Multiple Face Tracking Using Kalman and Hungarian Algorithm to Reduce Face Recognition Computational Cost

Willy Achmat Fauzi
Department of Electrical Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
willy.achmat@gmail.com

Supeno M Susiki Nugroho
Department of Computer Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
mardi@ee.its.ac.id

Eko Mulyanto Yuniarno
Department of Computer Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
ekomulyanto@ee.its.ac.id

Wiwik Anggraeni
Department of Electrical Engineering
Department of Computer Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
University Center of Excellence on
Artificial Intelligence for Haelthcare
and Society
wiwik@is.its.ac.id

Mauridhi Hery Purnomo
Department of Electrical Engineering
Department of Computer Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
University Center of Excellence on
Artificial Intelligence for Haelthcare
and Society
hery@ee.its.ac.id

***Abstract*—Currently, research in face recognition systems mainly utilized deep learning to achieve high accuracy. Using deep learning as the base platform, per frame image processing to detect and recognize faces is computationally expensive, especially for video surveillance systems using large numbers of mounted cameras simultaneously streaming video data to the system. The idea behind this research is that the system does not need to recognize every occurrence of faces in every frame. We used MobileNet SSD to detect the face, Kalman filter to predict face location in the next frame when detection fails, and Hungarian algorithm to maintain the identity of each face. Based on the result, using our algorithm 87.832 face that must be recognized is reduced to only 204 faces, and run at the real-time scenario. This method is proven to be used in surveillance systems by reducing the computational cost.**

Keywords—multiple face tracking, Kalman filter, Hungarian algorithm, video surveillance system

I. INTRODUCTION

The development of computer vision technologies has provided a wider space both for research and application ends. Various techniques and algorithms have been actively developed, for instance in image classification [1] and object detection [2][3]. This rapid development has proven to be beneficial to other fields such as public security, health care, educational institutions, and telecommunication providers.

Large-scale video surveillance technology can be used by government and law enforcer entities to provide better safety and security services to the community. This technology can also be used as a traffic monitoring system [4], vehicle detection, and counting [5], license plate identification, object detection, people re-identification [6], and emergency detection. Together with Intelligent Transportation System (ITS) and other emerging smart technologies could lead us to a concept commonly known as a smart city shortly.

One emerging technology that is becoming more important for smart cities is face recognition. The history of facial recognition systems gaining popularity in 1991 when eigenface [7] was introduced, the motivation behind the eigenface algorithm is that facial images have a statistically significant redundancy value. Principal Component Analysis (PCA) [8] can be used to reduce dimensions and form a more concise representation. In the 1990s holistic methods still dominated facial recognition systems. In the 2000s, the local handcraft-based method became popular followed by the local descriptor learning approach. Recently, DeepFace [9], DeepID [10], and FaceNet [11] achieved surprising results in their performance, so the researcher starts to focus on the deep learning-based approach. Using this approach the performance for the Labeled Face in-the-Wild dataset continues to increase from around 60% to above 90%. The major downside of the use of a deep learning model for face recognition system is computational intensity, requiring high-performance computational resources.

Per frame image processing to detect and recognize faces is computationally expensive, especially for large scale video surveillance system where a large number of cameras are mounted and sending video data continuously in a simultaneous manner. Consider a case as follows: given a CCTV camera that can record a video of HD quality at 30 fps and placed in a relatively crowded site e.g. shopping mall or train station. If a person identified by a system at frame 1 and the person is still on the frame 3 seconds afterward, the system will consider and recognize this person's identification 90 times

Addressing that problem this research real-time multi-face tracking in an uncontrolled environment is proposed to reduce computational cost in video surveillance systems.

II. RELATED WORKS

Many approaches of Multiple Object Tracking (MOT) have a common strategy, the first step is detecting object occurrences in each frame [3][12], the second step is

associating detection result with the previous frame [13][14][15]. Using this scenario the performance of the detector is crucial to obtain good tracking results.

Face detection becomes popular after the pivotal work by Viola-Jones and achieves high detection rates by using a well-trained classifier[16]. Other popular non-neural based methods such as HOG [17] also have a good result. However, many deep neural-based methods outperformed the traditional method [18][12]. Another approach to combine CNN's with object detection was made by Redmon et al. [3] and is called (You Only Look Once) YOLO. This approach also has been implemented in face detection [19].

Another researcher also focuses on increasing detection speed by reducing the model size to make it run in real-time. MobileNet is one of an efficient network architecture that is designed for mobile or embedded computer vision applications [20]. In some research MobileNet based architecture for object detection achieved a speed of 4.5 FPS when running on a raspberry pi [21].

MobileNet has small models, so it is effective across a wide range of applications and uses cases including object detection and classification. The accuracy of MobileNet is surprisingly high and good enough for many applications [22].

The second step is multiple object tracking, involving tracking each detected object in scope continuously while simultaneously maintaining the tagged identity of each object. A common problem plaguing multiple-object tracking is a failure when a detected object goes undetected for one or more frames, from occlusion, false negatives, or other factors. This problem spurs the adoption of tracking-by-detection by many algorithms, producing good results [23][24].

Some use cases used a combination of CNN-based object detector with Kalman and Hungarian filters for online MOT cases [25]. For example, kalman and iterative hungarian algorithm used for solving football player tracking [26].

Many approaches didn't use a tracker for the face recognition algorithm and didn't separate detection and recognition processing, both on server processing [27][28] and edge computing approach [29].

Another researcher proposed deep learning-based, distributed, and scalable surveillance architecture that separates detection and recognition. But still, all detected face is feed into face recognition process without any filter [30].

Based on the information contained in previous research and paying attention to the advantages of kalman and hungarian algorithm, this research proposed a real-time face detection system using MobileNet-SSD face detection combined with kalman and hungarian algorithm to reduce the computational cost of the deep learning-based face recognition system.

III. PROPOSED METHOD

This research aims to remove duplicate face that comes from video surveillance to reduce computational cost. The detection process is carried out using MobileNet SSD for face detection which is combined with the Kalman-Hungarian algorithm to track the same face. The process flow of this proposed combination is described in a workflow as in Figure 1.

Based on our proposed method in Fig. 1, the system starts by taking an input frame and putting it through the face detector, and obtaining the bounding box centers. A Hungarian algorithm correlates the face detected between the neighboring frames, then the system will updating Kalman filters for known faces, predicting the location for old faces within a time threshold when the face is not detected. The system will apply new kalman filters instance and recognition for new faces, and discard the face for old expired faces.

MobileNet SSD was used to detect faces in the current video frame. The detection output is the location of the face's bounding box related to the video frame (x_1, x_2, y_1, y_2). The location of the face is then defined as the center point of the face's bounding box (x, y).

We use the Hungarian algorithm to correlate the face detected between the neighboring frames. This used a tracking list filled with all the relevant detected faces currently in the system's scope as well as their locations.

Two criteria need to be fulfilled before a given previously detected face is put kept in the tracking list: whether it was detected in the current frame, and whether its last time of detection was less than the specified time threshold t If a given face was in the tracking list but fulfills none of the criteria, it is removed from the tracking list. If it was instead still under the time threshold, then the location predicted by the Kalman filter is set as its current location.

A detected face that was not in the tracking list is considered as a new face, and thus the system puts in a new entry in the tracking list containing the face, attached a new kalman filter to it, and sent the face to the recognition and verification process.

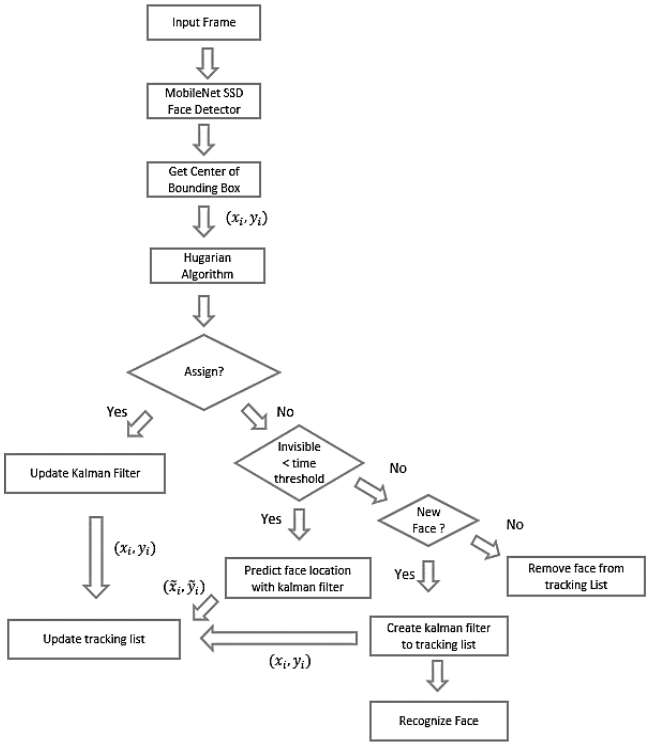

Fig. 1. Workflow of the proposed method

A. Dataset

Because there's no public dataset for face surveillance systems due to privacy concerns, objective comparison between multi-face tracker systems in the surveillance system is difficult due to the lack of generally accepted benchmarks. Due to this, we use the DukeMTMC ReID video dataset already being used in [31][32] for person re-identification systems, using videos of $1920x1080$ dimension at 60 fps and 10 minutes 40 second length.

The experiment described in this paper uses Camera 9 only in a specific ROI from the DukeMTMC dataset due to its restricted viewpoints in that the person used in the data videos is facing the camera and with their faces sufficiently large to be recognized and tracked. We crop-specific ROI at $x_1 = 918$, $x_2 = 1718$, $y_1 = 580$, $y_2 = 1080$, so the videos now has 800x500 dimension.

While the dataset was designed for use in ReID, due to its contents of low-resolution images and videos of real-life scenarios it is still relevant for extensive testing of face detection, recognition, and verification in public surveillance videos.

Because of the limited number of the person in a single frame on DukeMTMC dataset, we can't evaluate performance comparison in a large number of tracked faces. Another dataset is required to measure the run-time performance to full fill that scenario. We use Youtube videos of supporters in football matches to analyze tracking performance in a large number of faces detected in a single frame.

B. Face Detection

Mobilenet-SSD consist of SSD detector [12], and Mobilenet as the Network Model [20]. The SSD will manage the detection face by creating a bounding box. Mobilenet will work to extract the features that will later be classified. Combining SSD and Mobilenet will assist in the process of face detection application.

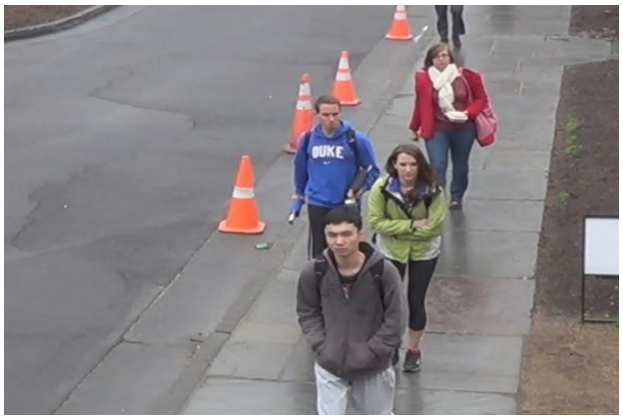

Fig. 2. Example of cropped frame of DukeMTMC ReID video surveillance dataset

The SSD approach illustrated in Fig. 3 used a feed-forward CNN that defined a number of bounding boxes and respective scores for the presence of objects in the boxes, to then merge groups of highly overlapping boxes into a single box using a non-maximum suppression step [33].

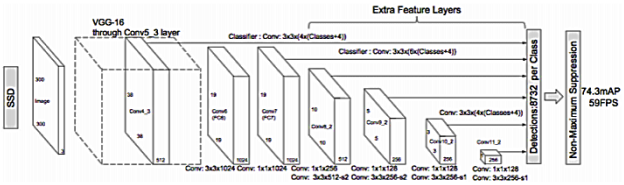

Fig. 3. SSD architecture

While the original research used VGG-16 [12], we replaced VGG-16 with MobileNet as our model. As seen in **Error! Reference source not found.** MobileNet architecture has pretty much the same accuracy as VGG-16, but it has superior performance and a smaller memory size [34][35]. The MobileNet architecture has multiple layers, with the first layer being fully convolutional, and the rest of the layers built on depthwise separable convolutions [20].

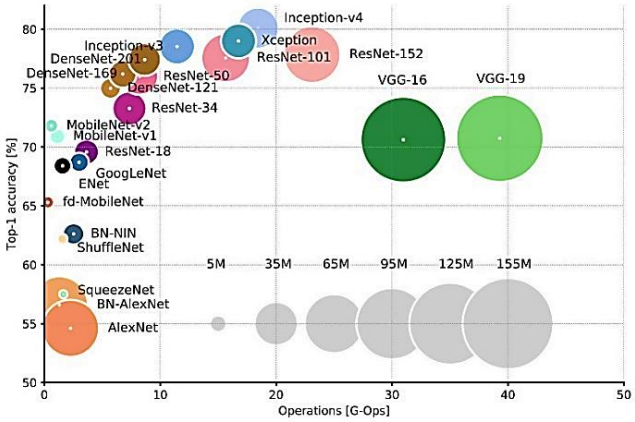

Fig. 4. Top1 vs. operations, size ∝ parameters. Top-1 one-crop accuracy versus the number of operations required for a single forward pass. The size of the blobs is proportional to the number of network parameters [36].

The system must be trained first before it can be used to detect faces. The WIDER FACE dataset [37] was used to train the MobileNet-SSD detector. Based on our research using real-world video inputs described in Table I, the MobileNet-SSD method was one of the fastest deep learning based detectors that still retains a high detection count and true positive rates even in small resolution frames.

C. Video Based Face Tracking and Labeling

The two main parts of multi-face tracking are point tracking and data association. Point tracking used Kalman filter to mathematically model the motions of a particular point and predict the tracking based on the model. The modeling parameters are current position, relative speed, and acceleration, used from the actual measurement values and for predicting the main state. This is applied to every moving object in the frame.

Data association maintains the identity of each detected object, defined in the first frame of the feed video. The identification persists in the following frames. We use Hungarian algorithm to handle this task.

We use a Kalman filter to track and predict the tracked face's position when it doesn't appear in some frames. The main purpose of the Kalman filter is the estimation of those variables which cannot be measured directly, by predicting

and updating. At the prediction stage, an a priori state prediction is modeled by:

$$X_{k\,|\,k-1} = Fx_{k-1|k-1} \quad (1)$$

where $X_{k\,|\,k-1}$ is previous face location. $x_{k-1|k-1}$ is the intermediate predicted location of the face without considering the Kalman gain.

The a priori predicted error covariance is then calculated with:

$$P_{k\,|\,k-1} = FP_{k-1|k-1}F^T + Q \quad (2)$$

where Q is the process noise covariance.

The face's Kalman gain is calculated with:

$$K_k = P_{k|k-1}H^T\left(HP_{k|k-1}HT + R\right)^{-1} \quad (3)$$

If a face is detected in the current frame, an update process is needed. Given the face's new measured location zk the measurement residual rk is:

$$rk = z_k - Hx_{k|k-1} \quad (4)$$

A posteriori state estimate is then updated as:

$$x_{k|k} = x_{k|k-1} + K_k rk \quad (5)$$

while the posterior error covariance is calculated with :

$$Pk_k = (I - K_k H)P_{k|k-1} \quad (6)$$

where *I* is an Identity matrix.

Live face tracking inevitably hit a problem where a detector fails to detect a face in frames for one or more frames due to occlusion or false negative, but then appeared again in future frames. On this intermittent detection face, our system will assume that if the reappearing face was in the expected position of a given face within the time threshold of its last detection then it will be considered to be the same face.

Another problem in live face tracking is matching faces between two given frames when there are many detected faces in each frame. We use the classic Hungarian method to solve this problem. We used the Euclidean distance between two same detected faces in two successive frames for this by registering them in the tracking list. The filter's cost matrix may then be constructed as:

$$C_{i,j} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \quad (7)$$

The aim of the Hungarian algorithm is to find the minimum total cost, which in our case is constructed as:

$$min \sum_{i=1}^{m}\sum_{j=1}^{n} C_{i,j}x_{i,j}$$
$$s.t. \sum_{i=1}^{m} x_{i,j} = 1, \sum_{j=1}^{n} x_{i,j} = 1 \quad and \quad x_{i,j} \in \{0,1\} \quad (8)$$

where m and n are the numbers of the tracked and new faces.

IV. Results and Analysis

A. Face Detection

We compare five face detectors, Haar Casscade OpenCV 3.4 [16], LBP Classifier OpenCV 3.4 [38], Dlib 19.2 HOG [17], MTCNN [18] and Mobilenet SSD [12] on low resolution face from DukeMTMC ReID video dataset to measure speed and accuracy.

DukeMTMC is ReID dataset, so it didn't contain much information about face detection metrics. Besides that creating a precise bounding box is challenging when detected faces that have low resolution. Thus, for choosing a face detector we will give more importance to recall than Intersection over Union (IoU). We run on Nvidia GeForce GTX 1060 GPU with Intel Core i7-7700HQ and 16GB of RAM. Total face detected and true positive rate from detection summarized in Table I.

TABLE I. Face Detection Count for Each Algorithm in Surveillance Footage

Method	Detected	TP (%)	Average FPS
LBP Classifier	735	98.77	68.4
HOG	8.553	90.63	14.62
Haar Cascade	58.481	87.22	87.22
MTCNN	98.443	99.73	14.97
Mobilenet SSD	87.832	99.63	42.58

As shown in Table I, non-CNN algorithms used in previous researches such as HOG, Haar cascade, and LBP classifier have weaker detection performance relative to other methods, with the lower total detected frames compared to CNN-based methods such as Mobilenet SSD and MTCNN in tests using real-world surveillance footage from our dataset. This contrast stems from the LBP's algorithm calibrated using predetermined camera angles in its video data that restricts variability in the dataset. LBP's 735 detected frames are all above 48 px in dimensions, while the footage data's frequency distributions averaged 50 px in dimensions as shown in **Error! Reference source not found.**. Due to this, LBP's viability suffers in real-world footages.

We obtained a comprehensive view of the performance and true positive of various face detection methods on the chosen dataset. TP in Table I is the level of the model that correctly predicts the positive class. As seen in Table I, the MobileNet SSD method is one of the fastest detectors that retains high detection count and true positive rates even in small resolution frames as shown in **Error! Reference source not found.**. Therefore, we chose the MobileNet SSD method [12] for further tests.

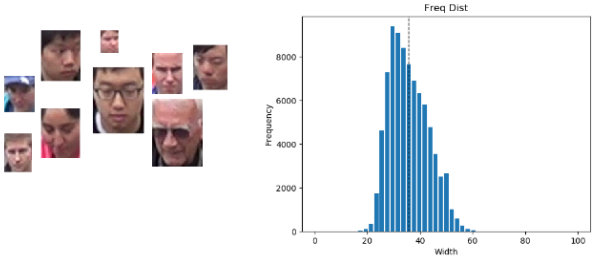

Fig. 5. Mobilenet sdd output face detector image size distribution

B. *Multiple Face Tracking*

Due to the lack of generally accepted multi-face tracking benchmarks, objective comparisons between different tracking systems are difficult, forcing us to resort to an evaluation with the DukeMTMC ReID video surveillance dataset. The result of the multiple tracking mark is shown in Figure 5.

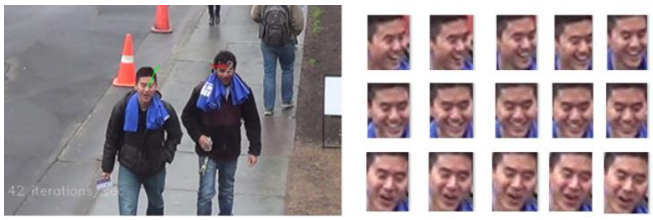

Fig. 6. Example of DukeMTMC frames with multiple tracking mark

We compare the frame rate result with and without Kalman Filter and Hungarian algorithm. Based on our tests, as seen in **Error! Reference source not found.** processing time using Kalman and Hungarian have a small difference in performance. Face detection speed with MobileNet-SSD has an average of 41.67 fps, with an average iteration drop of only 0.8 frames per second with Kalman filter and Hungarian algorithm active.

Fig. 7. Performance comparison of face tracking using mobilenet ssd with and without Kalman and Hungarian algorithm

In real-time cases, the stability of the runtime in the face of changing number of tracked faces is also a concerning factor. From our tests, we managed a somewhat stable runtime cost when faced against various numbers of faces in the test. The relation between runtime and tracked face numbers in the test was illustrated by **Error! Reference source not found.**. The figure showed that the runtime was not significantly impacted until 8 tracked faces.

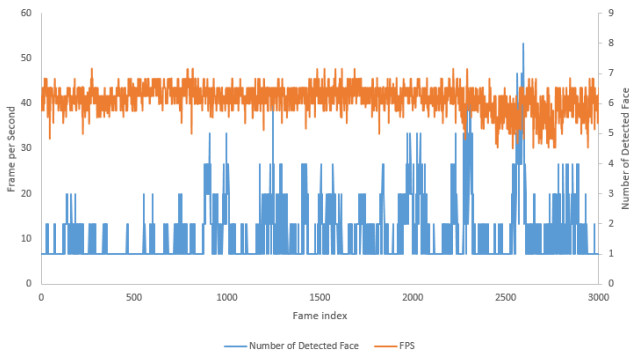

Fig. 8. Tracker runtime cost by an increase in the number of faces in a single frame. The X-axis as frame index. The left Y-axis for the orange color shows fps, The right Y-axis for the blue color is the number of the detected face.

Based on a current dataset that we use, the maximum face detected in each frame is 8 faces. To calculate the cost with a high number of the face we use video from football match from youtube to achieve this scenario. Based on our result, the runtime cost only starts to increase if the number of tracked faces also rises significantly.

From our tests, when the number of tracked faces rise from 2 to 34, the system's average fps dropped from 41.60 to 38.35 fps. Thus, in practice, we conclude that the runtime cost of our tracker will only increase sublinearly to the number of tracked faces.

We also test results combining with a well-known face recognition algorithm named Facenet [11] to verify face with and without our algorithm. In this scenario, we only have 8 people in the database to match faces, and the average face count on a single frame is 2 faces.

With a total face in each frame increase, facenet without our algorithm has suffered from a drop frame rate as linear as total face verified. Our proposed methods will drop only when the first face is detected as a new face, as displayed in **Error! Reference source not found.**.

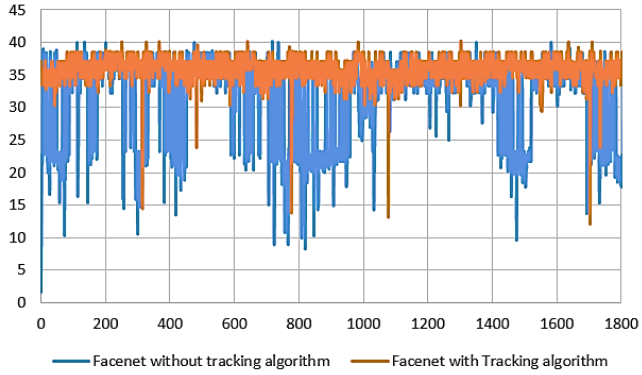

Fig. 9. Facenet algorithm [11] with and without face tracking algorithm

Another great improvement came from the total face that has to verify in the face recognition system. With normal process 87.832 face is feed into the face verification, this number greatly reduced to only 204 faces with our proposed method.

With this algorithm, we get the functionality of multi-face tracking without losing too much computing power and reduce face recognition computational cost, hence suitable for real-time video face recognition in the surveillance system.

V. CONCLUSIONS

The real-time multiple face detection and tracking system algorithm is proposed to maintain the identity of a person. To achieve the high detection rate, the algorithm in combination CNN based face detection using MobileNet SSD with Kalman and Hungarian algorithm is employed.

Our proposed multi-face tracking runs in real-time, separating concern between the detection and verification process and eliminating similar frames to avoid processing redundant data. This greatly speeds up runtime compared to traditional face recognition systems that processed all the faces in a given frame each time a new frame arrives. In the future, the method may be developed to improve execution

time and increase detection efficiency in real-time surveillance applications.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," Commun. ACM, 2017, doi: 10.1145/3065386.

[2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Trans. Pattern Anal. Mach. Intell., 2017, doi: 10.1109/TPAMI.2016.2577031.

[3] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017, doi: 10.1109/CVPR.2017.690.

[4] R. Elhakim, M. Abdelwahab, A. Eldesokey, and M. Elhelw, "Traffisense: A smart integrated visual sensing system for traffic monitoring," in IntelliSys 2015 - Proceedings of 2015 SAI Intelligent Systems Conference, 2015, doi: 10.1109/IntelliSys.2015.7361174.

[5] H. Tayara, K. G. Soo, and K. T. Chong, "Vehicle Detection and Counting in High-Resolution Aerial Images Using Convolutional Regression Neural Network," IEEE Access, 2017, doi: 10.1109/ACCESS.2017.2782260.

[6] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, "Performance Measures and a Data Set forMulti-Target, Multi-Camera Tracking," in European Conference on Computer Vision Workshops (ECCVW), 2016.

[7] M. Turk and A. Pentland, "Eigenfaces for recognition," J. Cogn. Neurosci., 1991, doi: 10.1162/jocn.1991.3.1.71.

[8] I. T. Jolliffe, "Principal component analysis," Springer, New York., 1986.

[9] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2014, doi: 10.1109/CVPR.2014.220.

[10] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in Advances in Neural Information Processing Systems, 2014.

[11] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2015, doi: 10.1109/CVPR.2015.7298682.

[12] W. Liu et al., "SSD: Single shot multibox detector," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2016, doi: 10.1007/978-3-319-46448-0_2.

[13] C. Jia et al., "A Tracking-Learning-Detection (TLD) method with local binary pattern improved," in 2015 IEEE International Conference on Robotics and Biomimetics, IEEE-ROBIO 2015, 2015, doi: 10.1109/ROBIO.2015.7419004.

[14] Y. Tian, A. Dehghan, and M. Shah, "On Detection, Data Association and Segmentation for Multi-target Tracking," IEEE Trans. Pattern Anal. Mach. Intell., 2018, doi: 10.1109/TPAMI.2018.2849374.

[15] H. Sheng, Y. Zhang, J. Chen, Z. Xiong, and J. Zhang, "Heterogeneous Association Graph Fusion for Target Association in Multiple Object Tracking," IEEE Trans. Circuits Syst. Video Technol., 2019, doi: 10.1109/TCSVT.2018.2882192.

[16] P. Viola and M. Jones, "Robust real-time face detection," Int. J. Comput. Vis., 2004, doi: 10.1023/B:VISI.0000013087.49260.fb.

[17] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, 2005, doi: 10.1109/CVPR.2005.177.

[18] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," IEEE Signal Process. Lett., 2016, doi: 10.1109/LSP.2016.2603342.

[19] Y. Wang and J. Zheng, "Real-time face detection based on YOLO," in 1st IEEE International Conference on Knowledge Innovation and Invention, ICKII 2018, 2018, doi: 10.1109/ICKII.2018.8569109.

[20] A. G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv. 2017.

[21] N. S. Sanjay and A. Ahmadinia, "MobileNet-Tiny: A deep neural network-based real-time object detection for rasberry Pi," in Proceedings - 18th IEEE International Conference on Machine Learning and Applications, ICMLA 2019, 2019, doi: 10.1109/ICMLA.2019.00118.

[22] J. Huang et al., "Speed/accuracy trade-offs for modern convolutional object detectors," in Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017, doi: 10.1109/CVPR.2017.351.

[23] A. Lukežič, T. Vojíř, L. Čehovin Zajc, J. Matas, and M. Kristan, "Discriminative Correlation Filter Tracker with Channel and Spatial Reliability," Int. J. Comput. Vis., 2018, doi: 10.1007/s11263-017-1061-3.

[24] Z. Yang, J. Wu, and C. Long, "Learning Spatial-Corrected Regularized Correlation Filters for Visual Tracking," in Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI, 2019, doi: 10.1109/ICTAI.2019.00-96.

[25] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in Proceedings - International Conference on Image Processing, ICIP, 2016, doi: 10.1109/ICIP.2016.7533003.

[26] B. Sahbani and W. Adiprawita, "Kalman filter and iterative-hungarian algorithm implementation for low complexity point tracking as part of fast multiple object tracking system," in Proceedings of the 2016 6th International Conference on System Engineering and Technology, ICSET 2016, 2017, doi: 10.1109/FIT.2016.7857548.

[27] F. Cahyono, W. Wirawan, and R. Fuad Rachmadi, "Face recognition system using facenet algorithm for employee presence," in 4th International Conference on Vocational Education and Training, ICOVET 2020, 2020, doi: 10.1109/ICOVET50258.2020.9229888.

[28] T. Nyein and A. N. Oo, "University Classroom Attendance System Using FaceNet and Support Vector Machine," in 2019 International Conference on Advanced Information Technologies, ICAIT 2019, 2019, doi: 10.1109/AITC.2019.8921316.

[29] E. Jose, M. Greeshma, T. P. Mithun Haridas, and M. H. Supriya, "Face Recognition based Surveillance System Using FaceNet and MTCNN on Jetson TX2," in 2019 5th International Conference on Advanced Computing and Communication Systems, ICACCS 2019, 2019, doi: 10.1109/ICACCS.2019.8728466.

[30] H. C. Kaşkavalci and S. Gören, "A Deep Learning Based Distributed Smart Surveillance Architecture using Edge and Cloud Computing," in Proceedings - 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications, Deep-ML 2019, 2019, doi: 10.1109/Deep-ML.2019.00009.

[31] F. Solera, S. Calderara, E. Ristani, C. Tomasi, and R. Cucchiara, "Tracking Social Groups Within and Across Cameras," IEEE Trans. Circuits Syst. Video Technol., 2016.

[32] E. Ristani and C. Tomasi, "Tracking Multiple People Online and in Real Time," in Asian Conference on Computer Vision, 2014, pp. 444–459.

[33] J. Hosang, R. Benenson, and B. Schiele, "Learning non-maximum suppression," in Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017, doi: 10.1109/CVPR.2017.685.

[34] T. Agarwal and H. Mittal, "Performance Comparison of Deep Neural Networks on Image Datasets," in 2019 12th International Conference on Contemporary Computing, IC3 2019, 2019, doi: 10.1109/IC3.2019.8844924.

[35] S. Bianco, R. Cadene, L. Celona, and P. Napoletano, "Benchmark analysis of representative deep neural network architectures," IEEE Access, 2018, doi: 10.1109/ACCESS.2018.2877890.

[36] A. Canziani, A. Paszke, and E. Culurciello, "An Analysis of Deep Neural Network Models for Practical Applications," 2016.

[37] S. Yang, P. Luo, C. C. Loy, and X. Tang, "WIDER FACE: A face detection benchmark," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016, doi: 10.1109/CVPR.2016.596.

[38] T. Ojala, M. Pietikäinen, and T. Mäenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary

patterns," IEEE Trans. Pattern Anal. Mach. Intell., 2002, doi: 10.1109/TPAMI.2002.1017623.