Reproducibility Challenge: Efficient License Plate Recognition via Holistic Position Attention

Emma Ritcey
School of Computing
Queen's University
Kingston, ON
`15er21@queensu.ca`

Maria Kantardjian
School of Computing
Queen's University
Kingston, ON
`20mk70@queensu.ca`

Ruikang Luo
School of Computing
Queen's University
Kingston, ON
`16rl46@queensu.ca`

Reproducibility Summary

Scope of Reproducibility

The authors propose a holistic position attention (HPA) license plate recognition (LPR) approach which consists of a position network and shared classifier, claiming that it achieves higher recognition accuracy and computational efficiency compared to state-of-the-art methods. The approach is end-to-end trainable, where character recognition can be performed concurrently, without the need for post-processing. The authors claim that concurrent training is important to the LPR performance and improves the accuracy and speed of the model. The second claim will not be tested, as the results for the end-to-end training are not significantly better than sequential training, and the original authors did not thoroughly prove that their concurrent training resulted in improved computation times compared to sequential training. The authors also claim that ResNet-101 implemented as a base network results in the best accuracy compared to other ResNet models, and that failure cases are due to poor semantic segmentations due to low quality license plate images.

Methodology

We re-implemented the 4-part LPR pipeline in Tensorflow, primarily by referring to the original paper's descriptions and figures of the model architecture. Part of the author's code is available on GitHub implemented in PyTorch; which we referred to when the paper did not provide enough detail in order to reproduce the architecture. The original paper implemented BiSeNet [1] for position and semantic feature extraction, we thus referred to [1] in order to implement it. A geforce gtx 1660 GPU was used for computation.

Results

Our results differed significantly from the paper's results. With ResNet-101, there was a difference of around 27% in accuracy when testing on AC, similarly when testing on LE, and 48% when testing on RP subset of the AOLP dataset. This is likely due to difference in the hyperparameter settings of the network and the framework used.

What was easy

Understanding the overall flow of the framework was easy as the paper had figures and concise explanations. The paper also describes in detail how to build the shared classifier, after having implemented the BiSeNet.

What was difficult

The first difficulty was that the original paper was not clear regarding how BiSeNet was implemented into the pipeline as multiple parts of BiSeNet were renamed in the LPR pipeline. Accessing the proper datasets was also difficult, as minimal details were provided in the paper regarding how the datasets were acquired.

Communication with original authors

We contacted the original authors via email, however, did not receive a response.

1 Introduction

License plate recognition (LPR) is one of the most important aspects of intelligent transportation systems since license plates uniquely identify vehicles [2]. To perform LPR, the semantic category and position information of each character should be extracted. Traditional methods include two steps performed separately; first detecting all the characters in the license plate and then cropping character subimages to perform the recognition. More recent methods attempt to perform these two steps in a parallel manner using techniques that involve LSTM, multiple classifiers, and semantic segmentation. The downfall of these attempts include noisy features, poor run-times, difficulty extracting position information of characters, and the need for post-processing.

Zhang et al propose a novel LPR pipeline which achieves superior accuracy and speed compared to previous state-of-the art methods. The proposed pipeline implements holistic position attention to encode the position information of each character in the license plate. Then, a shared classifier is used specifically for character recognition. The methods allow all characters in a license plate to be processed in parallel and eliminates the need for post-processing.

2 Scope of reproducibility

The original work proposes a novel LPR method which utilizes HPA to encode the position information of the license plate characters and a shared classifier to perform character recognition. The authors make many claims in the original study, three of which will be tested in this reproducibility study:

- The HPA approach achieves higher recognition accuracy compared to previous state-of-the-art methods.
- The ResNet-101 based HPA model achieves the best recognition accuracy compared to ResNet-50, ResNet-34, and ResNet-18 when tested on the AOLP, Media Lab, CCPD-base, and CLPD datasets.
- Failure cases are not caused by faults in the network, but are caused by poor semantic segmentations due to low quality license plate images.

These claims will be tested using the methodology in Section 3 and the reproduced results will be reported in Section 4.

3 Methodology

We aimed to re-implement the author's approach from the descriptions in their paper. When this failed, as not enough detail was provided, the author's code published on GitHub, which was implemented in PyTorch was used as a resource. The method also implemented BiSeNet as a large part of the model architecture, and thus we referred to [1] in order to implement it correctly as the original paper lacked a clear description. All experiments were performed on a geforce gtx 1660 GPU.

3.1 Model descriptions

The model is a 4-part pipeline consisting of a backbone network, a semantic network, a position network, and a shared classifier. The backbone network transforms the input image into global features. A ResNet model, pretrained on the ImageNet dataset [cite] was used for this purpose. ResNet-18, -34, -50, and -101 models were all tested as the backbone network, to compare if ResNet-101 had superior performance, which the original paper reported. The semantic network and position network separately extract the semantic and position information of each character and produce semantic features and HPA maps. The shared classifier then takes the semantic features and HPA maps and performs character recognition on the entire character sequence. The first three parts of the pipeline are implemented through a BiSeNet architecture, which consists of spatial and context paths and a feature fusion module. The spatial and context paths are part of the backbone network and are used for common feature extraction. The feature fusion module (FFM) is called upon twice, once for the position network and again for the semantic network to produce the position and semantic features, respectively.

Backbone Network: Spatial Path

The model begins by sending the image into the spatial path. The spatial path consists of three rounds of convolution, batch normalization, and a ReLU activation function. Convolution was performed with a (3x3) kernel, a stride of 2, and padding to preserve the image dimensions. The first round used 64 layers, the second used 128, and the third used 256. The output of the spatial path is then utilized in the position and semantic networks.

Backbone Network: Context Path

The context path consists of a ResNet model pretrained on the ImageNet dataset. The final two layers of the ResNet model, layerA and layerB, are fed into attention refinement modules (ARM). Each ARM consists of global average pooling, 1x1 convolution, batch normalization, and a sigmoid activation function. To produce the ARM output, the output of these 4 layers is multiplied by the original ARM input. The ARM outputs of layerA and layerB, called layerA2 and layerB2, are then upsampled to be of size 28x28. LayerB2 is then multiplied by the tail of the ResNet model, and this result is concatenated with LayerA2 to produce the context path output. This result is used for both the position network and semantic network.

Position Network: FFM

The position network is the FFM of BiSeNet. The output of the spatial and context paths is fed into the FFM and concatenated. Then 3x3 convolution with a stride of 1, batch normalization, and ReLU are performed consecutively to produce output1. Global average pooling is applied next to output1, followed by another 1x1 convolution with a stride of 1, ReLU activation, one more 1x1 convolution with a stride of 1, and sigmoid activation to produce output2. Next, output1 is multiplied by output2, and this product is added to output1 to produce the final output. For each convolution layer in the FFM, the number of filters is equal to the number of classes. The position network aims to extract position features from the image. Typically, a license plate contains 7 character positions, therefore, the number of classes and number of filters used in the position network FFM is 7. To produce the HPA maps, a softmax activation is applied to the final output.

Semantic Network: FFM

The semantic network is also implemented as a FFM. To produce the semantic network final output, the same steps are used as explained above for the position network, however the number of classes is now equal to the number of different characters that could exist in a license plate. There are 26 possible letters and 10 possible characters, therefore the number of filters used in the semantic network FFM is 36. Instead of applying a softmax, batch normalization is performed on the final output to produce the semantic features.

Shared Classifier

In the final section of the pipeline, the HPA map of each character is used to modulate the semantic features separately. The shared classifier is then used for character recognition. It consists of a 5x5 convolution layer, batch normalization, and ReLU activation function to process the features. Global pooling and a fully connected layer are then applied to perform the classification of each character.

3.2 Datasets

AOLP [3]: this dataset consists of 1874 images of Taiwan license plates. The dataset is divided into three subsets: access control (AC) with 681 images, law enforcement (LE) with 582 images, and road patrol (RP) with 611 images. Each subset of the data has the license plate images, the coordinates of the upper left and bottom right corners of the plate in each corresponding image, and the characters of the plates for each corresponding image. The dataset is divided into 70% training and 30% testing. It can be downloaded by reaching out to Prof. Hsu's e-mail: jison@mail.ntust.edu.tw and obtaining a password.

Due to time constraints, the re-implementation of the original work was only tested on the AOLP dataset. In the original paper, experiments were run on three other datasets:

Media Lab: this dataset consists of 706 images of Greek license plate.

CCPD: this dataset consists of 290,000 images of Chinese license plates.

CLPD: this dataset consists of 1200 images of Chinese license plates.

3.3 Hyperparameters

The hyperparameter values used for the final experiments are shown in Table 1. These values were found through our own testing, as the original paper did not provide details of the hyperparameters which produced their best results.

Table 1: Hyperparameters used for final experiments

Hyperparameter	**Value**
epochs	200
batch size	24
learning rate	1e-3
optimizer	Adam
loss	sparse categorical crossentropy

3.4 Experimental setup and code

Experiments were run using ResNet-18, ResNet-34, ResNet-50, and ResNet-101 as the base model. Following the same strategy as the paper, two subsets of the AOLP dataset were used to train the model. Then the model was tested on the other remaining AOLP subset. This process was performed three times, such that each AOLP subset was used for testing once for each ResNet model. Therefore, for each ResNet model, three experiments were performed:

- Trained on AC and RP and tested on LE.
- Trained on AC and LE and tested on RP.
- Trained on RP and LE and tested on AC.

The character recognition accuracy, which is calculated as (number of correct character predictions) / (total number of characters), was used to evaluate the performance of each model.

Our current code can be found at `https://github.com/emmaritcey/CISC867Project`.

3.5 Computational requirements

Experiments can be run on either a standard CPU or a GPU in order to decrease computation times. Our experiments were performed using both a CPU and a geforce gtx 1660 GPU. On a CPU, training required approximately 6 hours. On the GPU, training time was decreased to approximately 1 hour. It is recommended to use a strong GPU for the experiments, so that testing and hyperparameter tuning can be performed more efficiently.

4 Results

The results for the experiments are summarized in Table 2. The first claim, which stated that HPA achieves higher recognition accuracy compared to previous state-of-the-art methods was not supported by our re-implementation. Our reproduced results had lower accuracy than the state-of-the-arts methods mentioned in the paper for each of the experiments performed. The highest accuracy our methods produced was 73.35%, which is far lower than 85.70%, which is the lowest state-of-the-art accuracy mentioned in the paper.

The second claim, which stated that ResNet-101 as the base model network results in the best accuracy as compared to the other models was supported throughout the reproduced results. Our ResNet-101 model produced accuracies of 72.72%, 73.35%, and 52.40%, for an average of 66.16%. Although this is not a desirable accuracy, it out performed ResNet-50, ResNet-34, and ResNet-18 which achieved accuracies of 62.78%, 10.45%, 53.86%, respectively.

The claim which states that failure cases are not caused by faults in the network, but are caused by poor semantic segmentations due to low quality license plate images cannot be supported by this re-implementation. The accuracy across all methods is far too low to claim this fact. What can be seen is that for our results of ResNet-101 and ResNet-50, training on the subsets AC and LE while testing on the RP subset results in a lower test accuracy, especially for ResNet-101. This can likely be attributed to the fact that RP contains many skewed or partially cut off license plates, shown in Figure 1.

Table 2: Character Recognition Accuracy for previous state of the art LPR methods, Zhang et al's method, and our reproduction of Zhang et al's method. AC/LE/RP denotes the subset used as the test set.

Method	**AC**	**LE**	**RP**
(Hsu, Chen, and Chung 2012)	88.50	86.60	85.70
(Li and Shen 2016)	84.85	94.19	88.38
(Wu and Li 2016)	97.90	97.60	98.20
(Silva and Jung 2018)	-	-	98.36
(Zhuang et al. 2018)	99.41	99.31	99.02
SNIDER(Lee et al. 2019)	-	-	99.18
(Zhang et al. 2019)	-	-	99.67
(Zhang et al. 2020)	97.30	98.30	91.90
OpenALPR	92.36	86.08	93.13
Sighthound	94.57	96.56	89.03
Zhang et al. ResNet-18	98.83	97.74	98.20
Ours ResNet-18	41.17	72.79	47.61
Zhang et al. ResNet-34	98.79	98.97	98.37
Ours ResNet-34	8.60	12.51	10.23
Zhang et al. ResNet-50	99.12	98.97	98.53
Ours ResNet-50	68.93	64.92	59.78
Zhang et al. ResNet-101	99.56	100.00	99.84
Ours ResNet-101	72.72	73.35	52.40

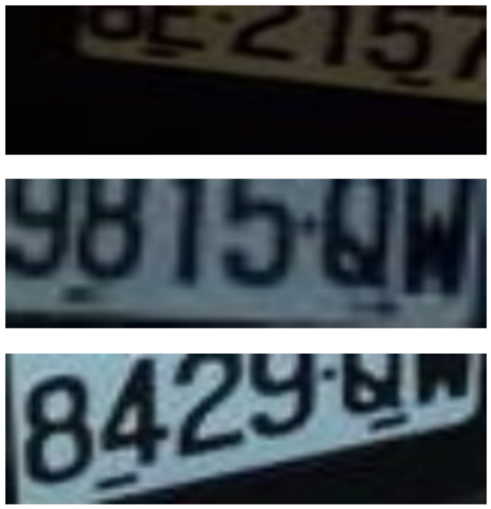

Figure 1: Three examples of poor quality license plate images from the RP dataset

5 Discussion

The results obtained did not have the same performance as the original paper. Even though the best performance was obtained by ResNet-101 which supports the second claim, the performance on all three subsets of the data used for testing was still considerably worse than the one in the paper. For the first experiment, using LE and RP subsets for training and AC for testing, the original paper had an accuracy of 99.56%, whereas the reproduction resulted in 72.72%. For the second experiment, training using AC and RP and testing on LE, the original paper resulted in 100% accuracy, whereas our reproduction resulted in an accuracy of 73.35%. As for the third experiment, using AC and LE for training and RP for testing, the original paper had an accuracy of 99.84%, and the reproduction resulted in 52.4%. This difference in the results is likely partly due to different hyperparameter settings. The paper did not mention what hyperparameters were used to train the network. Additionally, the original paper was implemented in PyTorch while our work was reproduced using TensorFlow, and different frameworks may have different default values for certain hyperparameters.

Another factor which likely influenced the decrease in performance is preprocessing. The original paper did not explain how the AOLP dataset was preprocessed. This meant we had to use our own preprocessing methods, with very little experience in LPR preprocessing. For images with license plates parallel to the image frame, our preprocessing methods worked fairly well, however with skewed images, this resulted in portions of license plates being cutoff. This definitely reduced the ability of our model to correctly produce characters. With improved preprocessing methods, our model likely would have performed better, although it is unknown whether it would be enough to improve our results to the level of accuracy the original paper was able to reach.

As previously stated, the third claim is not supported by this re-implementation. Although poor image quality played a role in misclassifications, specifically in the RP dataset, there are many other factors that contributed to the poor performance, such as poor preprocessing and potentially non-optimal hyperparameters. However, the decreased performance when testing on the RP dataset, which is most prevalent with ResNet-101, can likely be caused by the many skewed or partially cut off license plates in this dataset. Testing on a much more difficult dataset means that the model has not been trained to recognize these more difficult and skewed images. Thus, a portion of failure cases when testing on the RP subset could be due to low quality license plate images.

5.1 What was easy

The paper had many figures and clear explanations, which lead to a clear understanding of the overall flow of the framework of the model architecture. The paper also describes in detail how to build the shared classifier layer-by-layer, after integrating BiSeNet into the model as a whole. Obtaining access to the AOLP dataset was also easier than expected, as access was granted upon request.

5.2 What was difficult

We faced many difficulties while reproducing this paper. The original paper was not clear regarding how BiSeNet was implemented into the pipeline. Originally, it seemed as though BiSeNet was just the backbone network of the pipeline, however, once we dove deeper into the BiSeNet implementation, we realized that this structure was used for both the semantic network and position network. Accessing the datasets, aside from AOLP, was also difficult, as there are many different versions available and there was minimal details provided in the paper regarding how the datasets were acquired. Ultimately, these other datasets were not used for testing so this did not affect the final results.

The lack of preprocessing details also introduced many difficulties while trying to reproduce the results. Without any details as to how the images were preprocessed, we were forced to naively come up with our own preprocessing pipeline. This worked well for clean images, but for skewed images it resulted in parts of the license plates and therefore characters being cropped out of the image. With more LPR preprocessing knowledge, or better yet more preprocessing details in the original paper, this could have easily been avoided.

Lastly, the authors did not provide hyperparameter values for their final results, which made it much more difficult for us to tune our model to achieve similar results. Further tuning of hyperparameters, or the exact values of the hyperparameters from the original paper, would likely improve our results.

5.3 Communication with original authors

We reached out to the original authors via email, however we did not hear back. Our email was sent with the intention to gain information surrounding the hyperparameters used in their experiments as well as how the images were preprocessed prior to being processed by model.

References

[1] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang, "Bisenet: Bilateral segmentation network for real-time semantic segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018, pp. 325–341.

[2] Y. Zhang, Z. Wang, and J. Zhuang, "Efficient license plate recognition via holistic position attention," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 4, pp. 3438–3446, May 2021.

[3] G.-S. Hsu, J.-C. Chen, and Y.-Z. Chung, "Application-oriented license plate recognition," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 2, pp. 552–561, February 2013.