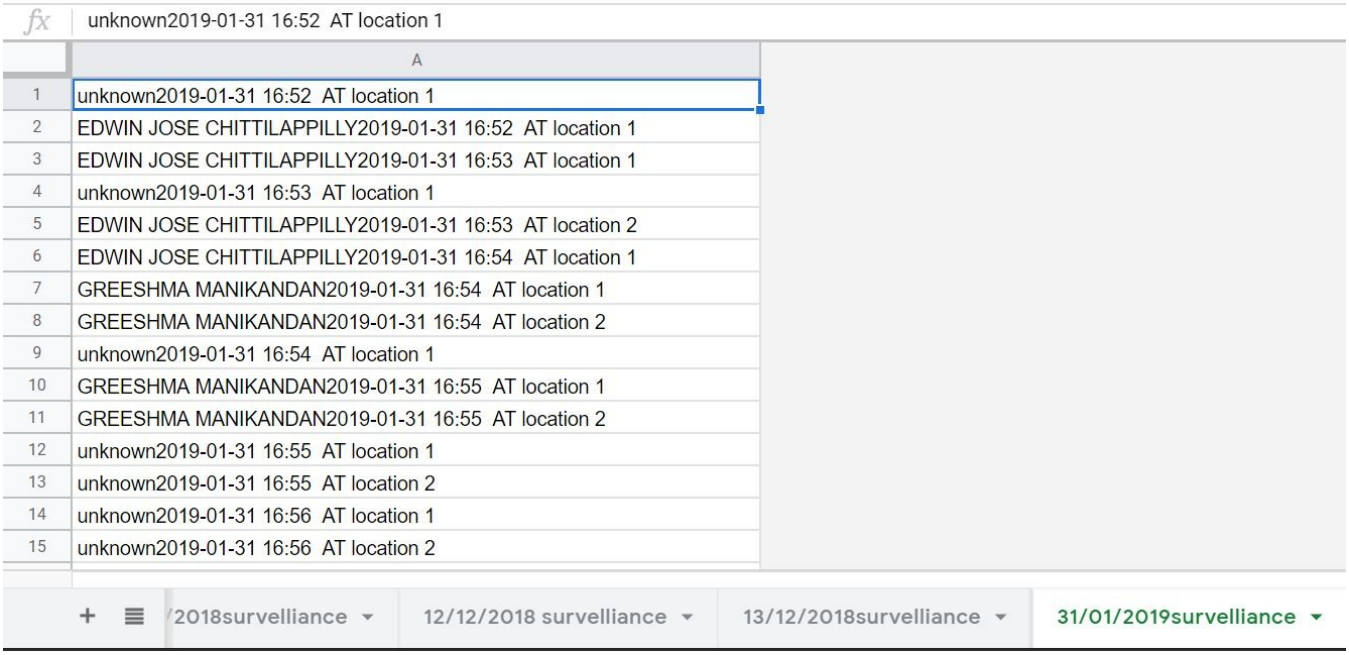


Fig. 5. Snapshot of the Excel sheet With the Sample Output of the Surveillance system Targets: Edwin Jose and Greeshma Manikandan

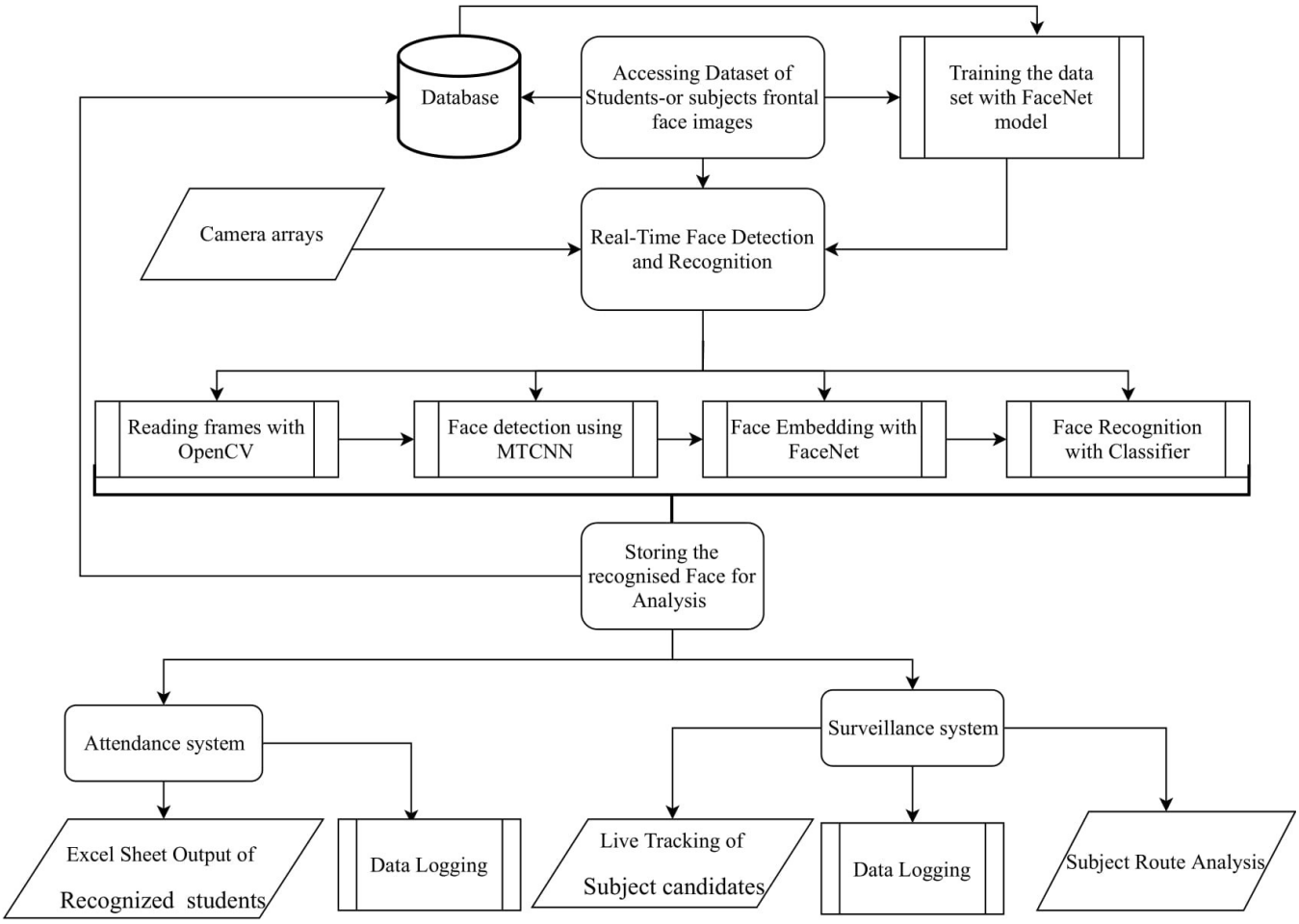


Fig. 6. Block diagram of multi camera Surveillance system

VII. CHALLENGES

JetPack 3.3 is used for the flashing of CUDA and load depended GPU related files and packages to Jetson TX2 [22]. The challenges faced with the OS and other packages, due to dependency issues were resolved and installation was successfully done in ubuntu 16.04 .The dependencies of the FaceNet source code was not supported by certain packages and dependencies in Ubuntu Linux x64 (v16.04) for Jetson TX2.Also, the processor of TX2 is ARM-based hence it took initial time to compile the various packages like OpenCV 3.3 and other packages like sklearn, which are necessary for implementing FaceNet. Sklearn is having the latest version of v0.20.2. Jetson TX2 repositories support only till version v0.17 of sklearn (at the time of experimentation), therefore it was necessary to change and update the source code for the same.

After having compiled the necessary packages and dependencies, new source code of FaceNet was ready to be executed. During the process, it was also learnt that the embedding required for the recognition should be made or trained on Jetson TX2 or on a computer with same versions of the dependencies. Hence the method of generating the embedding on the Jetson TX2 was adopted, because it makes overall system more real-time and focuses on the concept of a portable surveillance system. The threshold of detection is set to 80 percent hence it confirms an identity only if its above 80 percent. This improves the accuracy of the detection and hence provided an overall accuracy of 97 % in tracking the suspects in our various trials.

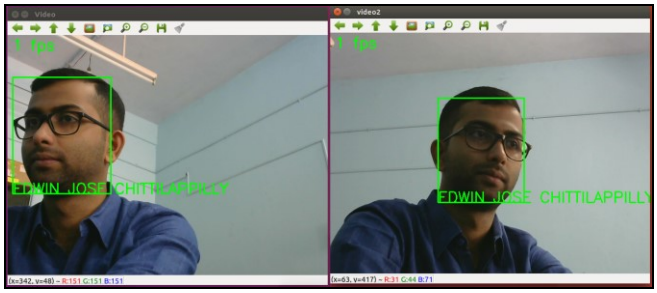


Fig. 7. Monitor output after multiple camera recognition on Jetson TX2

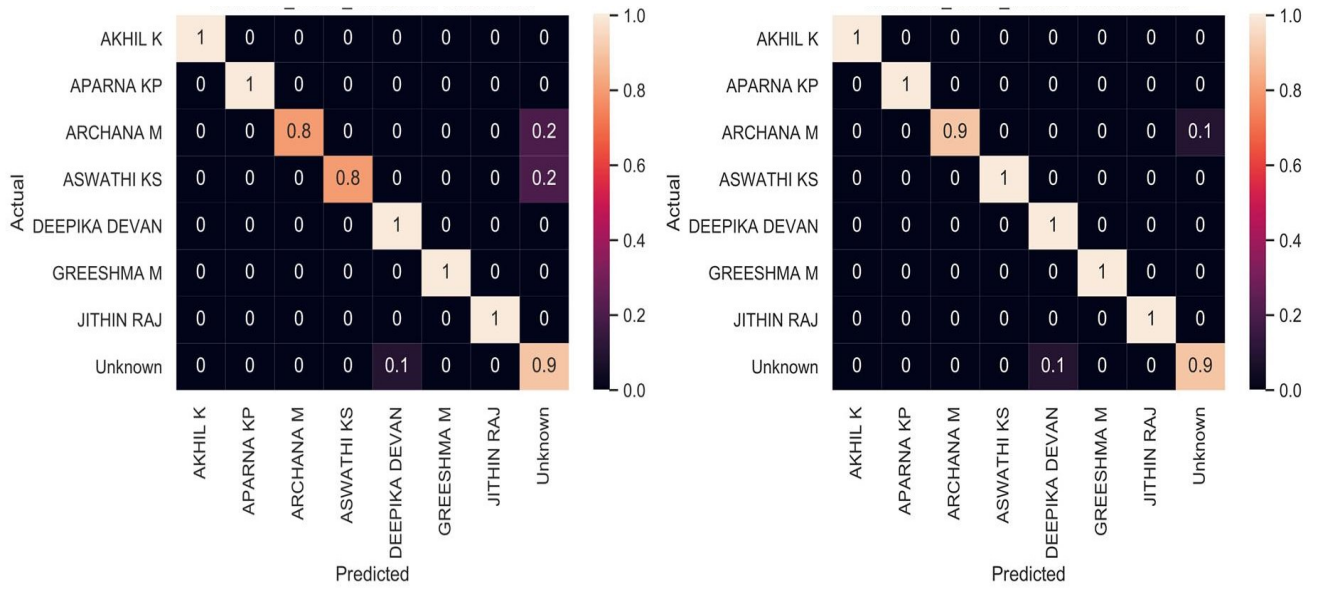
VIII. ANALYSIS

The performance of the model on Jetson TX2 is analyzed using confusion matrix as shown in the Fig 8. The metrics that are used to assess the performance of classification models are Accuracy, precision, recall, specificity, F1 score. Table I provides the model evaluation parameters and can be analyzed that the system is having an accuracy of 97% on the generated tracking of the seven suspects given in the dataset to track. The data given in the table I is from camera ID: CAM 1 and camera ID: CAM 2, since the system was implemented and tested for a multi camera arrangement. The maximum value in the table is 1

and it corresponds to 100% . Analysis on precision values of seven known classes (i.e of suspects) are above 90% in both the cameras. Recall has a value above 80%, this signifies the true positive rate of the system, while the specificity being above 98% implies that the system has a good true negative rate. F1 score of the known classes are also above 88% and have shown a maximum value of 100% this shows that there is good balance between the precision and recall.

TABLE I. CONFUSION MATRIX PARAMETER VALUES FOR EACH CLASS

PERSON	PRECISION		RECALL		SPECIFICITY		F1 SCORE		ACCURACY	
	CAM 1	CAM 2	CAM 1	CAM 2	CAM 1	CAM 2	CAM 1	CAM 2	CAM 1	CAM 2
AKHIL K	1	1	1	1	1	1	1	1	1	1
APARNA KP	1	1	1	1	1	1	1	1	1	1
ARCHANA M	1	1	0.8	0.9	1	1	0.888889	0.947368	0.975	0.9875
ASWATHI KS	1	1	0.8	1	1	1	0.888889	1	0.975	1
DEEPIKA DEVAN	0.909091	0.909091	1	1	0.985714	0.985714	0.952381	0.952381	0.9875	0.9875
GREESHMA M	1	1	1	1	1	1	1	1	1	1
JITHIN RAJ	1	1	1	1	1	1	1	1	1	1
Unknown	0.692308	0.9	0.9	0.9	0.942857	0.985714	0.782609	0.9	0.9375	0.975
Overall Statistics:									**0.9375**	**0.975**



(a) (b)

Fig. 8. Confusion matrix plots: (a) Plot of data from CAM 1 (b) Plot of data from CAM 2

IX. CONCLUSIONS

Face Recognition based Surveillance System Using FaceNet and MTCNN on Jetson TX2 was successfully implemented with multiple cameras (2 in number). This standalone system detects the person which was already given in the dataset to track and an embedding being created was successfully detected with an accuracy of 97%. This multiple camera surveillance helps in live tracking of the person with the location along with the time stamp which is stored as a log in a excel file.

The work can be extended to implement a fully-fledged surveillance and suspect tracking system, using facial recognition and Marking the suspects positions on the maps for more real time analysis. This system could also be implemented complete offline method and hence providing an inherent and enhanced security for our data. The small size and the high-power efficiency if the Jetson TX2 and

highly successful FaceNet algorithm could also find its own applications in the field of social and interactive robotics.

REFERENCES

[1] S. S. Thomas, S. Gupta and V. K. Subramanian, "Smart surveillance based on video summarization," 2017 IEEE Region 10 Symposium (TENSYMP), Cochin, 2017, pp. 1-5. doi: 10.1109/TENCONSpring.2017.8070003

[2] J. Kim, J. Park, K. Lee, K. Baek and S. Kim, "A Portable Surveillance Camera Architecture using One-bit Motion Detection," in IEEE Transactions on Consumer Electronics, vol. 53, no. 4, pp. 1254-1259, Nov. 2007. doi: 10.1109/TCE.2007.4429209

[3] W. Zhang et al., "A Streaming Cloud Platform for Real-Time Video Processing on Embedded Devices," in IEEE Transactions on Cloud Computing. doi: 10.1109/TCC.2019.2894621

[4] A. Ezzahout and R. O. Haj Thami, "Conception and development of a video surveillance system for detecting, tracking and profile analysis of a person," 2013 3rd International Symposium ISKO-Maghreb, Marrakech, 2013, pp. 1-5. doi: 10.1109/ISKO-Maghreb.2013.6728128

[5] L. Zhang, J. Wang and K. Zhang, "Design of Embedded Video Monitoring System Based on S3C2440," 2013 Fourth International Conference on Digital Manufacturing & Automation, Qingdao, 2013, pp. 461-465. doi: 10.1109/ICDMA.2013.108

[6] Y. Yu, X. Duan, S. Wang and B. Jiao, "The design and implementation of bluetooth video surveillance devices," 2010 3rd International Congress on Image and Signal Processing, Yantai, 2010, pp. 495-498. doi: 10.1109/CISP.2010.5647682

[7] Savath and Supavadee, "Real-Time Multiple Face Recognition using Deep Learning on Embedded GPU System," *Proceedings, APSIPA Annual Summit and Conference 2018*, pp. 1318–1324, Nov. 2018.

[8] "Google FaceNet scores almost 100% recognition," Biometric Technology Today, vol. 2015, no. 4, pp. 2–3, 2015.

[9] F. Schroff, D. Kalenichenko and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 815-823. doi: 10.1109/CVPR.2015.7298682

[10] S. D. Shendre, "An Efficient way to Trace Human by Implementing Face Recognition Technique using TensorFlow and FaceNet API," International Journal for Research in Applied Science and Engineering Technology, vol. 6, no. 4, pp. 605–608, 2018.

[11] Z. Ming, J. Chazalon, M. M. Luqman, M. Visani, and J.-C. Burie, "Simple Triplet Loss Based on Intra/Inter-Class Metric Learning for Face Verification," 2017 IEEE International Conference on Computer Vision Workshops (ICCVW), 2017.

[12] M. Ma and J. Wang, "Multi-View Face Detection and Landmark Localization Based on MTCNN," 2018 Chinese Automation Congress (CAC), Xi'an, China, 2018, pp. 4200-4205. doi: 10.1109/CAC.2018.8623535

[13] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li: "Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks", 2016; [http://arxiv.org/abs/1604.02878 arXiv:1604.02878]. DOI: [https://dx.doi.org/10.1109/LSP.2016.2603342 10.1109/LSP.2016.2603342].

[14] Li Fei-Fei, R. Fergus and P. Perona, "One-shot learning of object categories," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 4, pp. 594-611, April 2006.doi: 10.1109/TPAMI.2006.79

[15] J. Yuan, H. Guo, Z. Jin, H. Jin, X. Zhang and J. Luo, "One-shot learning for fine-grained relation extraction via convolutional siamese neural network," 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, 2017, pp. 2194-2199.doi: 10.1109/BigData.2017.8258168

[16] "Jetson TX2 Module", NVIDIA Developer, 2018

[17] T. Amert, N. Otterness, M. Yang, J. H. Anderson and F. D. Smith, "GPU Scheduling on the NVIDIA TX2: Hidden Details Revealed," 2017 IEEE Real-Time Systems Symposium (RTSS), Paris, France, 2018, pp. 104-115. doi:10.1109/RTSS.2017.00017

[18] "Harness AI at the Edge with the Jetson TX2 Developer Kit", NVIDIA Developer, 2018..

[19] Vis-cs.umass.edu. (2019). LFW Face Database.

[20] Labeled Faces in the Wild: A Survey. In Advances in Face Detection and Facial Image Analysis, edited by Michal Kawulok, M. Emre Celebi, and Bogdan Smolka, Springer, pages 189-248, 2016

[21] D. Sandberg, "davidsandberg/FaceNet", GitHub, 2018.

[22] NVIDIA Developer. (2019). JetPack.