future internet

Article

Deep Classifiers-Based License Plate Detection, Localization and Recognition on GPU-Powered Mobile Platform

Syed Tahir Hussain Rizvi [1,*], Denis Patti [1,*], Tomas Björklund [2], Gianpiero Cabodi [1] and Gianluca Francini [3]

1 Dipartimento di Automatica e Informatica (DAUIN), Politecnico di Torino, 10129 Turin, Italy; gianpiero.cabodi@polito.it
2 Dipartimento di Elettronica (DET), Politecnico di Torino, 10129 Turin, Italy; tomas.bjorklund@polito.it
3 Joint Open Lab, Telecom Italia Mobile (TIM), 10129 Turin, Italy; gianluca.francini@telecomitalia.it
* Correspondence: syed.rizvi@polito.it (S.T.H.R.); denis.patti@polito.it (D.P.); Tel.: +39-011-090-7048 (S.T.H.R. & D.P.)

Received: 14 September 2017; Accepted: 18 October 2017; Published: 21 October 2017

Abstract: The realization of a deep neural architecture on a mobile platform is challenging, but can open up a number of possibilities for visual analysis applications. A neural network can be realized on a mobile platform by exploiting the computational power of the embedded GPU and simplifying the flow of a neural architecture trained on the desktop workstation or a GPU server. This paper presents an embedded platform-based Italian license plate detection and recognition system using deep neural classifiers. In this work, trained parameters of a highly precise automatic license plate recognition (ALPR) system are imported and used to replicate the same neural classifiers on a Nvidia Shield K1 tablet. A CUDA-based framework is used to realize these neural networks. The flow of the trained architecture is simplified to perform the license plate recognition in real-time. Results show that the tasks of plate and character detection and localization can be performed in real-time on a mobile platform by simplifying the flow of the trained architecture. However, the accuracy of the simplified architecture would be decreased accordingly.

Keywords: convolutional neural network; visual analysis; embedded platforms; general purpose GPU; license plate detection

---

1. Introduction

The task of a vehicle license plate detector and recognizer is to identify a vehicle by its license plate. There are numerous practical applications of these detectors, such as in traffic surveillance, toll collection, etc. [1,2]. A complete automatic license plate recognition (ALPR) system comprises multiple blocks. First, the plate has to be detected and located. Then, characters can be segmented and recognized for the final or target application.

Different computer vision algorithms and image processing techniques can be used to detect a license plate and recognize its contents [3–8]. However, like other computer vision applications, a convolutional neural network can be used for highly accurate localization and recognition. Nonetheless, neural network-based classifiers are normally too computationally expensive, and can be difficult to realize on an embedded platform such as a mobile phone or tablet.

There are two phases to the utilization of a neural network, (i) Training and (ii) Inference. The training and inference (utilization) phases of a neural network have different requirements. The training phase of a network is computationally and memory expensive due the large number of samples, parallel classifications, and multiple repetitions required for learning. State-of-the-art architectures therefore require a powerful GPU board or even GPU clusters to perform the training phase, which cannot be performed on a mobile platform. However, a neural network in the inference phase typically only classifies single images. By exploiting the computational power of an embedded GPU and simplifying the flow of a trained neural architecture developed for running on desktop and server environments, this can become feasible on a mobile platform. Trained parameters of a neural network can be imported to replicate the same architecture on a mobile platform. By replicating the same flow on a mobile device, a deep neural network can be used in real-time when there is no network connectivity.

The remainder of the paper is organized as follows: Section 2 reviews the related work in the field of automatic license plate recognition (ALPR) systems using embedded platforms. The architecture of an ALPR system is presented in Section 3. Section 4 discusses the simplified flow of this ALPR system for the mobile platform. Section 5 presents the results, and finally, this work is concluded in Section 6.

2. Related Work

Several license plate detection and recognition approaches have been proposed recently [3–8]. However, there are very few realizations of these approaches on embedded platforms (e.g., mobile devices). For the realization of a classifier, several computational packages and libraries need to be installed on a system. Due to limited available resources on embedded devices, realizations of these approaches become a challenge [9]. Furthermore, an approach needs to be effective and efficient in order to satisfy the real-time constraints of an embedded platform.

A Jetson TX1 embedded board-based Korean license plate recognition system is presented in [10]. A simple convolutional neural architecture "AlexNet" is used for license plate recognition. High recognition accuracy is claimed. However, Korean license plates contain only digits, which are easier to detect compared to alphanumeric characters.

A digital signal processor (DSP)-based license plate recognition system is presented in [11]. Various hardware modules are interfaced with a DSP to fulfill the requirement of memory, input image acquisition, networking, etc. Furthermore, details related to execution time and classification accuracy that can be used for fair comparison are not discussed.

An embedded plate recognition system is proposed in [12]. This system is built through C language on a Linux operating system that can be extended for embedded applications. There are some limitations of this system, such as the level of rotation, light intensity, etc.

There are also other desktop implementations of neural network-based license recognition systems. However, the realization of a neural network architecture-based license recognition system on a mobile platform is not well explored. In this paper, a simplified flow for Italian license plate recognition on an embedded platform is presented. A neural network-based embedded framework is used to replicate a highly accurate automatic license plate recognition (ALPR) system [13,14]. Trained parameters are imported and used by a CUDA-based replicated network. By using a portable embedded device powered by a GPU, there is no need for internet connectivity for visual analysis. Furthermore, other resources of an embedded mobile device can be utilized (e.g., the camera) to capture the scene without any external device. The computational power of an embedded GPU can be exploited for the realization of multiple concurrent neural network-based image processing tasks.

3. Network Developed for Desktop and Server Environments

A fully convolutional neural network architecture is trained using a synthetic dataset containing images of a wide variation. It was tested on a dataset of real images to ensure that the learned features are robust to different image capturing conditions [14]. Torch computing framework was used to construct and train this architecture for a license plate recognition system. Two different networks were used for detection and localization of Italian plates and characters. Both networks (for plates and characters) are able to simultaneously perform the task of detection and localization using shared connections of convolutional layers, as shown in Figures 1 and 2. By training with the new dataset and minor modifications, the same classifiers can be used for multi-style license plate detection [15].

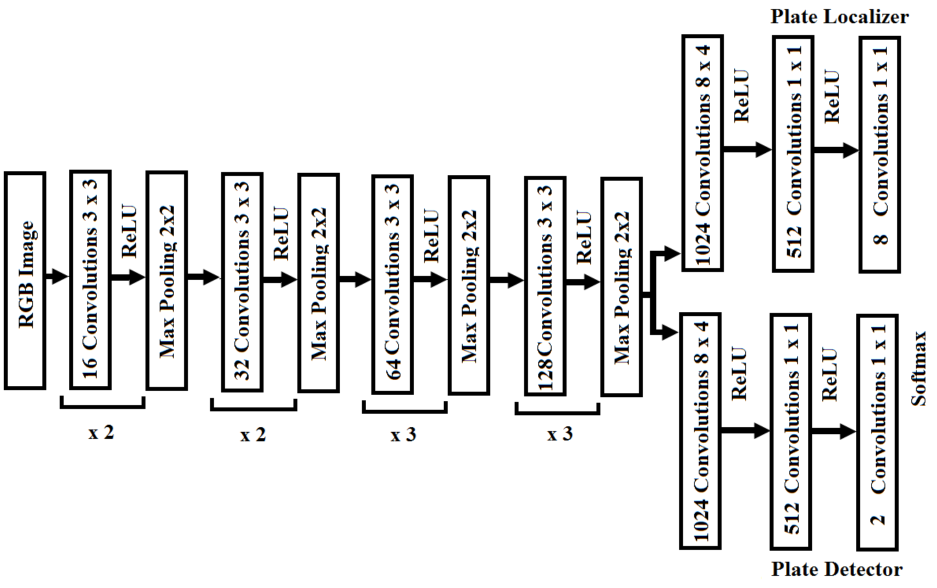

Figure 1. Italian plate detector and localizer.

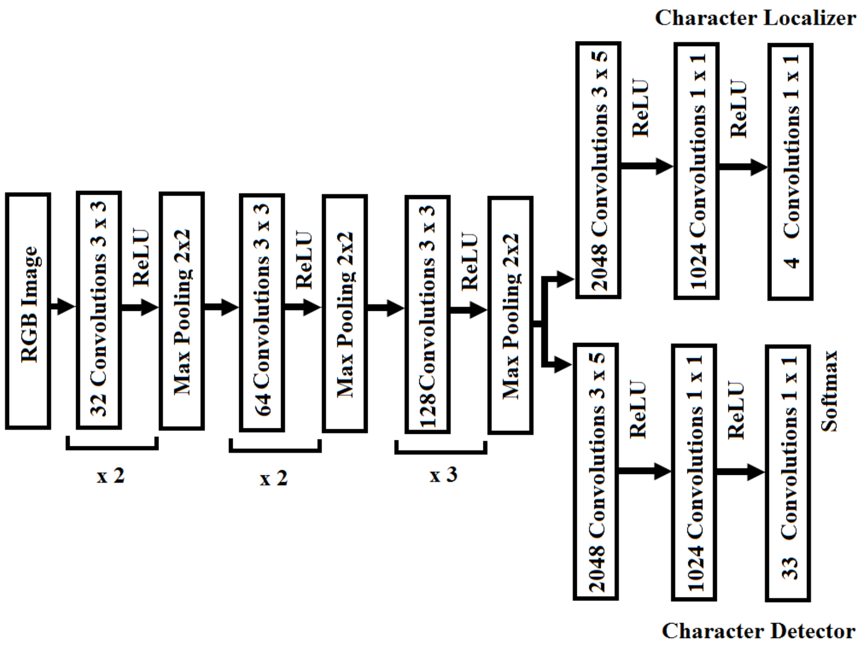

Figure 2. Character detector and localizer.

Instead of fully connected layers, these networks use weights in a convolution pattern together with max-pooling to decrease the spatial resolution and relevance, keeping only the strongest feature responses. By restructuring the fully connected layers into convolutional layers, an image of any size can be classified. The output of this network becomes a matrix where each element is a classification of a corresponding window in the original image.

Figure 3 shows the flow of a neural network-based Italian ALPR system developed for computationally powerful environments like a desktop workstation or GPU server.

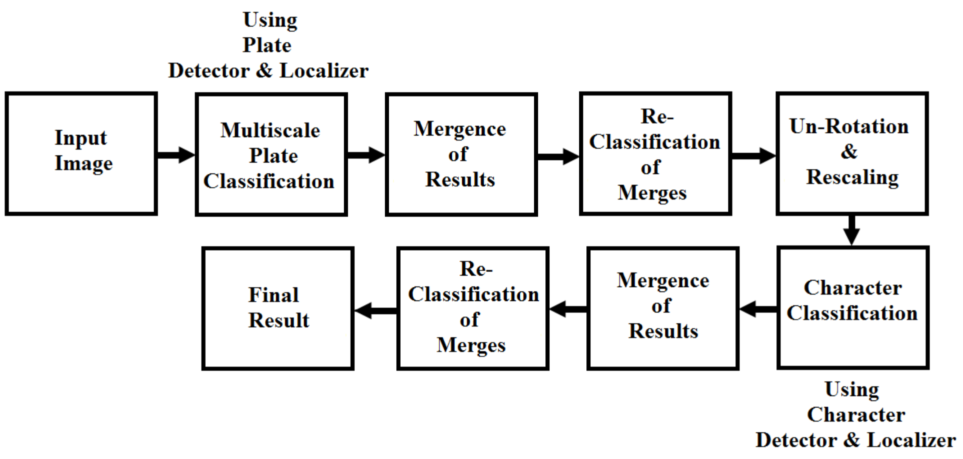

Figure 3. Flow of neural network-based automatic license plate recognition system.

This system has an accuracy of 98% on real photos. This highly precise license plate recognition system comprises multiple image processing blocks that perform different operations, including multiscale classification (detection and localization) of a single frame for the plates and then characters. This complete architecture and its components has high computational requirements. An embedded device such as a mobile platform has limited resources in terms of computational power and memory storage; it is very difficult to realize a computationally intensive neural architecture on the mobile platform. By simplifying the flow of the trained architecture and using an optimized framework for the neural networks, architectures like these can be realized on a mobile platform. However, there would be a tradeoff between the accuracy and the execution time for the simplified architecture.

4. Simplified Flow for Mobile Platform

This section presents our mobile platform-based simplified flow of the ALPR system. Trained parameters of the plate and character networks discussed in the previous section are imported to be used by CUDA-based replicated networks [13]. A CUDA-based framework is used to accelerate and realize these license plate detection, localization, and recognition operations using the GPU of a mobile platform.

Flow of the realized architecture is shown in in Figure 4. The trained network for plate detection is implemented in its original form, to accurately localize a single vehicle license plate in a real-time complex scene. The trainable parameters are reduced to float from double datatype to improve performance and reduce memory consumption; the accuracy of plate detection remains the same as the source network on the test set. Since the complete network for plate and character recognition is complex and requires multiple classifications that can be time and memory consuming on an embedded device, the flow of the original network is simplified by sacrificing some recognition performance. A new block of error correction is added in the simplified flow in order to increase the classification accuracy.

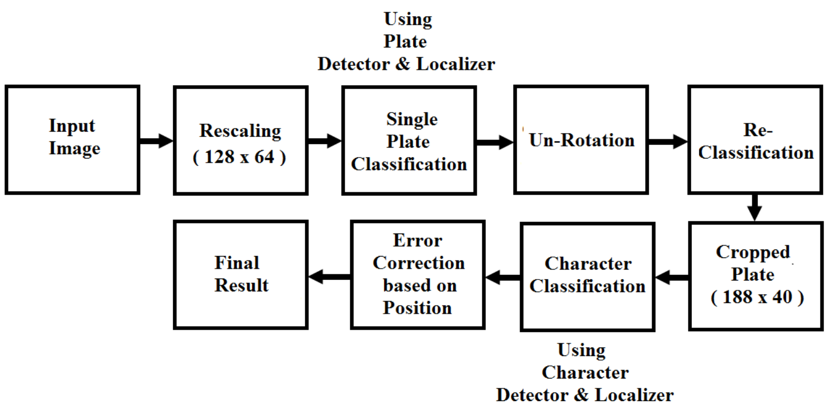

Figure 4. Simplified flow of neural network-based automatic license plate recognition system.

While the original system is completely scale-independent by performing multiple classifications by gradually scaling down the image in steps by a factor of 2, the mobile adaption is limited to a scale factor of 2 between the smallest and largest recognizable scale. This significantly reduces the computational complexity at the cost of requiring the user to apply appropriate zoom for a correct reading. The image needs to be captured such that the plate spans about 30–90% of the image; it may be rotated up to about 45 degrees. Light conditions should be such that the plate can be read in the image by a human; in darkness, a flash needs to be used. Some examples of classifications of such conditions are shown in Figure 5.

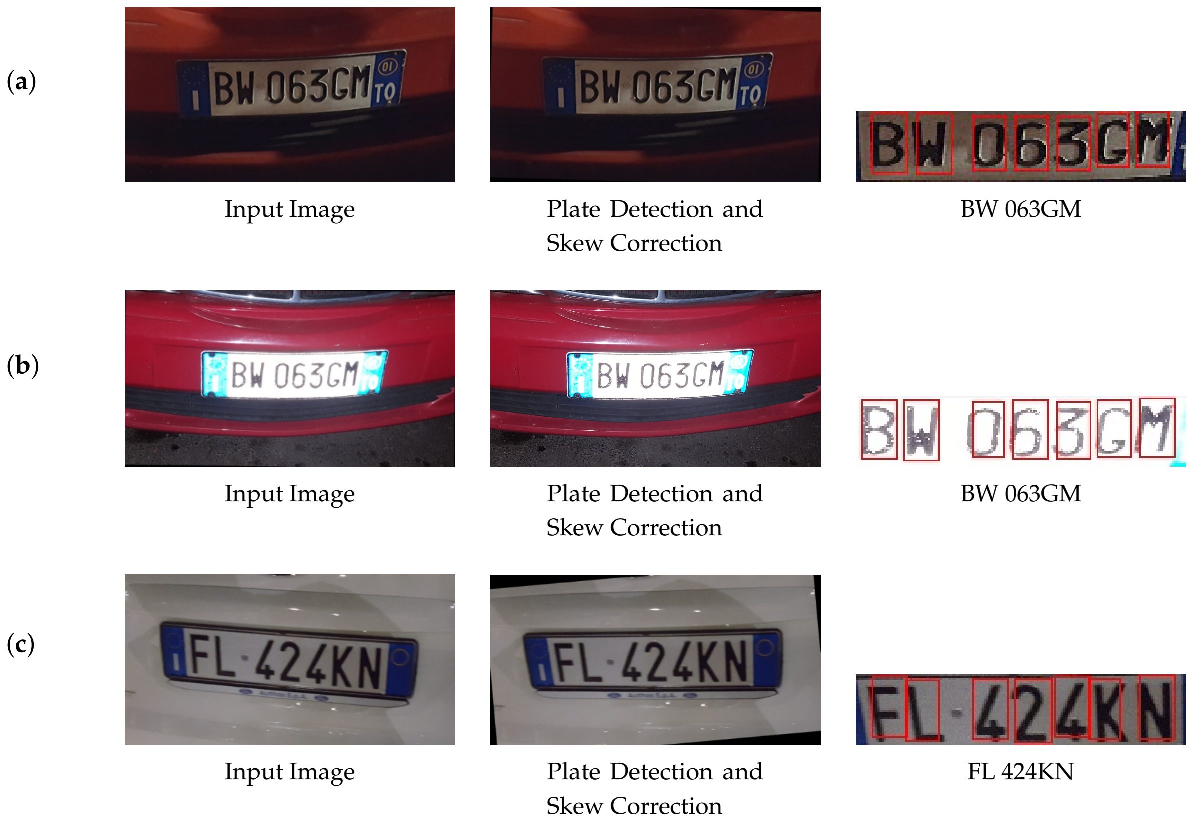

Figure 5. Examples of different challenging imaging conditions: (**a**) image captured at night with insufficient illumination; (**b**) image captured at night with over-exposure and reflections due to speedlight; (**c**) image captured at night and with perspective distortion.

4.1. Flow of Plate Detection and Localization

First of all, the input image is resized as shown in Figure 6. The flow of the trained neural architecture discussed in Section 3 is simplified by rescaling the input image (128 × 64) to allow for a single license plate classification per image. Then, the plate classification (detection and localization) is performed to find the position of the plate inside the input image, if a plate is detected. This network is composed of 16 convolutional layers. The first 10 convolutional layers of this network provide shared connections for simultaneous plate detection and localization, as shown in Figure 1. The remaining six layers are for either the plate detection or localization (three layers for each task). The plate detector has two outputs, defining the detection of a plate or no plate, while the output of the localizer consists of eight values that represent the x- and y-coordinates of the four points that enclose the plate.

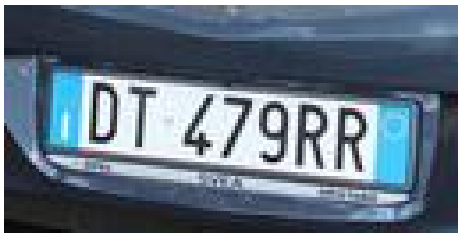

Figure 6. Rescaled input image.

The plate classifier normally generates a parallelogram. To perform the character classification, the input image is un-rotated to obtain a perfect rectangle. With other factors, the skewness of a detected plate also affects the performance of character classification. There are some skewness approaches that can be employed for reliable skew correction [16].

After rotation (skew correction), it is necessary to obtain the updated coordinates of the detected plate. It can be performed in two different ways: by computing the coordinates based on the rotation or by reclassifying the plate. While plate reclassification is more computationally expensive, it also provides a more accurate localization, which is necessary to obtain better character detection and localization. After this re-classification, the detected plate is cropped from the original image based on the obtained coordinates; this yields a crop where the plate and characters have a fixed location and scale (given an accurate plate localization), in order to provide good conditions for the character classifier and enable it to be simplified.

4.2. Flow of Character Detection and Localization

The next part of the ALPR system is detection and localization of the characters in the cropped image of the detected plate. The character classification network is also a fully convolutional neural network, comprised of 12 convolutional layers. Similar to the plate classifier, it simultaneously performs the detection and localization of characters. The character detector provides 33 output values per classification. These values represent the probability for each of the 32 alpha-numeric characters valid for Italian license plates or no character present (the 33rd output). The output of the character localizer comprises four values (the top left and the bottom right coordinates (x,y)) that represent the rectangle that encloses the detected character. Although this localization is not required for a correct classification of the license string, it provides a graphical output specifying the location of each character.

First of all, the crop from plate classifier is resized (188 × 40) to provide appropriate character classifications (note that this crop is of higher resolution than the input image for plate classification, in which the resolution is not necessarily required to be good enough to read the characters). The network has three max-pooling layers, the stride between classifications are eight pixels, and by rescaling the crop to a height of 40 pixels the classifications are reduced to a single line, significantly reducing the complexity. Maintaining the aspect ratio, this yields a crop of 188 × 40 pixels, which, given the stride length of eight, yields 21 partly overlapping classifications (as shown in Figure 7).

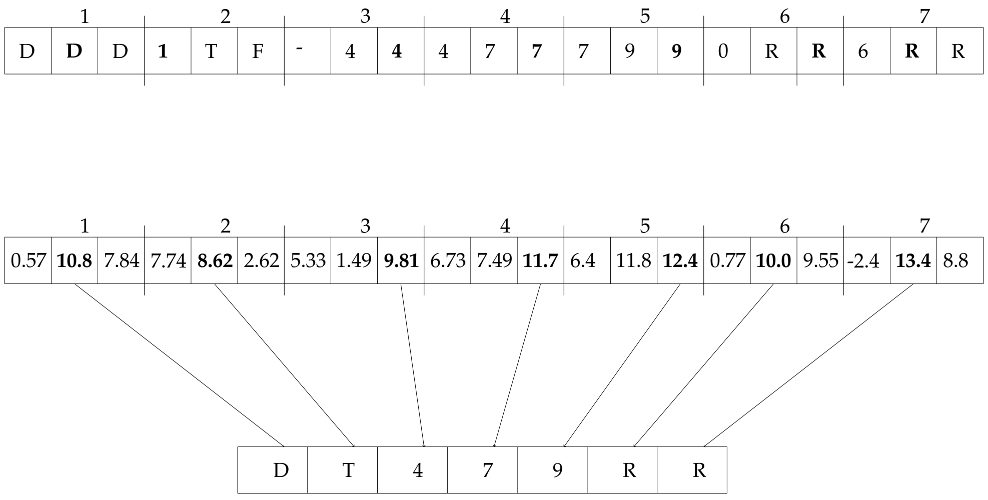

Figure 7. Classifications of overlapping slots and selection of characters for the final result.

A single classification (slot) represents a 24 × 40 portion of the image which may contain a character. The detected character in each classification (slot) is decided by the highest value of the detector output (out of 33 values). These 33 values represent the confidence ratings for each of the digits (0 to 9), Italian letters (22), and a background (null) value. These outputs are not softmaxed, as these values can be directly used for classification.

A generic Italian license plate contains seven characters (three digits and four alphabetic characters). This simplified ALPR system must select seven characters from the 21 outputs (classifications) generated by the network. To generate the final result, a trivial heuristic approach is adopted in the proposed flow. This heuristic method divides the 21 classifications (slots) into seven small clusters composed of three slots each. Note that the slots are overlapping, and depending on the accuracy of the plate localization, the same slot is not always the most centered. The classification corresponding to the highest confidence rating within each cluster is selected for the final result, as shown in Figure 7.

Furthermore, the size of the cropped image influences the result given by the character detector. By choosing an offset on the plate localization defining the character input crop, the accuracy and the number of classifications detected by the character classifier can be varied. Figure 8 shows that the cropped plate with an offset of four pixels provides exactly seven classifications (characters) of an Italian license plate.

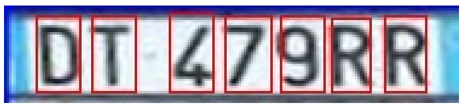

Figure 8. Output image.

5. Results

First of all, the execution times were examined for the original network discussed in Section 3 on two different types of GPU environments: on a desktop workstation equipped with a Quadro K2200 GPU card and a powerful Jetson TX1 embedded board. The Quadro K2200 and the Jetson TX1 have 640 and 256 CUDA cores, respectively. The execution time of these torch-based networks was measured for the performance analysis of the original network. An input image size of 640 × 480 was selected for multi-scale classifications of license plates. As shown in Table 1, there are two scenarios in the original plate recognition system. If a plate is not detected in the frame, then the remaining steps of plate and character classification would not be performed, as depicted in Figure 3. This significantly reduces the computation time, as can be seen in Table 1. Conversely, if a plate is detected, then all processing blocks would be executed to find the final results that include processes like the merging of results and reclassifications. As mentioned earlier, the original plate recognition system is highly precise and has an accuracy of 98% on real photos. However, it can be observed that the execution time of the original network on an embedded platform like Jetson TX1 board was higher (1.45 + 3.24 = 4.69 s) in the case of positive classification. The results are listed in Table 1.

Table 1. Execution times of original and simplified networks.

	Quadro K2200		**Jetson TX1**		**Nvidia Shield K1**
Network	**Original**	**Simplified**	**Original**	**Simplified**	**Simplified**
Plate Classification Time	150 ms	26 ms	1450 ms	244 ms	∼300 ms
Character Classification Time	510 ms	27 ms	3240 ms	248 ms	∼250 ms

This execution time can be much higher on the mobile platforms that are usually less powerful than embedded boards and desktop workstations. By simplifying the original flow and eliminating the computationally intensive tasks such as merging results over and multi-scale classifications, this execution time can be reduced significantly at the expense of some classification accuracy.

The simplified flow proposed in Section 4 was first evaluated on Quadro K2200 and Nvidia Jetson TX1 embedded boards for comparison of performance with the original network in terms of execution time. Results show that the simplified flow significantly improved the execution time. Then Nvidia Shield K1 tablet was used to measure the performance. This mobile device has a Tegra K1 GPU with 192 CUDA cores. The execution time of the plate detector and localizer was approximately 300 milliseconds for an input image of 128×64. Additionally, the character detector classified a cropped license plate of size 188×40 in around 250 milliseconds. Results show that by using input images of smaller dimension and single-scale classification, the execution time of an automatic license recognition system can be decreased.

Furthermore, an embedded platform like a mobile device has a limited storage capacity; thus, the size of the used framework is also an important constraint to consider. In the case of deep classifiers, enough memory space is required to store the trained parameters, the used framework (torch), and the required computational packages. By using a CUDA-only replicated network, there is no need to install extra computational packages. Furthermore, by converting the trained parameters to float data type, the size of all parameters is reduced by half. Table 2 lists the size of the trained imported from Torch computing framework and converted parameters for CUDA-based simplified flow.

Table 2. Memory consumption of trained and converted parameters.

	Trained Parameters	**Converted Parameters**
Network	**Double**	**Float**
Plate Classifier	75.70 MB	37.85 MB
Character Classifier	65.63 MB	32.81 MB

However, the accuracy of this simplified architecture was indeed decreased. Classification accuracy of a single character detection and full plate detection (all seven characters) is listed in Table 3 for different offsets. The testing dataset consists of 788 crops of aspect ratio 2:1 roughly centered around Italian rear license plates. These are real-world images downloaded from the internet (http://platesmania.com/—the database is available as supplementary material) and are publicly available.

Table 3. Classification accuracy of character detector and localizer network.

Offset for Cropping Detected Plates	Full Plate Accuracy	Single Character Accuracy
0	17%	68%
2	28%	78%
4	30%	82%
8	25%	79%
4 (+2) (21 detected characters)	54%	90%
4 (+2) + Error correction	61%	92%
Original Network	94%	98%

Based on the best offset value (=4), the result generated by the plate detector was moved two pixels to the right. The results of this experiment are listed in Table 3. It shows that this change in offset further increased the classification accuracy. Different offset and shifting values were tested, and results show that the best result was obtained by cropping the detected plate with offset four outside the localization classification and shifted two pixels to the right.

Furthermore, an error correction step was added on the detected characters, based on the format of Italian license plates. Italian plates have a specific format: two letters, three digits, and then again two letters, as shown in Figure 8. Error correction based on the position of the character was applied to improve the accuracy of the simplified flow. Table 4 shows some rules defined for correction, and Table 3 also lists the classification accuracy after error correction. The results show that there was an increase of two percent in single character accuracy and an increase of 13 percent in full plate accuracy after error correction.

Table 4. Character correction based on position.

Original Character	Correction
B	8
D	0
Z	2
J	1
G	6

Furthermore, the temporal redundancy approach can be employed for classification using multiple frames [17]. Instead of a single frame, multiple frames can be processed and averaged using the simplified proposed flow of the ALPR system for accuracy improvement.

6. Conclusions

This paper presents the realization of an automatic license plate recognition system on a mobile platform by simplifying the flow of a trained neural architecture developed for running on desktop and server environments. The performance of this already-trained architecture was tested on two different GPU platforms (Quadro K2200 and Jetson TX1 embedded board). Performance analysis shows that the neural network-based ALPR system cannot be realized on a mobile platform in its original form. By reducing the resolution of the input image and architectural complexity, license plate recognition was performed in real-time on an Nvidia Shield K1 tablet. However, simplification and reduction of the original components directly affected the classification accuracy. Furthermore, an error correction was employed in simplified flow to increase the accuracy.

The accuracy of this mobile platform-based license recognition system can be further improved by employing multi-scale classification, merging, etc., from the neural architecture developed for computationally powerful environments and by increasing the resolution of the input image. To decrease the execution time of these computational tasks, other optimized frameworks for embedded devices having efficient memory transfer schemes and computational algorithms can be exploited [18–20].

Acknowledgments: This research work is funded by Telecom Italia and performed at Joint Open Lab, Telecom Italia, Torino, Italy. We would also like to thank the Attilio Fiandrotti for his helpful comments and suggestions.

Author Contributions: Syed Tahir Hussain Rizvi, Denis Patti and Tomas Björklund conducted the experiments and worked on the draft of the paper. Gianpiero Cabodi and Gianluca Francini are the academic tutors. They coordinated, supervised and approved the entire work.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lawlor, S.; Sider, T.; Eluru, N.; Hatzopoulou, M.; Rabbat, M.G. Detecting Convoys Using License Plate Recognition Data. *IEEE Trans. Signal Inf. Process. Netw.* **2016**, *2*, 391–405.
2. Suryatali, A.; Dharmadhikari, V.B. Computer vision based vehicle detection for toll collection system using embedded Linux. In Proceedings of the 2015 International Conference on Circuits, Power and Computing Technologies (ICCPCT-2015), Nagercoil, India, 19–20 March 2015; pp. 1–7.
3. Yuan, Y.; Zou, W.; Zhao, Y.; Wang, X.; Hu, X.; Komodakis, N. A Robust and Efficient Approach to License Plate Detection. *IEEE Trans. Image Process.* **2017**, *26*, 1102–1114.
4. Issaoui, S.; Ejbeli, R.; Frikha, T.; Abid, M. Embedded approach for edge recognition: Case study: Vehicle registration plate recognition. In Proceedings of the 2016 13th International Multi-Conference on Systems, Signals & Devices (SSD), Leipzig, Germany, 21–24 March 2016; pp. 336–341.
5. Gou, C.; Wang, K.; Yao, Y.; Li, Z. Vehicle License Plate Recognition Based on Extremal Regions and Restricted Boltzmann Machines. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 1096–1107.
6. Wafy, M.; Madbouly, A.M.M. Efficient method for vehicle license plate identification based on learning a morphological feature. *IET Intell. Transp. Syst.* **2016**, *10*, 389–395.
7. Corneeto, G.L.; da Silva, F.A.; Pereira, D.R.; de Almeida, L.L.; Artero, A.O.; Papa, J.P.; de Albuquerque, V.H.C.; Sapia, H.M. A New Method for Automatic Vehicle License Plate Detection. *IEEE Latin Am. Trans.* **2017**, *15*, 75–80.
8. Chen, Y.; Zhao, D.; Lv, L.; Li, C. A visual attention based convolutional neural network for image classification. In Proceedings of the 2016 12th World Congress on Intelligent Control and Automation (WCICA), Guilin, China, 12–15 June 2016; pp. 764–769.
9. Calderon, J.A.F.; Vargas, J.S.; Pérez-Ruiz, A. License plate recognition for Colombian private vehicles based on an embedded system using the ZedBoard. In Proceedings of the 2016 IEEE Colombian Conference on Robotics and Automation (CCRA), Bogota, Colombia, 29–30 September 2016; pp. 1–6.
10. Lee, S.; Son, K.; Kim, H.; Park, J. Car plate recognition based on CNN using embedded system with GPU. In Proceedings of the 2017 10th International Conference on Human System Interactions (HSI), Ulsan, Korea, 17–19 July 2017; pp. 239–241.
11. Luo, X.; Xie, M. Design and Realization of Embedded License Plate Recognition System Based on DSP. In Proceedings of the 2010 Second International Conference on Computer Modeling and Simulation, Sanya, Hainan, China, 22–24 January 2010; pp. 272–276.
12. Neto, E.C.; Gomes, S.L.; Filho, P.P.R.; de Albuquerque, V.H.C. Brazilian vehicle identification using a new embedded plate recognition system. *Measurement* **2015**, *70*, 36–46.
13. Rizvi, S.T.H.; Cabodi, G.; Patti, D.; Francini, G. GPGPU Accelerated Deep Object Classification on a Heterogeneous Mobile Platform. *Electronics* **2016**, *5*, 88.
14. Björklund, T.; Fiandrotti, A.; Annarumma, M.; Francini, G.; Magli, E. Automatic License Plate Recognition with Convolutional Neural Networks Trained on Synthetic Data. In Proceedings of the IEEE 19th International Workshop on Multimedia Signal Processing (MMSP 2017), Luton, UK, 16–18 October 2017, in press.
15. Elbamby, A.; Hemayed, E.E.; Helal, D.; Rehan, M. Real-time automatic multi-style license plate detection in videos. In Proceedings of the 2016 12th International Computer Engineering Conference (ICENCO), Cairo, Egypt, 28–29 December 2016; pp. 148–153.

16. Nguyen, C.T.; Nguyen, T.B.; Chung, S.T. Reliable detection and skew correction method of license plate for PTZ camera-based license plate recognition system. In Proceedings of the 2015 International Conference on Information and Communication Technology Convergence (ICTC), Jeju, Korea, 28–30 October 2015; pp. 1013–1018.
17. Gonçalves, G.R.; Menotti, D.; Schwartz, W.R. License plate recognition based on temporal redundancy. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 2577–2582.
18. Rizvi, S.; Cabodi, G.; Francini, G. Optimized Deep Neural Networks for Real-Time Object Classification on Embedded GPUs. *Appl. Sci.* **2017**, *7*, 826.
19. Wang, P.; Cheng, J. Accelerating Convolutional Neural Networks for Mobile Applications. In Proceedings of the 2016 ACM on Multimedia Conference, Amsterdam, The Netherlands, 15–19 October 2016.
20. Lavin, A.; Gray, S. Fast Algorithms for Convolutional Neural Networks. CoRR. 2015. Available online: http://arxiv.org/abs/1509.09308 (accessed on 18 May 2016).