One-shot license plate recognition using YOLOv2

Xavier Querol Bassols

Mentor: Dimosthenis Karatzas

June 2022

1 INTRODUCTION

One of the most important services in urban public policy in recent years is mobility management. Decision-making in this area is very delicate as the challenges to be faced are very complicated: management of density, pollution, and time spent by the user during transportation. Quality and efficiency, two of the clear goals in any public policy or measure, become even more important in the cities of the present and especially in the cities of the future.

Most part of the regulations being implemented by local councils in terms of mobility are linked to the reduction of motorized private transport. Many of them have one thing in common: the need to identify the vehicle. Goals can be very varied. Here are some examples:

- Restricted entry to the city, neighborhood, street
- Parking time limit
- Speed limit

The identifier of a vehicle is its registration number and to read it, a system capable of automatic recognition of the vehicle is essential. Automatic license plate recognition (ALPR) systems are already being used and their models have completely evolved in the last few years thanks to the success of convolutional neural networks (CNNs) on object and text detection and recognition. Not only the reading of one license plate is possible, but also on multiple vehicles at the same time.

2 OBJECTIVES

The main objective of this work, is to study the state of the art models and design a new one making it capable to detect and recognise multiple license plates on images taken in a real scenario. Concretely, these goals have been set:

- Learn which models are being used to detect and recognise license plates as well as their advantages and disadvantages.
- Build a new model to detect and recognise license plates.
- Train and evaluate the model built.
- Conclude if the model is useful for this task.

3 RELATED WORK

Most license plate recognition models have a first part, license plate detection (LPD), that finds the license plate bounding box, and then another part, license plate recognition (LPR), which recognises the license plate text in the detected areas of the image. Nowadays, new object detection methods based on transformers, such as DeTR and variants have been proposed [2]. Despite this LPD still uses the typical models of object detection, such as R-CNN, both simple and Fast or Faster [9, 12], or different versions of YOLO [7, 8, 6]. Despite this, in this field, the newest object detection methods based on transformers, such as DeTR and variants have not been used yet. It is important to say that all have an initial backbone with CNN. In the LPR part, there are models that first segment the different letters and then recognize them with OCR models [7], while there are other more accurate models that do a recognition by detection also using YOLO [8, 6, 12] or BRNN's [9].

More specifically, Laroca et al. use an improvement of YOLOv2 for vehicle detection, Fast-YOLOv2 for LP detection and CR-NET for LP recognition. RPNet is a CNN-based model with ROI pooling where at the end it has a fully connected layer for each character [16]. EILPR is also a CNN-based model with ROI pooling but uses its own system to recognize the license plate. All models follow this trend of first finding the license plate and then finding the digits, either with YOLO, R-CNN or ROI pooling [15, 16]. However, it is necessary to distinguish especially

those models that allow to find multiple license plates [9, 7, 8] from those that can only find one [15, 16, 11, 4, 1, 12].

There is no state of art of a model that uses YOLO end-to-end in order to read license plates. To explore whether this is possible or not a modification of the YOLO could be done. Actually, Andrés Mafla et al.[3, 10] use a modification of the YOLOv2 model to recognize different texts in an an end-to-end manner, obtaining as output the location of the different texts and which letters appear in their transcription. They represent the words using a Pyramidal Histogram Of Characters (PHOC), where words are represented by what letters and pairs of them they have. In their paper demonstrate how the trained model can represent words it has never seen before. This would be easily replicated by copying the model and fine-tuning it, changing the PHOC representation so that it returns the different symbols of the license plates.

Depending on whether they are trained and operated separately, they can be considered end-to-end models or not. When end-to-end models need to learn everything at once, require many more images to do so. In license plate recognition, the most popular database is the Chinese City Parking Dataset (CCPD), with 250000 images of Chinese license plates. 200000 of these images are in good conditions, and the 50000 remaining are atypical ones (blurred, tilted, rotated, with strange weather, etc). Due to its size it is a good database to train a model. Despite its volume, it has no images with more than one vehicle or without any, therefore more images should be included to cover the scenarios of more than one license plate or no license plate.

4 DATASET

In order to develop and check the reliability of the model is necessary to define a training set, a validation set and a test set. Images used in training and validation are from the CCPD, while the ones used in testing are not, so in this case the test set comes from a different domain than the validation and training sets. The CCPD is chosen because it has a lot of images of license plates and the documentation of how to use it is well explained.

4.1 Training and validation set

As mentioned above, the training set will consist of images from the CCPD because is the bigger dataset of license plates. Chinese license plates consist of 7 digits.

Figure 1: Chinese license plate

Analysing the statistics of this dataset is quite revealing of inherent biases, so below it is shown the probability of the possible characters for each digit for the total number of images in the dataset.

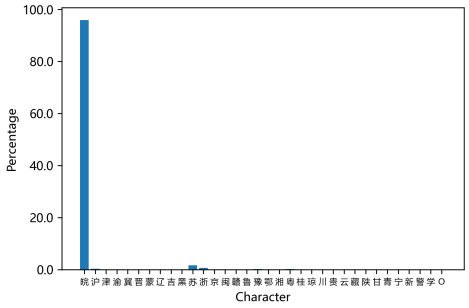

Figure 2: 1st digit

The first digit is almost always the character 皖, with a probability of $X\%$, which means that all images come from the same province (Anhui). This bias when training the model can cause you to always define this digit with this character.

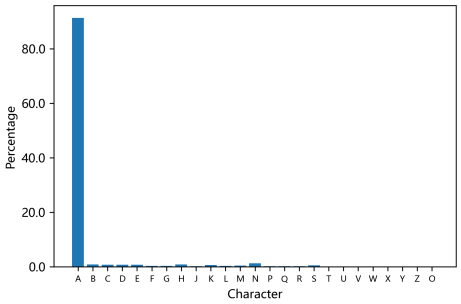

Figure 3: 2nd digit

As in the previous case, there is a character that has much more prominence on the dataset. In this case it is the letter A.

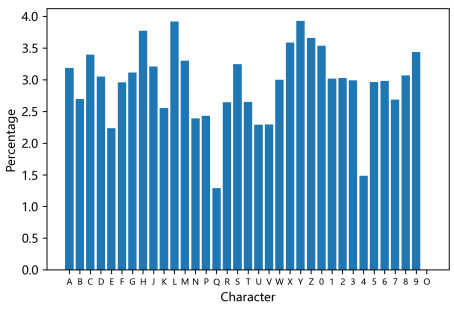

Figure 4: 3rd digit

The 3rd digit is the only one that the probability of all the characters oscillate in the same range.

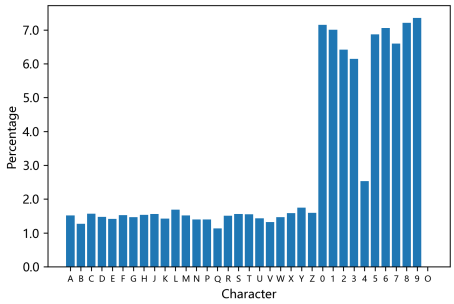

Figure 5: 4th digit

This digit can be a letter or a number. On one hand, all the letters and the number 4 have approximately the same probability of appearing. On the other hand, all the numbers except the number 4 more than triplicate it. This divergence with the number 4 with other numbers is most probably because these cultures consider 4 an unlucky number[17].

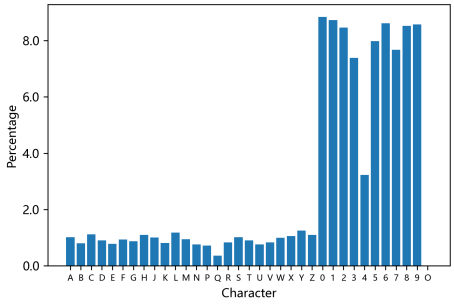

Figure 6: 5th digit

It occurs the same as in the 4th digit but all the numbers gain a little bit more of probability.

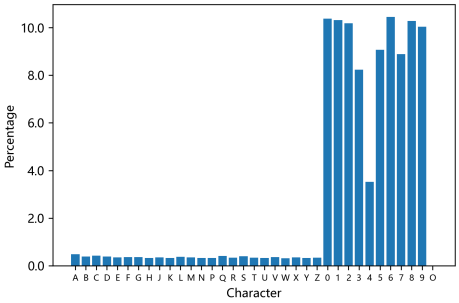

Figure 7: 6th digit

The probabilities of the letters is practically absent. On the contrary a few numbers achieve the 10% of probability.

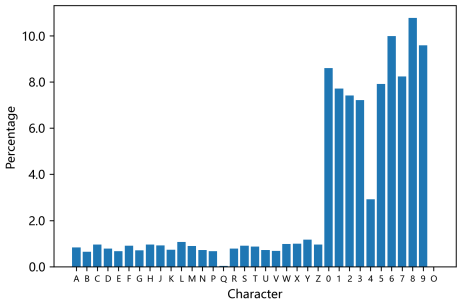

Figure 8: 7th digit

The probability distribution seems like a mix of 5th and 6th digit. Low probabilities for letters and high for numbers.

All the positions of the license plate have a distribution of probability different from each other. This can affect the model if it is not considered. Another characteristic disadvantage the dataset has is the position of the license plate in the image. As it can be seen in the figure below, the probability of its location being in the middle of the image is really high. This could affect the model because it could learn to only detect license plates that are in that part of the image. So, for example, plates that are close to the the corners won't be detected.

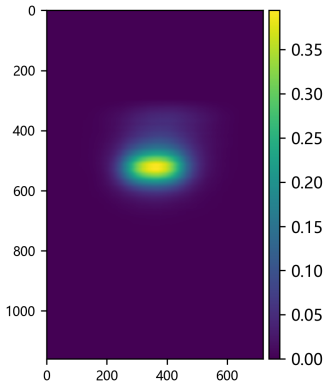

Figure 9: License plates positions

4.2 Transformations

As explained before, there are two inherent biases in the position of the license plates. They appear only one time and in the middle of the image. To overcome the above inherent biases of this dataset it is introduced more variability by creating synthetic images in two different ways:

- Combining multiple images creating a new one.
- Randomly distributing theses images, changing its sizes and positions.

In order to do that a new dataset is created combining one, two, three or four cars, distributed as randomly as possible in the image. Therefore, the training set is composed of 30000 images with 4 cars, 10000 with 3 cars, 17500 with 2 cars and 3000 with 1 car. So, the majority of images of the CCPD are used to create the 4-grid new images, and the others are used more or less equitably to build images with 2 or 3 of them. Also, some of them are used alone. Approximately a 5% of the images have randomly cars with no license plate, despite of the images where appear 1 car that are half and a half.

In the middle of the training, this dataset is expanded with 8000 images of 1 car per image of blurred, tilted, rotated and with strange weather photos, 2000 of each type. These photos are also from the CCPD.

90% of all those images created are used in training and 10% in validation.

The final dataset used can be very artificial but can serve to settle some weights in the model in case later there exists a large enough database with multiple registrations.

Figure 10: Image combination

4.3 Test set

As the objective is to identify license plates from images taken in a real scenario, to test if the model can work on license plates that appear in a different context than the original training set, a new test set has been created with images taken by the author. The pictures taken are of cars with European license plates, more concretely in Cerdanyola del Vallès and Sabadell (Catalonia). These license plates have a different format never seen before by the trained model, so different license plates that appear in the CCPD (some of the used in the validation set) have been embedded in them. This test set has 100 images from different positions, angles and with different number of cars.

Figure 11: License plate casting

Also, different sets of blurred, tilted, rotated and with strange weather photos, with 1000 each, will be used to test the model.

5 MODEL

5.1 Architecture

The proposed architecture, illustrated in Figure 12, is an adaptation of the YOLOv2[13, 14] object detection model modifying the last layer to carry out the license plate recognition task. Basically, it consists of a single shot CNN model that predicts at the same time bounding boxes and a compact license plate representation within them.

The YOLOv2 architecture combines 21 convolutional layers of of 3×3 filters with a leaky ReLU activation and batch normalization. In addition, 1×1 filters are placed between the 3×3 convolutions to compress the feature maps. There are 5 max pooling layers that double the number of channels and halve the spatial dimensions every step they appear. The backbone includes a pass-through layer from the second convolution layer and is followed by a final 1×1 convolutional layer with a linear activation with the number of filters matching the desired output tensor size for object

detection. Generally, the number of filters needed are the number of possible objects plus 4 (values for the box) plus 1 (for the confidence), and all this multiplied by the number of boxes the model can calculate. In Figure 12 there is represented the backbone with the output of the 3×3 convolution layers, the 1×1 convolution layers and the max-pooling, with blue, grey and red, respectively.

If the representation of each different license plate belongs to a different class, it will be needed $34x25x35x35x35x35x35$ (possibilities of each digit) plus 5 (box and confidence), multiplied by the number of boxes the model can calculate, filters.

To have a model capable to recognize license plates not seen at training time and computationally more efficient the last layer comprises multiple one hot classifiers, one for each digit on the license plate. This representation can represent $34x25x35x35x35x35x35$ different license plate numbers with only $34 + 25 + 35 + 35 + 35 + 35 + 35 = 234$ values. This way the model will learn how to identify characters in specific positions in the string independently.

The output tensor represented in Figure 12 has a dimension of $13x13x7x239$. Actually, it is a minimal representation of the initial image, where the 7×239 vector of each 13×13 grid are more or less the possible license plates in the corresponding part of the image. As YOLOv2 uses anchor boxes and width and height are used inside the model as the difference between them and the anchor boxes, in this model each vector commented below is seven times the representation of 239 (234 for digits + 5 for bounding box and confidence). So in this case, the model can find in each part of the image (more or less the proportional part of the 13×13 output grid) license plates with 7 different shapes (aspect ratios).

5.2 Anchor boxes

As in [4] and [1], to learn faster and reduce the distances between real bounding boxes and anchors, the ideal set of bounding boxes is calculated using k-means and requiring that for each bounding box annotation there exists at least another one with an intersection over union of at least 0:6. Figure 13 shows the 7 bounding boxes. Their relative dimensions are: [[0.13, 0.08], [0.25, 0.09], [0.13, 0.03], [0.17, 0.04], [0.36, 0.1], [0.19, 0.11], [0.19, 0.14]]. This reflects the fact that most license plates are almost horizontal (larger width than height).

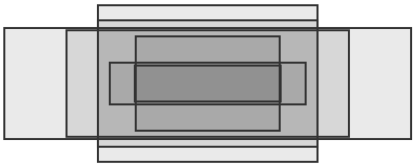

Figure 13: Anchor boxes

5.3 Losses

The total loss used to optimize the model is calculated as:

$$\begin{aligned} L(b, O, c, b', O', c') = &\lambda_{box} L_{box}(b, b') \\ &+ \lambda_{obj} L_{obj}(O, O') \\ &+ \lambda_{cls} L_{cls}(c, c') \end{aligned}$$

where λ_{box}, λ_{obj} and λ_{cls} are hyperparameters that need to be tuned.

Loss for bounding box regression:

$$\begin{aligned} L_{box}(b, b') = &\frac{1}{2m} \sum_{i=1}^{m} ((x^{(i)} - x'^{(i)})^2 \\ &+ \frac{1}{2m} \sum_{i=1}^{m} (y^{(i)} - y'^{(i)})^2 \\ &+ \frac{1}{2m} \sum_{i=1}^{m} (w^{(i)} - w'^{(i)})^2 \\ &+ \frac{1}{2m} \sum_{i=1}^{m} (h^{(i)} - h'^{(i)})^2 \end{aligned}$$

where x, y, w, h are the correct position and size offsets to an anchor bounding box, and x', y', w', h' its predictions. M is the number of boxes to compare.

Loss for objectness estimation:

$$L_{obj}(O, O') = \frac{1}{2m} \sum_{m=1}^{M} (O^{(i)} - O'^{(i)})^2$$

being O the probability of that bounding box containing an object (confidence) and O' its prediction (1 if model is sure and 0 if model it isn't). M is the number of boxes to compare.

Loss for classification:

$$L_{cls} = \sum_{n=1}^{N} \sum_{d=1}^{D} \sum_{c=1}^{C_d} w_{d,c} log \frac{exp(x_{n,d,c})}{\sum_{i=1}^{C_d} exp(x_{n,d,i})} y_{n,d,c}$$

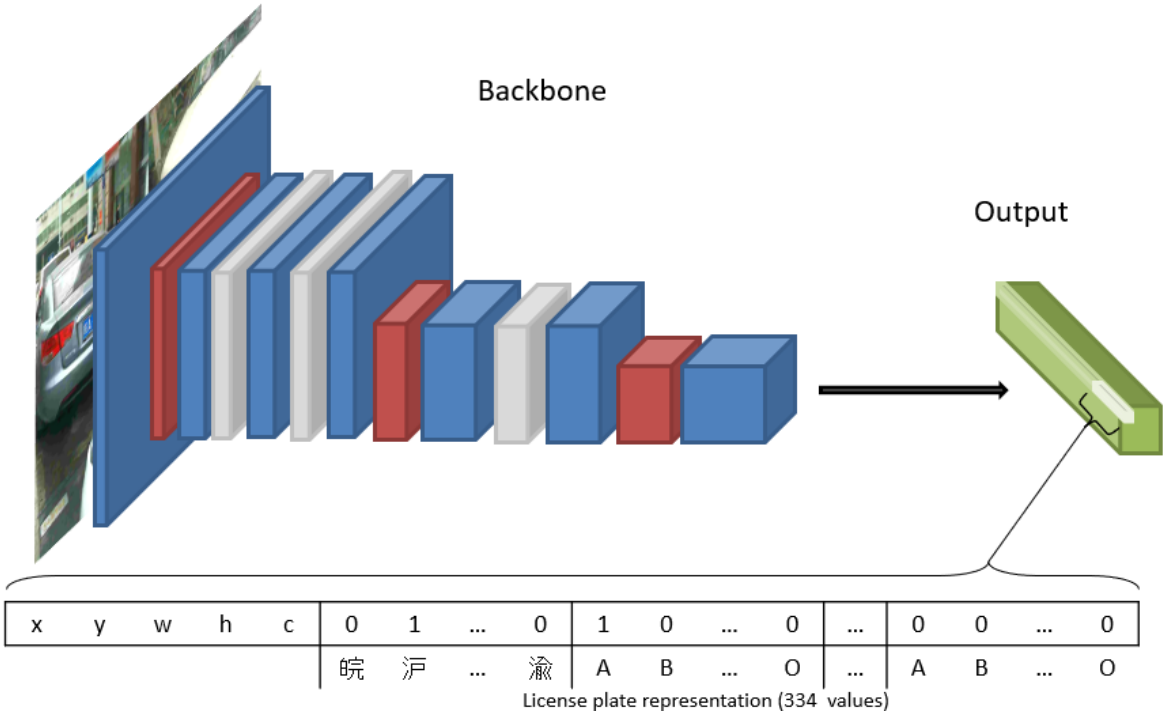

Figure 12: Model

where x is the input, y is the target, w is the weight, C_d is the number of classes of a concrete digit, P is the number of digits of the license plate and N is the number of license plates to compare. This last loss is modified from the original one, changing from a binary cross entropy loss with logits to multiple cross entropy losses, one for each digit. This cross entropy loss is the same as using a softmax and then a negative log likelihood loss. The weights of each class of each digit are inversely proportional to the probability of appearance in a scale from 0.05 to 5.

5.4 Implementation details

The backbone of the model was initialized with the weights of the pre-trained YOLOv2 PHOC of Mafla et al.[10]. They initialized it with the weights of the YOLOv2 pre-trained on Imagenet. Then it was trained using a modified version of the synthetic dataset of Gupta et al.[5]. They generated a dataset of 1 million images with background images and text rendered on top using a custom dictionary with the 90K most frequent English words. That model was trained for 30 epochs using SGD with a batch size of 64, an initial learning rate of 0.001, a momentum of 0.9, and a decay of 0.0005. the first 10 epochs was trained for only word detection and without backpropagating the loss of the PHOC prediction. The following 10 epochs they started learning the PHOC output with the λcls to 1.0. The last 10 epochs the learning rate was set to 0.0001 and the parameters λ_{box} and λ_{cls} to 5.0 and 0.015 respectively. At first the input size was fixed but during last epochs they adopted a multi-resolution training, randomly resizing the input images.

The current model starts training from these initial weights. The first 18 layers were frozen in order to maintain the feature extraction backbone already learnt previously. The hyperparameters that did not change during the training were the batch size of 32, the momentum of 0.9 and the decay of 0.0005. The scales for the loss were set as follows:

- 30 epochs: learning rate 10^{-4}, class scale = 1, coord scale = 2, object scale = 1-
- 90 epochs: learning rate 10^{-4}, class scale = 5, coord scale = 2, object scale = 1.
- 30 epochs: learning rate 10^{-3}, class scale = 5, coord scale = 2, object scale = 1,.
- 20 epochs: learning rate 10^{-3}, class scale = 5, coord scale = 2, object scale = 1. With the 8000 photos added.

While training and testing the model, a confidence threshold of 0.5 is set to to filter all boxes

with lower confidence. In testing if this condition is passed, as in real rife license plates don't overlap, a non-maximal suppression (NMS) of 0.4 is applied to remove overlapping boxes, preserving the ones with highest confidence. These boxes are the final output. To test if the bounding box is correct, the intersection over union (IOU) with the target is calculated.

6 EXPERIMENTS

The model is tested with the training set, the validation set, and the test set. On one hand the training and the validation set come form the same distribution. On the other hand the test set does not. Also, despite of not training over a lot of iterations the model with blurred, tilted, with high rotation or affected by the weather car images, as the CCPD includes images of these characteristics the model will be also tested with them, but only with images with one license. All these results can be seen in Table 1.

Training and validation set obtain the same results, so the model trained does not overfit the data and makes evident the model is learning properly. Practically, all the licenses plates are found using a threshold of 0.5 of IoU. However, not all the digits are well recognised.

Results over the custom test set are not as good as the previous ones, as expected: bounding box quality indexes decay only a couple of points, but with digits decay almost 10 points. The model is learning but it is not useful enough to predict digits well on the test set. This means, either that geometrical transformation of the Chinese license plates when casted into the European ones degrades the information, or that the domain gap between the training and the custom test set is too big

The results in the other sets of images can not be compared directly because as told before they present all only one license plate on the image, what makes it easier for the model to hit the target as the license plates are present in more pixels. Despite this, they can be compared within them. All these sets have high values of F1 score, exceeding 0.95, except the blurred one who has only 0.73, due to its bad recall. So, the model does not detect well when the image does not have enough quality.

There is no test set where the combined digits recognition is below 0.5, so neither can be it the detection and recognition together. This score is so low because a single digit wrongly recognised, leads to a failure. To do a correct examination of the results, and concretely of the failure of the digits there is the need to look when and how the model fails. For that, the confusion matrix of each digit was constructed after evaluating on the test set. This is shown in figure 17. A possible failure of the model is to confuse characters that look alike. For example, typical similarities are: (5 and S), (3, 6, and 9), (T, 7 and 1), etc.

It has been calculated the percentage of each total number of errors per prediction. It is shown below. It appears a Gaussian with the mean at 5. This reinforces what it has been said: the prediction of each digit is independently from others.

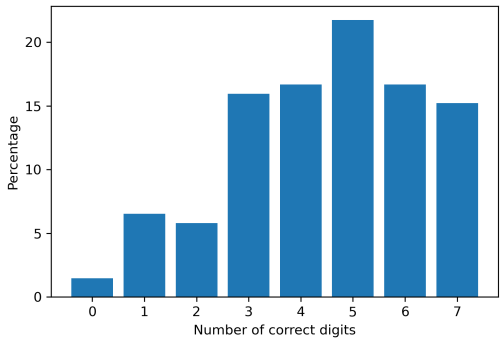

Figure 14: Correct digits per prediction

Apart from that, it has been measured the Pearson correlation between the relative width of the licenses plates and the number of digits the model succeeds on. Its value is of 0.58 and that indicates there is a connection between these two variable, depending the reliability of the model on what are the dimensions of the license plate. Also, it has been measured the Pearson correlation between the ratio of width and height with the same number of digits the model gets right. In this case, there is no relation.

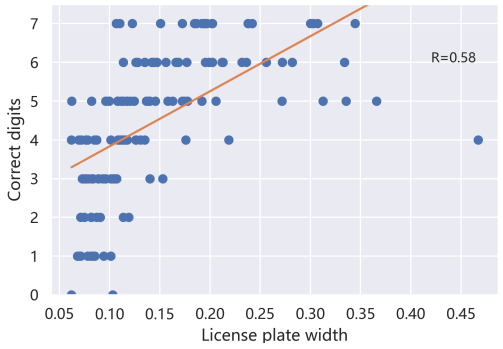

Figure 15: Correlation between width and correct digits

	Bounding box			Digits							
	Precision	Recall	F1-score	1	2	3	4	5	6	7	All digits
Training set	1.00	1.00	1.00	0.94	0.98	0.88	0.80	0.81	0.82	0.88	0.44
Validation set	1.00	1.00	1.00	0.94	0.98	0.88	0.79	0.80	0.81	0.87	0.43
Test set	0.99	0.95	0.97	0.86	0.86	0.65	0.47	0.54	0.57	0.53	0.16
Blurred set	0.99	0.58	0.73	0.85	0.85	0.51	0.48	0.47	0.51	0.51	0.04
Tilted set	0.99	0.96	0.97	0.81	0.88	0.74	0.68	0.67	0.66	0.69	0.19
High-rotated set	0.99	0.98	0.98	0.84	0.96	0.86	0.80	0.77	0.74	0.82	0.34
Rare weather set	1.00	0.99	0.99	0.94	0.97	0.93	0.86	0.85	0.82	0.89	0.50

Table 1: Results

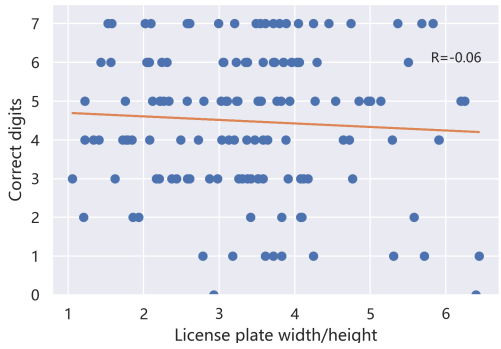

Figure 16: Correlation between scale and correct digits

Moreover, it has been calculated the percentage of failed characters that its prediction appear in another position on the ground truth of the same license plate. It is a 25% and shows that the model does not understand completely the order of the digits.

In next figure there are examples of the model failing and succeeding on recognising the license plate digits.

Figure 18: Example of output

7 CONCLUSION

Detection and recognition of multiple license plates in one image using a single-shot model, and concretely doing it with an adaptation of the YOLOv2, is possible. It was demonstrated that such a model can be successfully trained in a simple scenario, indicating that it can be reproduced with other data or motivations.

The final results in this thesis are significant but not as good as it could be, and there are a lot of factors. If this project was to be replicated, a lot of changes wold be needed. To construct the model it is used an old version of YOLO and a newer one should be tried. Despite of this, the current model is learning, so other modifications should be done.

First of all, the images for training the model need to be different and not as equal and artificial as the ones used. The best option it could be taking a lot of images of multiple cars in various environments and create our own dataset with proper annotations. There is no dataset like this and creating one is expensive in terms of money and time, so, another option is to use images from large datasets and insert license plates on them, changing the position, the scale, the rotation, the color, etc. It is also an artificial solution, but with the creation of large number of images it will probably work better. In real life, license plates appear only on vehicles, so if the model can detect license plates anywhere, they will be obviously from vehicles, and that will serve. Despite of this, as license plates are always on vehicles, the model can be less complex if it can use the contextual information of the car to locate the plate. So, in order to achieve that, it could be used a first model to segment cars from an image and then use the crops to synthesize new images.

Despite of the importance of having a good dataset to train, most part of the possible changes to achieve good results are on the training part. In order to not train the model from scratch, layers from the YOLO PHOC were frozen, so the previous knowledge could be preserved. The number of layers to freeze was chosen intuitively and to do it correctly it would be useful to do an ablation study by choosing to freeze different numbers of layers.

Another modification on the training part is the learning rate chosen. In our case an initial learning rate was chosen that was too small and not the optimum to train fast. It was changed during the training by multiplying it with a factor of ten but it was not sufficient to achieve good re-

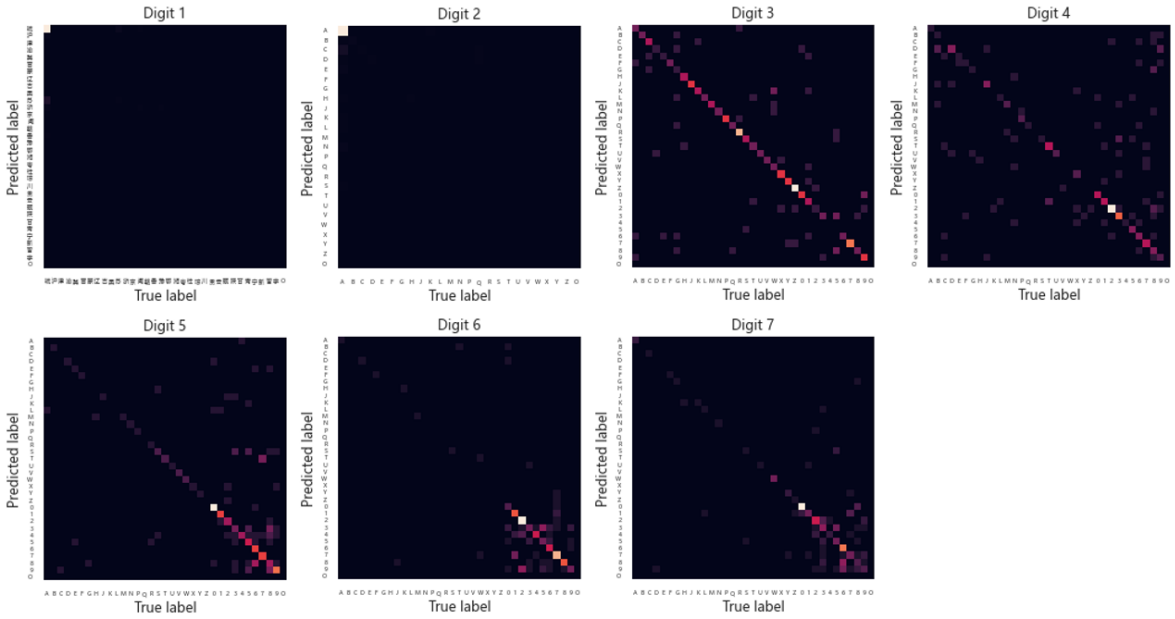

Figure 17: Confusion matrices

sults, because of a lack of time. This one is a big error of the work done here: the time planned to train the model was only of days and it was not sufficient. If the training is done in a single shot and with all the hyperparametres chosen correctly and using also multiple GPUs this is easily achievable, but if not, it is complicated to reach a model completely trained, as seen in the results, where the precision of digits could be better.

Even with all these cons, the thesis demonstrates the practical use of this model is possible and also establishes a model with its weights for applying transfer learning if a similar one needs to be created and trained more faster.

References

[1] Syed Talha Abid Ali, Abdul Hakeem Usama, Ishtiaq Rasool Khan, Muhammad Murtaza Khan, and Asif Siddiq. Mobile registration number plate recognition using artificial intelligence. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 944–948. IEEE, 2021.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[3] Lluís Gómez, Andrés Mafla, Marçal Rusinol, and Dimosthenis Karatzas. Single shot scene text retrieval. In *Proceedings of the European conference on computer vision (ECCV)*, pages 700–715, 2018.

[4] Luís Gómez, Marçal Rusinol, and Dimosthenis Karatzas. Cutting sayre's knot: reading scene text without segmentation. application to utility meters. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 97–102. IEEE, 2018.

[5] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2315–2324, 2016.

[6] Chris Henry, Sung Yoon Ahn, and Sang-Woong Lee. Multinational license plate recognition using generalized character sequence detection. *IEEE Access*, 8:35185–35199, 2020.

[7] Rayson Laroca, Evair Severo, Luiz A Zanlorensi, Luiz S Oliveira, Gabriel Resende Gonçalves, William Robson Schwartz, and David Menotti. A robust real-time automatic license plate recognition based on the yolo detector. In *2018 international joint conference on neural networks (ijcnn)*, pages 1–10. IEEE, 2018.

[8] Rayson Laroca, Luiz A Zanlorensi, Gabriel R Gonçalves, Eduardo Todt, William Robson

Schwartz, and David Menotti. An efficient and layout-independent automatic license plate recognition system based on the yolo detector. *IET Intelligent Transport Systems*, 15(4):483–503, 2021.

[9] Hui Li, Peng Wang, and Chunhua Shen. Toward end-to-end car license plate detection and recognition with deep neural networks. *IEEE Transactions on Intelligent Transportation Systems*, 20(3):1126–1136, 2018.

[10] Andrés Mafla, Ruben Tito, Sounak Dey, Lluís Gómez, Marçal Rusiñol, Ernest Valveny, and Dimosthenis Karatzas. Real-time lexicon-free scene text retrieval. *Pattern Recognition*, 110:107656, 2021.

[11] Syed Zain Masood, Guang Shu, Afshin Dehghan, and Enrique G Ortiz. License plate detection and recognition using deeply learned convolutional neural networks. *arXiv preprint arXiv:1703.07330*, 2017.

[12] 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. IEEE, 2021.

[13] J Redmon, S Divvala, R Girshick, and A Farhadi. You only look once: Unified, real-time object detection. 2016 ieee conf. comput. vis. *Pattern Recognition (CVPR)*, pages 779–788.

[14] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. corr abs/1612.08242 (2016). *arXiv preprint arXiv:1612.08242*, 2016.

[15] Hui Xu, Xiang-Dong Zhou, Zhenghao Li, Liangchen Liu, Chaojie Li, and Yu Shi. Eilpr: Toward end-to-end irregular license plate recognition based on automatic perspective alignment. *IEEE Transactions on Intelligent Transportation Systems*, 23(3):2586–2595, 2021.

[16] Zhenbo Xu, Wei Yang, Ajin Meng, Nanxue Lu, Huan Huang, Changchun Ying, and Liusheng Huang. Towards end-to-end license plate detection and recognition: A large dataset and baseline. In *Proceedings of the European conference on computer vision (ECCV)*, pages 255–271, 2018.

[17] Zili Yang. "lucky" numbers, unlucky consumers. *The Journal of Socio-Economics*, 40(5):692–699, 2011.