Simultaneous License Plate Recognition and Face Detection at the Edge

Fikret Alim[a], Enes Kavakli[a], Sefa Burak Okcu[a], Ertugrul Dogan[a], and Cevahir Cigla[a]

[a]Aselsan Inc., Yenimahalle, Ankara, Türkiye

ABSTRACT

Face recognition (FR) and license plate recognition (LPR) are very crucial algorithms for identification of humans and vehicles in several applications such as surveillance, traffic and access-control. The advances in small single-board computers with high parallel processing power capabilities and the use of low-power Neural Processing Units (NPU) inside embedded System on Chips (SoC), enable real-time face detection (FD) and LPR at the edge. On the other hand, it is still a challenge to run multiple algorithms concurrently with high accuracy and prompt execution (high frame rates) that requires a very efficient software/video analytics algorithm development. Both FR and LPR algorithms need two-stage processing that involve detection and recognition. In this study, we propose a method that enables simultaneous face detection associated with landmark and quality information and LPR at the edge. The FD pipeline detects and tracks the faces, extracts landmarks and quality of faces, to select appropriate faces for recognition and then sends them to face recognition server. LPR algorithm consecutively performs detection and recognition on the embedded platform. Extended YOLO model is utilized for face selection while pruned YOLO and LPRNet models are exploited for license plate detection and license plate reading, respectively. In order to enable real-time performance with high accuracy; optimized AI-models and software architecture are used. As a result of this study, we obtain a high-performance, high-precision and real-time combined face/LPR recognition system which can be very useful for surveillance and security applications.

Keywords: Face detection, face landmark detection, license plate recognition, YOLO, prune, edge processing, embedded, real-time, surveillance, cameras

1. INTRODUCTION

There are several techniques for human identification process such as biometric, iris and fingerprint detection. Although the accuracy of facial recognition systems as a biometric technology is lower than iris recognition and fingerprint recognition, it is widely adopted due to its contactless process.[1] Therefore, facial recognition is the most convenient way to recognize people's identities.

The biggest advantage of facial recognition is to enable mass identification; it can be used at crowded places such as airports, multiplexes and other public places without passers-by even being aware of the system.[2] However, as compared to other biometric techniques, face recognition may not be most reliable and efficient. Quality measures are very important in facial recognition systems as large degrees of variations are possible in face images. Factors such as illumination, expression, pose and noise during face capture can affect the performance of facial recognition systems.[2]

Vision-based vehicle identification systems can extract several information such as type, brand, model, color data and license plates of the vehicles. However, the unique information that belongs to a specific vehicle is the license plate which can be thought as the identity of a vehicle. So, the only way to identify a vehicle with a camera system based-on image processing is to read its license plate number. License plate recognition (LPR)

Further author information:
Fikret Alim: E-mail: fikretalim@aselsan.com.tr,
Enes Kavakli.: E-mail: eneskavakli@aselsan.com.tr,
Sefa Burak Okcu: E-mail: burakokcu@aselsan.com.tr,
Ertuğrul Doğan: E-mail: ertugrul@aselsan.com.tr,
Cevahir Cigla: E-mail: ccigla@aselsan.com.tr

or automatic number-plate recognition (ANPR) is a technology which uses optical character recognition (OCR) on images in order to read vehicle license plate numbers. It can use closed-circuit television (CCTV) cameras, enforcement cameras, or cameras specifically designed for the task. LPR is a widely used technology for vehicle management operations such as perimeter security, car parking management, tolling, intelligent transportation services (ITS), stolen vehicles detection, smart billing and many other applications worldwide.

In this study, we introduce a combined FD and LPR application that can run together inside a low-cost surveillance camera with real-time performance and high accuracy. Combination of these two algorithms can be very useful for surveillance and security applications such as car theft, traffic safety, criminal search and access control. We demonstrate that it is even possible to identify driver and license plate of the vehicle by using a single camera with proper installation.

For face detection algorithm, we use You Only Look Once (YOLO) v3[3,4] model-based R-CNN, which provides one of the best trade-offs between speed and accuracy in the literature. For LPR algorithm, we use two-stage processing. License plate detection is implemented by using improved YOLOv5 model[5,6] which is a newer version of YOLOv3. And for reading license plate numbers we use LPRNet[7] model. LPRNet is an end-to-end method for Automatic License Plate Recognition without preliminary character segmentation. LPRNet consists of the lightweight Convolutional Neural Network, so it can be trained in end-to-end and it is the first real-time License Plate Recognition system that does not use recurrent neural networks (RNN).

In this paper, we focus on the optimizations and novelties in terms of both software and algorithm development for running these two applications simultaneously, at an embedded platform by taking full advantage of the so-called platform's features. Optimized and high-performance ML models are trained and used for this application. As a result of these optimizations, these two algorithms can run together at 13 fps, which provides enough speed for real-time recognition of humans and vehicles in the wild.

The structure of the paper is the following: In Section II, we discuss the related works on this application. In Section III, the motivation of this study is provided. In Section IV, we explain the proposed implementation including FD, LPR subsystems and SoC implementation. In Section V, we provide some useful applications of using FD and LPR algorithms together with examples. In Section VI, the experimental results including accuracy and timing measurements are provided. We explain the evaluation metrics and methods we used to measure the performance of algorithms. Finally, in Section VII, conclusion of this study is given.

2. RELATED WORK

In recent literature, various approaches have been proposed for both FD and LPR algorithms, due to importance and interest of these topics. Classical techniques were used for developing these algorithms in the past, but with the revolution of machine learning in the last decade, mostly machine learning based algorithms are being used today. A. Kumar et al. gives a detailed overview of FD techniques in.[8] The FD techniques are classified as feature-based which contains active shape model, low level analysis and feature analysis methods and image-based which contains neural networks, linear subspace and statistical approaches. The details, advantages and disadvantages of each technique is given in this paper. W. Yang and Z. Jiachun presents real-time FD based on YOLO in.[9] In this study, YOLO target detection system is applied to face detection. It is mentioned that experimental results show that the face detection method based on YOLO has stronger robustness and faster detection speed. Still in a complex environment can guarantee the high detection accuracy and the detection speed can meet real-time detection requirements.

S.B.Okcu et al[10] describes an efficient face quality assessment (FQA) technique by utilizing face quality score calculation with an addition to face landmark detection network. As described in paper and samples shown in Figure 1, blurring, occlusion and change in head orientation are different types of distortions in surveillance scenarios affecting quality of captured faces as well as face recognition performance. In this paper, face quality assessment is associated with face recognition scores for surveillance scenarios which is also necessary for our application. The FD network proposed in our study is the extended one-shot version of the techniques described in of S.B.Okcu et al.'s paper. In one shot, face region, landmarks and face quality metrics are provided as shown in Figure 1.

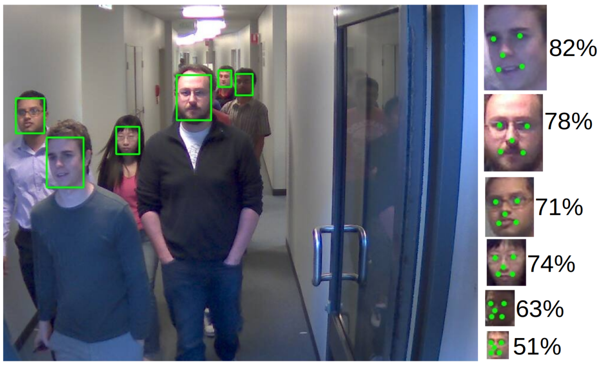

Figure 1. One shot face-landmark detection and face quality assessment can be performed at the edge. Image is taken from[11]

[12] delivers a systematic review of ANPR in terms of technical advancements, influential factors over sensing performance and application cases of this technology. [13] describes the breakdown of some techniques used for the implementation of LPR systems, with significant emphasis on Convolutional Neural Networks (CNN). The strength and weaknesses of the highlighted techniques were discussed. Also, areas of further improvement of some of the selected CNN-based techniques were suggested. There are several studies that use YOLOv5[5,6] for license plate detection[14–20] and LPRNet[7] or other OCR techniques[21] for license plate recognition. For example, a recent study published in 2022 proposes a solution for LPR and vehicle re-identification for surveillance systems.[22] In this paper, a method uses YOLOv5 for detection and OCR-net[23] for recognition is described and the use of LPRNet for OCR function is also mentioned. LPRNet[7] proposes the LP character recognition module with the end-to-end method for automatic LP recognition (ALPR) without preliminary character segmentation. Moreover, this method is lightweight enough to run on a variety of platforms, including embedded devices. In,[24] S. Luo and J. Liu proposes an improved LPR method by using YOLOv5m and LPRnet and it is mentioned that the improved LPR method in this paper performs well in robustness and speed.

In spite of the fact that there is a great number of studies related to FD and LPR topics separately, there are only a few studies related to the combination of these two algorithms. Y.N.Chen et al. has two papers related to the combination of face and license plate detection.[25,26] In ,[25] two detectors are proposed, one for face and the other for license plates, both based on a modified CNN verifier. Pyramid-based localization techniques were applied to fuse the candidates and to identify the regions of faces or license plates. In,[26] a facial/license plate detection method using a two-level cascade classifier and a single convolutional feature map is defined; a coarse-to-fine cascade classifier was designed to detect facial or LP objects. These two papers are focus on the detectors and classifiers for face and license plate objects and do not deal with the OCR part. [27] describes a super-resolution approach to improve quality of facial and license plate images. [28] proposes a method for combination of face and license plate recognition, aims to identify driver and license plate of the vehicles in controlled and well-illuminated environments. In ,[29] the benefits of combining face recognition system (FRS) and automatic license plate recognition (ALPR) with CCTV technology are revealed. It explains in detail that emerging technology in the field of FRS, CCTV and ALPRs could make it possible to create a system capable of identifying suspected terrorists, current terrorist watch list suspects, other wanted criminals, and missing persons.

3. MOTIVATION

As given in Section II, the studies related to face/license plate detection/recognition systems are mostly about developing algorithms that do not have real-time performance concerns at edge devices. There are also studies for real-time implementation of either FD or LPR. However, to the best of our knowledge, the combination of these two algorithms has not been studied so far at an edge device with real-time and high performance requirements. Both FD and LPR algorithms inherently require high processing power, so it is not an easy task to run these algorithms simultaneously at an edge device with real-time operation requirement. At this point, it is necessary to emphasize that the so-called edge device contains a low-power and low-cost processor, otherwise it is always

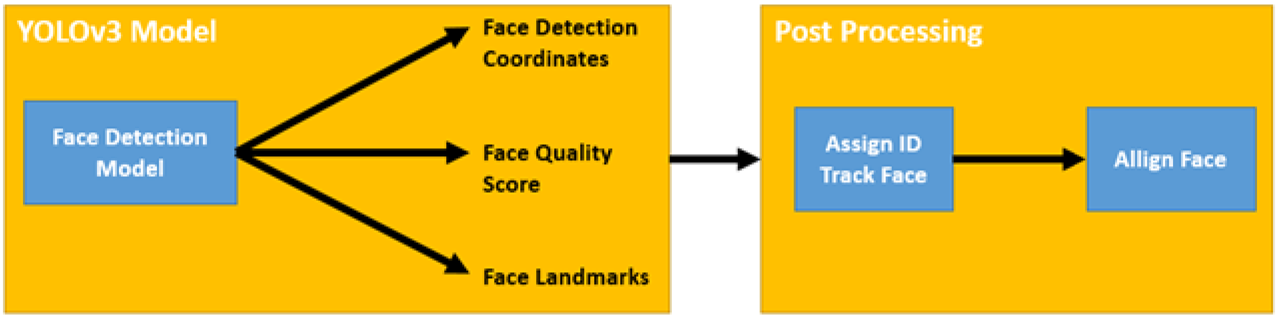

Figure 2. Proposed face detection pipeline

possible to find processors such as an FPGA or GPU with high power, high cost and high processing capabilities which can be used to implement any kind of algorithm even without optimization.

In this study, we aimed to run FD and LPR algorithms at a low-cost edge device with real-time, high accuracy and high performance expectations. For this purpose, optimized machine learning models for both algorithms are used and an efficient software architecture is designed by developing optimized software modules and using hardware resources as much as possible. In addition to using optimized ML models, several post-processing steps are applied to prevent false alarms and increase accuracy. Performance, accuracy and timing measurements are taken in order to prove that the use of proposed system is convenient for edge devices. Through several experiments on use-cases, we illustrate the importance and advantages of using the combination of these algorithms for various applications.

4. THE PROPOSED IMPLEMENTATION

We divide the proposed solution into three categories: FD subsystem, LPR subsystem and system-on-chip (SoC) implementation. In section 4.1 and 4.2, we reveal the algorithmic studies; used ML models and post-processing techniques and in section 4.3, we explain the software implementations for developing this hybrid application on the SoC.

4.1 Face Detection Subsystem

FD networks are mostly developed to extract only face coordinates from a given image. This generates too many irrelevant and redundant data for the final application. The final application is mostly a FR system whose purpose is to identify faces sent from the FD subsystem. Therefore, it is better for the FR system to acquire only relevant faces that are appropriate for recognition task.

With this motivation, we develop a YOLOv3-based detection model for detecting faces, face quality scores and face landmarks. Figure 2, shows the proposed pipeline of FD; the machine learning (ML) model and post-processing stages.

For developing the machine learning model, we inspire from the article by S. B. Okcu et al.[10] In this paper, an efficient face quality assessment (FQA) technique by utilizing face quality score calculation with an addition to face landmark detection network is described. With the same logic, we developed a ML model that does not only extract face coordinates, but also creates face quality scores and face landmarks with extra layers added to model. The network generates five facial landmark points: two for eyes, two for mouth and one for nose regions as shown in Figure 1. Face quality score is a normalized data between 0-1 range for detecting face quality problems including blur, rotation and occlusion.

After the model generates outputs, we apply the following post-processing steps to select the best possible face images for the identification task. The faces are tracked by bi-directional bounding box association, so each individual is assigned to a unique ID. The faces are aligned for recognition and the face crops are sorted for each track ID with respect to estimated face qualities. The aligned faces are sent to the FR server after we make sure that the face is appropriate for identification.

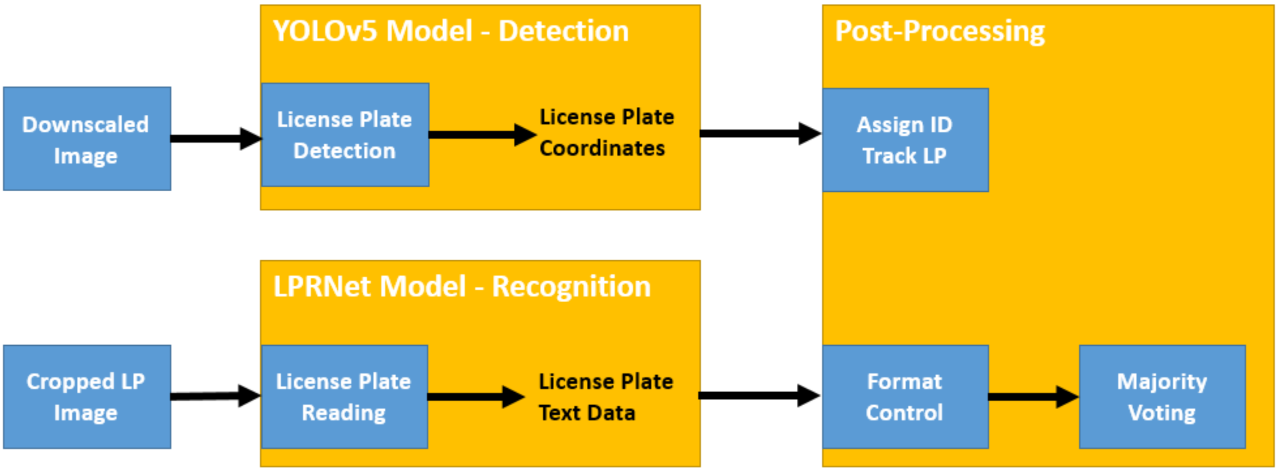

Figure 3. Proposed license plate recognition pipeline

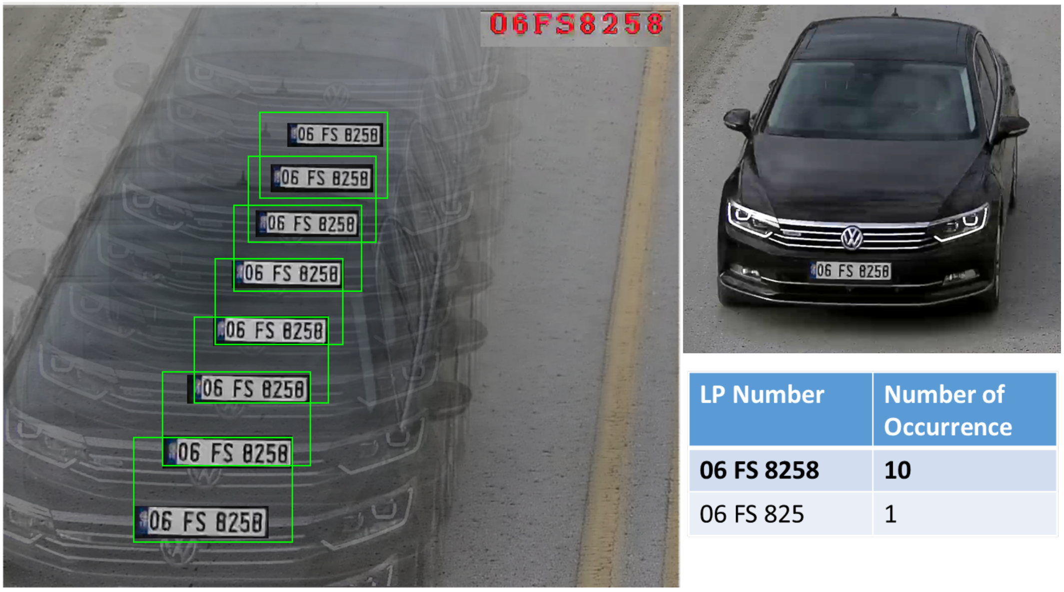

Figure 4. LPR Post-processing Operations Demo

4.2 License Plate Recognition Subsystem

LPR application is implemented by using two machine learning models: YOLOv5 network for detecting LPs in a given image and LPRNet network for reading the LPs detected by the first model. Figure 3, shows the proposed pipeline for LPR. YOLOv5 model detects the coordinates of the LPs in a low resolution image. By using the coordinates generated by the model, the LPs are cropped from the full resolution image and fed into the LPRNet network which processes the LP images and generates text data.

In order to increase the accuracy of LPR, we apply some post-processing steps: tracking, format control and majority voting. The first post-processing step is tracking which traces LPs by using the outputs of the detection network. Detected LPs are tracked by bi-directional bounding box association and an ID is assigned to each LP. When the vehicle enters into the camera field of view, the detection and recognition pipelines run several times. Due to occlusion, instant changes in brightness or small LP size, the recognition model misreads the LPs at some frames. If the algorithm only trusts LP reading, misreading would be evaluated as a new vehicle pass.

Figure 5. Sample Turkish license plates captured by the camera

Therefore, we use the tracker to trace the LPs and generate only one output for any vehicle LP.

Next step of the post processing operation is the majority voting. Tracking LPs solves the problem of generating more than one output for a specific LP. But, the LP can still be misread at some frames because of the reasons described above. In order to solve this problem and decide which reading is correct, we use a majority voting algorithm and return only the LP candidate that gets the most reading for a specific ID. These two post-processing steps ensure that only one and most reliable result for a given LP is returned by the algorithm. Figure 4 shows a sample application of tracking and majority voting.

The last post-processing step is the format control. We use format control mechanism in order to discard false readings. Turkish license plate format, except official vehicles such as military, cabinet members or province governors, is in the form of "city code (01 to 81) - letters - numerals" and a LP can contain total of either 7 or 8 characters. Figure 5 shows different types of Turkish LPs. Format control function, basically discards the license plates that do not obey this rule.

4.3 SoC Implementation

The SoC implementation is mainly designed as a smart camera application which captures raw video from a sensor block, apply video analytics, encode media and create an rtsp[30] stream. The final product is a surveillance camera including motorized zoom lens. The SoC inside camera has Dual-core ARM Cortex-A7 CPU, Image Signal Processor (ISP), intelligent video engine (IVE) and neural network interface engine (NNIE) with processing performance up to 1.0 TOPS.[31]

In this application, the native resolution of the sensor is Full HD (1920x1080) which is used as the main streaming channel. We create two down-scaled channels having a resolution of 416x224 for FD and 512x288 for LPR. The scaling operation is implemented by using the hardware resources, so there is no timing loss. The main and sub streaming channels goes to encoder module directly in order to decrease video latency and prevent fps drops. The video analytics applications can run at lower frame rates when compared to encoder fps. Therefore, the encoder pipelines and video analytics pipelines must be separate and the analytics results are added to metadata stream as the algorithms generate results. So the video analytics frame rates can be different from the streaming frame rates.

The different scales of the same frame captured from the sensor block are both sent to encoder, FD and LPR pipelines. The sensor output is YUV, but the networks need RGB image, so color space conversion is applied first. The FD network gets its down-scaled RGB image and generate face coordinates, face quality and landmark points as the outputs. The post processing steps defined in 4.1 are applied and final faces to be sent to server are decided. Faces assigned as appropriate for recognition are cropped from the full resolution image and converted to JPEG[32] image by using OpenCV.[33] The face coordinates are also sent to metadata generator to add in the rtsp stream.

In the same way, the LPR network gets its down-scaled image and generate LP coordinates. By using the LP coordinates, the LPs are cropped from the full resolution image. The cropped LP images are scaled to 94x36

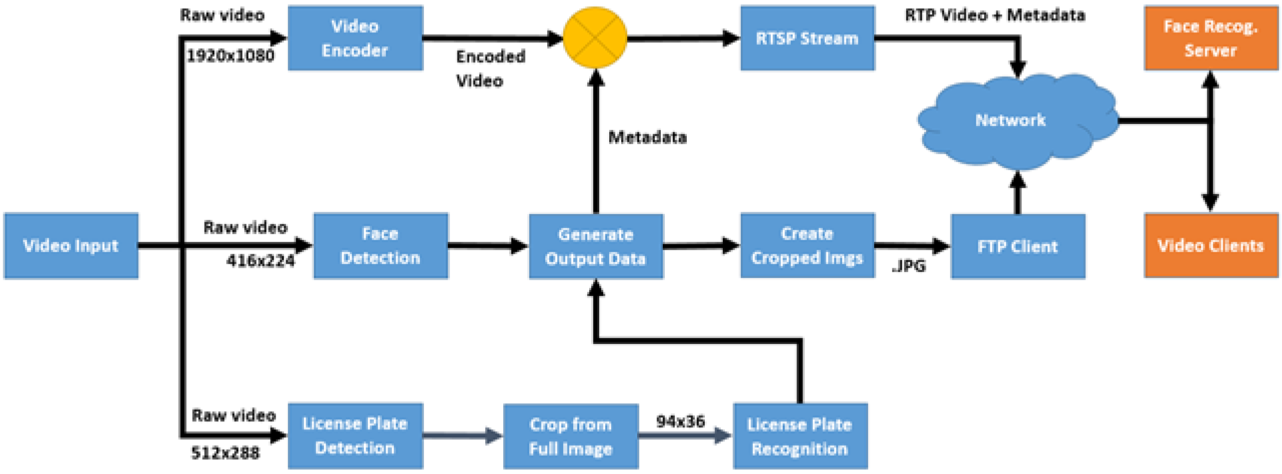

Figure 6. Proposed system-on-chip application pipeline

Figure 7. FD and LPR application at surveillance scenario

by using IVE since the input resolution must be fixed for the NPU. The cropped images are fed to recognition network and LP text data are obtained. As described in section 4.2, the post-processing steps are applied in order to filter out redundant data. The final LP images are converted to JPEG image and coordinates are sent to metadata generator to add in rtsp stream.

Here, we apply scaling and color space conversion operations. All these operations are implemented by using the hardware resources of SoC and OpenCV is compiled with ARM Neon support which provides further speed-up at OpenCV functions.

5. APPLICATIONS

In this section, we provide some use cases of the proposed detection and recognition pipelines with examples. As described in introduction section, combination of FD and LPR algorithms can be very useful for surveillance, security and traffic applications such as car theft, criminal search, access control, traffic management and road safety. Figure 7, shows an example of LPR and FD application for a general purpose surveillance application. With this configuration, it is possible to detect faces of pedestrians while reading license plates of the vehicles. A sample application that can be applied to traffic management, is to detect no helmet violation as shown in Figure 8. With a correct camera angle and proper illumination, it is even possible to detect LP and passengers' faces inside the vehicle at both day and low-light conditions as shown in Figures 9, 10. Combination of FD and LPR applications can also be used to detect car thefts as shown in Figure 11.

Figure 8. Detecting traffic violations - no helmet worn on motorcycle

Figure 9. FD inside vehicle and LPR application at daytime

Figure 10. FD inside vehicle and LPR application at low-light conditions

Figure 11. Using proposed FD and LPR application for detecting car thefts

6. EXPERIMENTAL RESULTS

We measure the performance of LPR algorithm in terms of accuracy, FD algorithm in terms of average precision (AP) and execution timings on the chosen platform. The experiments are conducted by using the developed surveillance camera; the images are captured with the camera and the algorithms are run directly on the SoC inside camera.

For LPR evaluation, first we captured vehicle images by using a vehicle detection network running on the camera. All the images are given as input to a well known commercial LPR engine for reading LPs and filter redundant data such as vehicles with no or unreadable license plates. The vehicle images whose license plates are read by the reference LPR engine are recorded with filenames renamed as the license plate texts. Then these files are given as input to our LPR algorithm running on the SoC. The outputs are compared and classified as matching, non-matching and unread LPs with automated test scripts. If same readings are obtained from reference and our LPR engine, these LPs are classified as matching and accepted as correct reading. If LPR engines generate different results, these LPs are classified as non-matching and if the reference LPR engine generates a result, but the proposed can not generate a result, these are classified as unread LPs. Initially, we obtained 1990 non-matching and 450 unread license plates out of 23398 images as the output of automated tests. When visually checking the non-matching LPs one-by-one, it is seen that the vast majority of these LPs read correctly by the proposed method; only 173 out of 1990 images are misread. The test results and accuracy calculation are given in Table 1.

Table 1. Accuracy Rate of LPR Algorithm

Accuracy	Matching	Non-Matching	Unread	mAP
LPR	22775	173	450	%97.34

Table 2. Average Precision of FD on WiderFace Dataset

AP	Easy	Medium	Hard
MTCNN[34]	0.785	0.753	0.497
RetinaFace[35]	0.899	0.876	0.711
yolov5Face[36]	**0.937**	0.907	0.698
Proposed	0.926	**0.908**	**0.765**

Table 3. Execution Timings of the algorithms on SoC

Timing(ms)	Detection	Recognition*	Post-Processing	Total
LPR	37	7	3	47
FD	27	N/A	2	29
FD+LPR	64	7	5	76

Figure 12 shows some examples of LPs being misread by the proposed method. Misreadings can be divided into three groups: incorrect letter readings at two-line LPs, missing or extra characters, misidentified characters. Group 1 includes two line LPs which are rare (around ‰1-2) in Turkish highways. The recognition accuracy for two line LPs is measured around %60 at our tests since the character size is small when compared to regular LPs and LPRnet is originally developed for single line LPs. Group 2 contains missing or extra characters at LPs. For example "06AY8961" LP is recognized as "06AY896"; the "1" number at the end may interfere with the LP frame border and "06BIA134" LP is recognized as "06BA134" because "I" letter is thinner than other

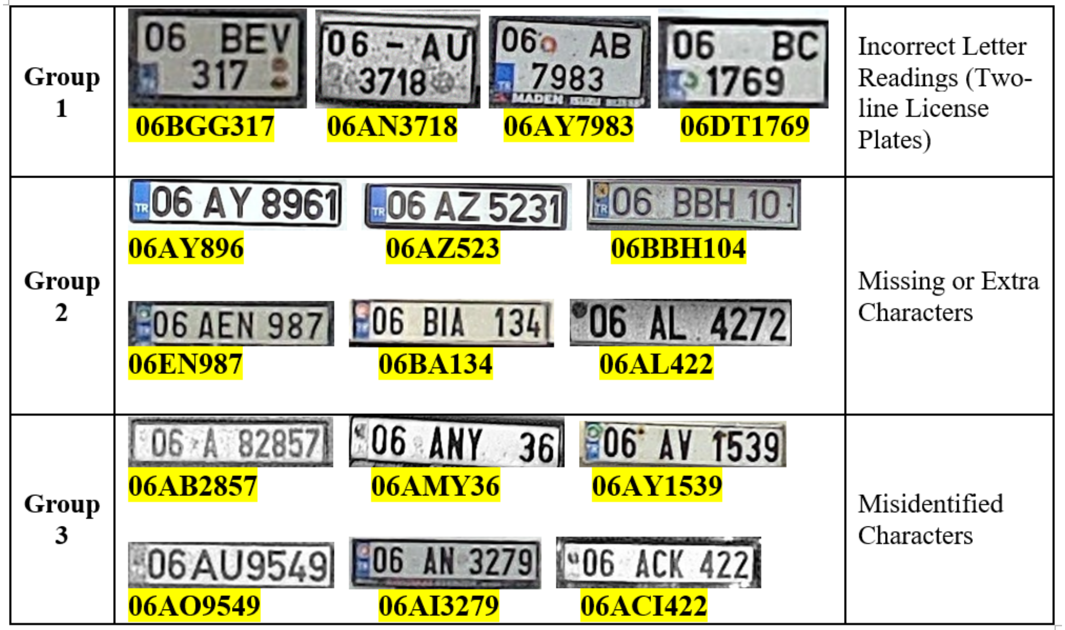

Figure 12. Several Examples of Misread LPs

characters. Group 3 includes misidentified characters; it can be seen from examples that sometimes "8" character may interfere with "B" character, or "V" character may interfere with "Y" character and "N or K" may interfere with "I" character.

It is important to note that that the accuracy value is calculated by using a single image for a vehicle and the outputs of the ML networks are directly used. Tracking and Majority voting operations can not be applied since there is only one image for a vehicle in this evaluation. When these post-processing operations are applied after ML networks generate result, it is expected and verified with some tests that the accuracy would increase further and the false readings given in Figure 12 disappear to some extent.

The accuracy of FD algorithm is first measured on a well known face detection dataset (WiderFace)[37] which includes three different types of faces, easy-medium and hard in terms of detect-ability. We also included three well performing and known algorithms, MTCNN,[34] RetinaFace[35] and yolov5Face.[36] We applied the algorithms in the literature with moderate network sizes that could fit in the SoC. In that manner, MTCNN is used as is without any size modifications, while RetinaFace and yolov5Face are applied in (640x640) and (416x416) resolutions respectively. Proposed approach is also applied in (416x416). The average precision values are presented in Table 2, where the proposed approach outperforms the other techniques in medium and hard face, while it is the runner-up method after yolov5Face that is rather a recent technique.

In Table 3, the average execution timings of the algorithms measured on the SoC are given. The average timing of LPR detection network is measured as 37 ms and the average timing of recognition network for one LP is measured as 7 ms. The recognition* given in the table is only applicable to LPR and measured for one LP in the image. If there is more than one LP detected in the image, the given timing must be multiplied with the number of LPs. With post-processing time of 3 ms, a total of 47 ms is obtained for LPR pipeline. For FD pipeline, we have only detection and post-processing timings which are 27 ms and 3 ms, respectively. The combination of FD and LPR timing is calculated as the sum of FD-only and LPR-only pipelines.

7. CONCLUSION

In this study, a combined FD/LPR application is presented which is further incorporated into a full HD surveillance camera for real-time analyses at the edge. The proposed system is able to run at real-time with high performance and high-accuracy. Each functionality runs at 25 frames-per-second individually, on the other hand execution rate drops to 13 fps when both applications are combined. This speed is still sufficient and within the limits of real-time applications in the wild. The proposed face detection model extracts face boundaries, five landmark points and quality in terms of face recognition through one pass of YoloV3 CNN. On the contrary, YoloV5 and LPRNet are combined and cascaded for License plate recognition that provide high accuracy. Tracking and post-processing steps applied in addition to detection-OCR networks that clear redundant data and improve accuracy. The uses cases and presented examples indicate that the fusion of outputs of FD and LPR algorithms can be very useful at several surveillance and security applications.

REFERENCES

[1] Chang Y.H Chen, S.K. Destech publications, inc. p. 21. isbn 9781605951508. *International Conference on Artificial Intelligence and Software Engineering (AISE2014)*, 2014.

[2] Top five biometrics: Face, fingerprint, iris, palm and voice. *Bayometric*, 2018.

[3] Divvala S. Girshick R. Farhadi A. Redmon, J. You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 779–788*, 2016.

[4] Redmon. J. and A. Farhadi. Yolov3: An incremental improvement. *arXiv*, 2018.

[5] Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, ChristopherSTAN, Liu Changyu, Laughing, tkianai, Adam Hogan, lorenzomammana, yxNONG, AlexWang1900, Laurentiu Diaconu, Marc, wanghaoyang0106, ml5ah, Doug, Francisco Ingham, Frederik, Guilhen, Hatovix, Jake Poznanski, Jiacong Fang, Lijun Yu , changyu98, Mingyu Wang, Naman Gupta, Osama Akhtar, PetrDvoracek, and Prashant Rai. ultralytics/yolov5: v3.1 - Bug Fixes and Performance Improvements, October 2020.

[6] Shi Y. Wang Z. Zhao, Y. The improved yolov5 algorithm and its application in small target detection. *Intelligent Robotics and Applications. ICIRA 2022. Lecture Notes in Computer Science(), vol 13458. Springer, Cham.*, 2022.

[7] S. Zherzdev and A. Gruzdev. Lprnet: License plate recognition via deep 517 neural networks. *Comput. Vis. Pattern Recognit., no. 6, pp. 248-253*, 2018.

[8] Ashu Kumar, Amandeep Kaur, and Munish Kumar. Face detection techniques: a review. *Artificial Intelligence Review, Rev 52*, pages 927–948, 08 2019.

[9] Wang Yang and Zheng Jiachun. Real-time face detection based on yolo. In *2018 1st IEEE International Conference on Knowledge Innovation and Invention (ICKII)*, pages 221–224, 2018.

[10] Sefa Burak Okcu, Burak Oğuz Özkalaycı, and Cevahir Çığla. An efficient method for face quality assessment on the edge. In Adrien Bartoli and Andrea Fusiello, editors, *Computer Vision – ECCV 2020 Workshops*, pages 54–70, Cham, 2020. Springer International Publishing.

[11] Yongkang Wong, Shaokang Chen, Sandra Mau, Conrad Sanderson, and Brian C. Lovell. Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition. In *IEEE Biometrics Workshop, Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 81–88. IEEE, June 2011.

[12] Junqing Tang, Li Wan, Jennifer Schooling, Pengjun Zhao, Jun Chen, and Shufen Wei. Automatic number plate recognition (anpr) in smart cities: A systematic review on technological advancements and application cases. *Cities*, 129:103833, 2022.

[13] Oladapo Ibitoye, Temitayo Ejidokun, Olufemi Dada, and Olusegun Omitola. Convolutional neural network-based license plate recognition techniques: A short overview. In *2020 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 1529–1532, 2020.

[14] Shreya Raj, Yash Gupta, and Ruchika Malhotra. License plate recognition system using yolov5 and cnn. In *2022 8th International Conference on Advanced Computing and Communication Systems (ICACCS)*, volume 1, pages 372–377, 2022.

[15] Sasakorn Rattanawong, Gee-Sern Hsu, and Sheng-Luen Chung. Thailand license plate detection and recognition. In *2021 25th International Computer Science and Engineering Conference (ICSEC)*, pages 116–121, 2021.

[16] Yu-Liang Chiang. Real time license plate detection based on machine learning. Master's thesis, The University of Wisconsin-Milwaukee, 2021.

[17] Aman Jain, Jatin Gupta, Somya Khandelwal, and Surinder Kaur. Vehicle license plate recognition. 4:15–21, 06 2021.

[18] Piyush Batra, Imran Hussain, Mohd Abdul Ahad, Gabriella Casalino, Mohammad Afshar Alam, Aqeel Khalique, and Syed Imtiyaz Hassan. A novel memory and time-efficient alpr system based on yolov5. *Sensors*, 22(14), 2022.

[19] Hamdan Taleb, Zhipeng Li, Changan Yuan, Hongjie Wu, Xingming Zhao, and Fahd A. Ghanem. An effective method for yemeni license plate recognition based on deep neural networks. In De-Shuang Huang, Kang-Hyun Jo, Junfeng Jing, Prashan Premaratne, Vitoantonio Bevilacqua, and Abir Hussain, editors, *Intelligent Computing Methodologies*, pages 304–314, Cham, 2022. Springer International Publishing.

[20] Diego Hernán Suntaxi Dominguez, Samantha Carolina Quintanchala Sandoval, and Manuel Eugenio Morocho-Cayamcela. End-to-end license plate recognition system for an efficient deployment in surveillance scenarios. In Álvaro Rocha, Carlos Ferrás, Abel Méndez Porras, and Efren Jimenez Delgado, editors, *Information Technology and Systems*, pages 697–704, Cham, 2022. Springer International Publishing.

[21] Shubham Srivastava, Ajay Verma, and Shekhar Sharma. Optical character recognition techniques: A review. In *2022 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*, pages 1–6, 2022.
[22] Jacobo González-Cepeda, Álvaro Ramajo, and José María Armingol. Intelligent video surveillance systems for vehicle identification based on multinet architecture. *Information*, 13(7), 2022.
[23] Sérgio Montazzolli and Claudio Jung. Real-time brazilian license plate detection and recognition using deep convolutional neural networks. In *2017 30th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 55–62, 2017.
[24] Shan Luo and Jihong Liu. Research on car license plate recognition based on improved yolov5m and lprnet. *IEEE Access*, 10:93692–93700, 2022.
[25] Ying-Nong Chen, Chin-Chuan Han, Cheng-Tzu Wang, Bor-Shenn Jeng, and Kuo-Chin Fan. The application of a convolution neural network on face and license plate detection. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 552–555, 2006.
[26] Ho G-F Fan K-C Chen Y-N, Han C-C. Facial/license plate detection using a two-level cascade classifier and a single convolutional feature map. *International Journal of Advanced Robotic Systems*, 2015.
[27] Chin-Chuan Han, Yan-Shin Tasi, Cheng-Ta Hsieh, and Chih-Hsun Chou. The interpolation of face/license-plate images using pyramid-based hallucination. In *43rd Annual 2009 International Carnahan Conference on Security Technology*, pages 239–244, 2009.
[28] Abayomi-Alli Adebayo. Development of an automatic facial and licence plate recognition system. 10 2015.
[29] Michael J. Thomas. Combining facial recognition, automatic license plate readers and closed-circuit television to create an interstate identification system for wanted subjects. Master's thesis, Naval Postgraduate School, 2015.
[30] R. Lanphier H. Schulzrinne, A. Rao. Rfc2326: Real time streaming protocol (rtsp). *RFC Editor, United States*, 1998.
[31] Hisilicon dv300 smart vision system-on-chip. https://www.hisilicon.com/en/products/smart-vision/consumer-camera/IOTVision/Hi3516DV300.
[32] G.K. Wallace. The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):xviii–xxxiv, 1992.
[33] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
[34] Zhang Z. Li Z. Zhang, K. and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. In *IEEE Signal Processing Letters*, page 1499–15032, 2016.
[35] Jiankang Deng, Jia Guo, Evangelos Ververas, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-shot multi-level face localisation in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5203–5212, 2020.
[36] Delong Qi, Weijun Tan, Qi Yao, and Jingfeng Liu. Yolo5face: Why reinventing a face detector. *CoRR*, abs/2105.12931, 2021.
[37] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.