aurum

AURUM MÜHENDİSLİK SİSTEMLERİ VE MİMARLIK DERGİSİ
AURUM JOURNAL OF ENGINEERING SYSTEMS AND ARCHITECTURE

Cilt 5, Sayı 1 | Yaz 2021
Volume 5, No 1 | Summer 2021, 11-28

ARAŞTIRMA MAKALESİ / RESEARCH ARTICLE

VEHICLE POSITION ESTIMATION AND VEHICLE CLASSIFICATION USING DEEP CONVOLUTIONAL NEURAL NETWORKS

Bashaer Isam Hasan KABEAYLA[1]
[1]Information Technology, Altinbas University, Istanbul, Turkey
radbash6@gmail.com, ORCID No: 0000-0002-6698-5653

Yasa Ekşioğlu ÖZOK[2]
[2]Electronic and Computer Engineering, Altinbas University, Istanbul, Turkey
yasa.eksioglu@altinbas.edu.tr, ORCID No: 0000-0003-2406-1310

Geliş Tarihi/Received Date: 04.03.2021 Kabul Tarihi/Accepted Date: 10.06.2021

Abstract

The aim of this paper is to classify the vehicles and estimate the position with license plate localization using deep convolutional Neural Network (DCNN). Vehicle pose estimation with license plate localization serves as one of the most widely used real-world applications in fields like toll control, traffic scene analysis, and suspected vehicle tracking. We proposed a one-stage anchor-free vehicle classifier for simultaneously localizing the region of license plates and vehicles' poses. The classifier, rather than bounding rectangles, gives bounding quadrilaterals, which gives a more precise indication for vehicle pose estimation with license plates localization. For single scale input, we reached mean Precision Accuracy mAP/mAP50 of 35.4/82.3 on the LISA benchmark dataset, already outperformed the existing commercial systems OpenALPR and Sighthound. For multi-scale input, we reached the best mAP/mAP50 of 40.8/90.1. For the vehicle pose (front-rear), classification accuracy reached 98.8%, average IoU reached 71.3%, giving a promising result as an end-to-end vehicle position estimation and license plate localization with contextual information. The work has performed in python programming language with several libraries of deep learning were being used for this purpose. Our DCNN model training started from an initial weight which we had already trained for about 110000 iterations in the model without classification head, so the total training iterations will be around 780000 including the transfer learning part in DCNN. Transfer learning made the DCNN model start at a smart point and made it easier to optimize all of the functional heads simultaneously.

Keywords**:** Vehicle classification, pose estimation, optimization, DCNN, transfer learning, license plate, localization, deep learning

DERİN EVRİŞİMLİ SİNİR AĞLARINI KULLANARAK ARAÇ KONUMU TAHMİNİ VE ARAÇ SINIFLANDIRMASI

Özet

Bu çalışmanın amacı, araçları sınıflandırmak ve derin evrişimli sinir ağı (DCNN) kullanarak araçların plaka lokalizasyonunu tahmin etmektir. Plaka lokalizasyonu ile araç poz tahmini geçiş ücreti kontrolü, trafik sahnesi analizi ve şüpheli araç takibi gibi alanlarda en yaygın kullanılan gerçek dünya uygulamalarından biri olarak hizmet etmektedir. Plakaları ve araçların pozlarını eşzamanlı olarak saptamak için tek aşamalı ankrajsız bir

araç sınıflandırıcı önermekteyiz. Sınıflandırıcı, dikdörtgenleri sınırlamak yerine sınırlayıcı dörtgenler vererek araç plakalarının lokalizasyonu ile araç poz tahmini için daha hassas bir işaret vermektedir. Tek ölçekli girdi için, LISA kıyaslama veri kümesinde hassas doğruluk için mAP / mAP50 35.4/82.3 değerine ulaştık, halihazırda mevcut ticari OpenALPR ve Sighthound sistemlerinden daha iyi performans göstermiştir. Çok ölçekli girdiler için, 40.8 / 90.1'lik en iyi mAP / mAP50'ye ulaştık. Araç duruşu için (ön-arka), sınıflandırma doğruluğu%98,8'e, ortalama IoU %71,3'e ulaşarak uçtan-uca araç konumu tahmini ve bağlamsal bilgilerle plaka lokalizasyonu açısından umut verici bir sonuç vermektedir.

Çalışma, python programlama dilinde gerçekleştirilmiştir bu amaçla çeşitli derin öğrenme kütüphaneleri kullanılmıştır. DCNN model çalışmamız sınıflandırma başlığı olmadan modelde yaklaşık 110000 iterasyon için önceden çalıştığımız bir ilk ağırlıktan başlar, bu nedenle DCNN'deki aktarım çalışma bölümü dahil olmak üzere toplam çalışma iterasyonu yaklaşık 780000 olacaktır. Transfer öğrenimi, DCNN modelinin akıllı bir noktadan başlamasını sağlar ve tüm işlevsel başları aynı anda optimize etmeyi kolaylaştırır.

***Anahtar Kelimeler*:** Araç sınıflandırma, poz tahmini, optimizasyon, DCNN, öğrenme transferi, araç plakası, lokalizasyon, derin öğrenme

1. INTRODUCTION

The broader goal of this paper is to investigate how neural systems can stay robust to incomplete information for vehicle classification and position estimation with license plate localization. Humans are naturally gifted at extracting knowledge and taking decisions based on imperfect observations about the state of their environment as described in (S. Du, M. Ibrahim, 2013). The evolutionary advantage of developing this ability is hardly questionable, however it remains to be clearly understood how exactly our brains can achieve this. This downsized connectivity and the weight sharing characteristic of convolutional networks produce a reduction in the size of the weight search space, which simplifies the network optimization procedure. Besides of being of great interest to improve our understanding of learning and information processing in neurobiology, this question will awaken interest in any engineer attempting to design more robust intelligent systems. In the context of this research paper, the focus will be kept on the second perspective by concentrating on the study of artificial neural networks as provided by (Sayanan Sivaraman, 2010) . The specific problem of interest will be the recognition of vehicles in occluded images. This is a strategic choice for research on this topic, because classical vehicle recognition is a well-researched territory, both in the brain as well as in artificial systems as mentioned in (S. Silva, 2017). This is why it will be useful to review the current state of the field and introduce important theoretical notions before expanding on the core of the paper.

1.1 Problem Statement

In the vehicle pose estimation stage, most existing systems are based on the frontal view of vehicle. The open-source version of commercial software struggles to find the vehicle pose estimation and license plate with transformation, like a tilted license plate or the license plate which belongs to the car in an oblique view. This problem states a critical performance bottleneck for machine learning based systems

since the vehicle position estimation and detection failure means the total inability to comprehend a specific position. Here, we conclude the problems of vehicle classification and pose estimation with license plate localization that's would be solved:

1. Lack of estimation ability for tilted and oblique image of vehicles in real time.
2. Loss of contextual information while dealing with more general vehicle classification and position estimation.
3. Is it possible to generate training data by using a vehicle image and a measured ground truth of average vehicle classification?
4. Does the approach of deep learning techniques as a way of automating vehicle classification and position estimation measurements with the help of real time images?
5. What aspects need to be considered if further work on the subject is to be performed?

1.2 Research Contribution

There are four sub-tasks in the vehicle classification and vehicle pose estimation with license plate localization process, our research focuses on the vehicle pose estimation with license plate localization after vehicle classification task, aiming to accurately locate the vehicle position in various scenes with single or multiple vehicles inside. As to maintain the contextual information when dealing with higher vehicle classification performance, the license plate localization will be detected as well, the pose of the car, on which side (front or rear) does the license plate lay on will also be classified. We will show a detection example, the vehicle poses with license plate localization missing problem mentioned in the problem statement can be solved by our proposed method. We conclude our main contributions as followed:

- We designed a DCNN-based network for vehicle classification and pose estimation vertices with a variety of angles in real-world scenes.
- Along with vehicle pose estimation and the license plate localization, vehicle's pose information will also be given.
- Our model will detect and classify the region of the car's front part and the rear part where a vehicle position is onto.
- To train our model, we proposed a new dataset with manually annotated front-rear bounding boxes.
- The classification process is based on a novel anchor-free method presented in methodology, no manually decided anchor-box size is needed; it might be an inspiration for other vehicle classification and detecting applications.

2. RELATED WORK

A popular benchmark for vehicle detection is the Pascal Visual Vehicle Class (VVC) challenge, the version released in 2012 (M. Everingham, 2012) includes 11,530 images with 20 classes and 27,450 bounding box annotations in the training and validation dataset. Figure 1 shows the mAP (mean Average Precision) increasing trend for the Pascal VOC challenge, the first three methods are based on traditional visual descriptors like Histogram of Gradient (HOG) used in SVM-HOG and Scale-Invariant Feature Transform (SIFT) used in DPM-MKL, lastly the CornerNet is a key method that finds the top-left and bottom-right points of a vehicle and further performs bounding box refining.

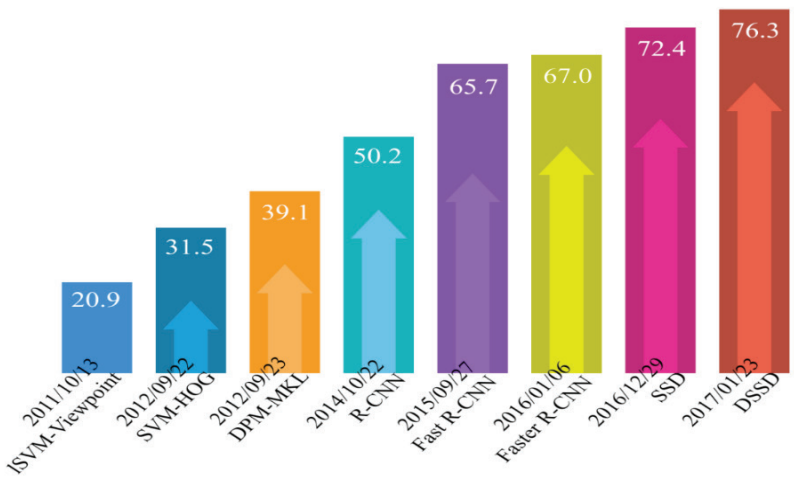

Figure 1: The trend chart of mAP statistics is drawn to represent the trends

Researchers in (Available Online) proposed an anchor-free method (region proposal without anchor boxes), which can be applied to one-stage classifier, with the combination of anchor-based method and anchor-free method, they achieved the state-of-the-art vehicle detection results. Table 1 lists the components in the recent vehicle classifier.

Table 1. Components of Vehicle Pose Estimation with License Plate Localization

Vehicle Pose Estimation with License Plate Localization			
Classification Paradigms	Feature Extraction	Proposal Generation	Backbone Architecture
One-Stage	Image Pyramid	Anchor-Based	VGG16
Two-Stage	Detection Pyramid Integrated Features	Anchor-Free	ResNet
	Feature Pyramid		Hourglass Net

After 2013, deep learning-based detection methods almost dominate the Pascal VOC challenge, including Region-based Convolutional Neural Network (R-CNN) (R. Girshick, 2014) and its extensions Fast R-CNN (R. Girshick, 2015) and Faster R-CNN (S. Ren, 2015), SSD (W. Liu, 2016) and its extension DSSD (C.-Y. Fu, 2017) , after 2017, the newly released state-of-the-art classifier are still improving their detecting ability but tend to shift to another more challenging benchmark MS COCO (T. Lin, 2014) which has 80 categories. The deep learning-based vehicle classifier broke through the limitation of traditional classifier and made a tremendous improvement in mAP scores. This section will introduce several famous vehicle classifier and related algorithms.

Stacked Hourglass Network (A. Newell, 2016) was first designed for human pose estimation, it is a feature extraction backbone with several Hourglass Networks connected together. A single Hourglass Network is shown in Figure 2, the architecture of Hourglass Network is a sequence of down-sampling DCNN followed by a sequence of up-sampling DCNN. To utilize the feature with rich spatial information (shallow part of DCNN) and feature with rich semantic information (deeper part of DCNN), there're lateral connections that merge the early features with latter features, it's the central concept of Hourglass Network and make it suitable for vehicle detection tasks since the spatial information often disappears in the latter part of a simple DCNN architecture.

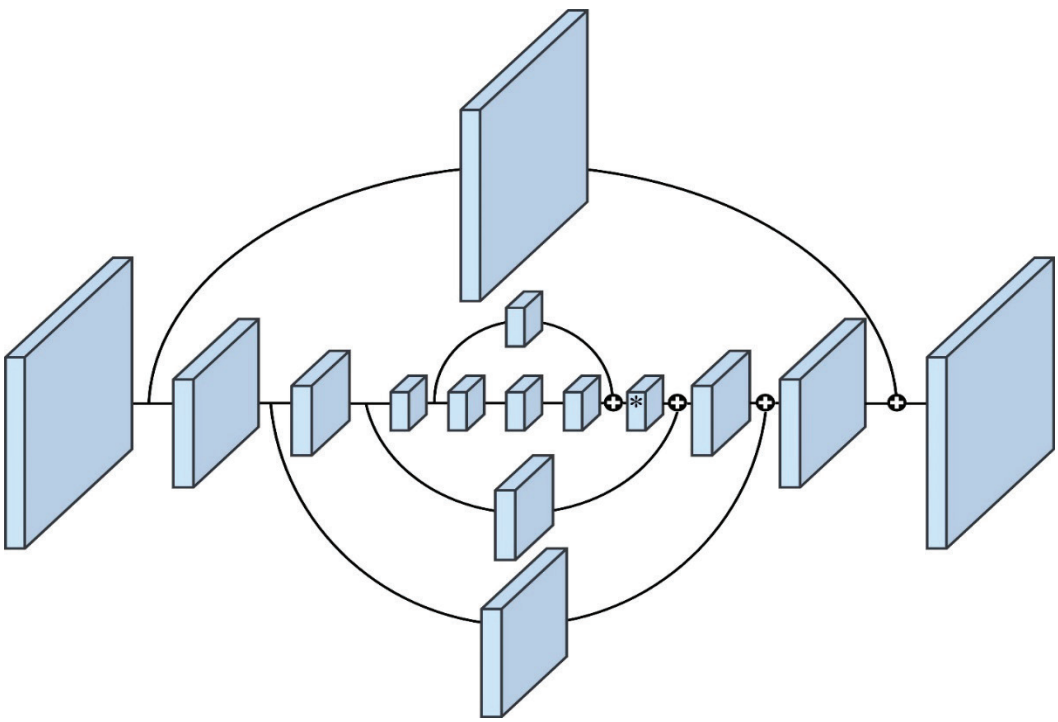

Figure 2. Hourglass Network. The block with * mark refers to simple feature addition, other blocks refer to residual blocks

Each block (except the block with * mark) in Figure 2 refers to a residual block. Residual block was first proposed by (K. He, 2016) , a skip connection from input layer directly to output layer was added, it was designed to address the problem of gradient vanishing problem when training a deep neural network, when the network goes deeper, information from shallower layer might disappear due to the gradient-based back-propagation, residual block makes it possible to train a network as deep we want as mentioned by (2018). The channel amount (kernel amount) is basically 256 in the entire Hourglass Network, as we can see in the input and output of the residual block. Inside the residual block, there are two DCNNs with 128 channels with kernel sizes 1 and 3, followed by a DCNN with 256 channels with kernel size 1. Stacked Hourglass Network was used for feature extraction backbone network in recent vehicle detection researches like CornerNet and CenterNet.

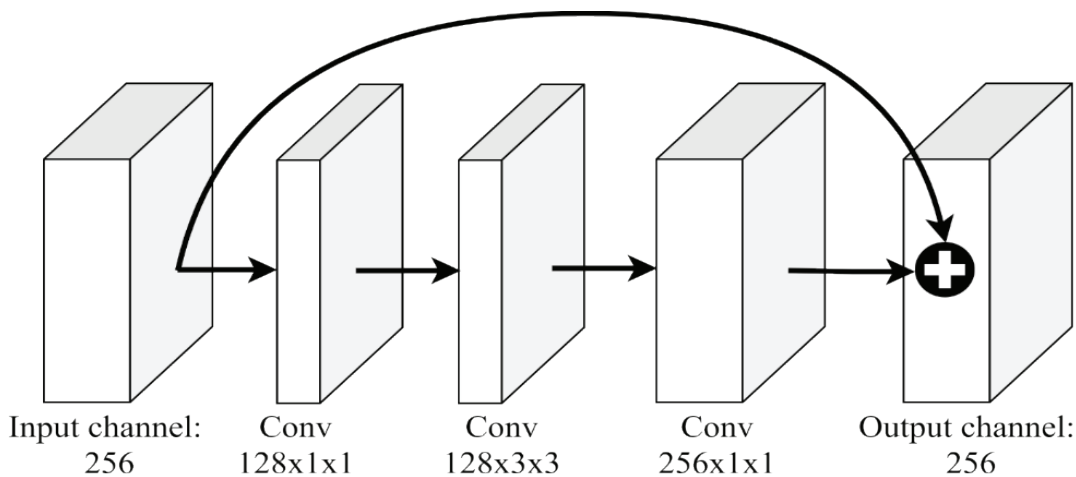

Figure 3. Residual block used in Hourglass Network

3. METHODOLOGY

The proposed model is called DCNN (Deep Convolutional Neural Network). DCNN is a one-stage and anchor-free deep learning classifier. The whole model is shown in Figure 4. The width and height of an input RGB image will be first downscaled to 1/4 and pass through two stacks of Hourglass Network, until here is the backbone feature extraction part as described by (T.-Y. Lin, 2017). By utilizing the obtained features, three parallel heads will then handle localization, region regression, and classification individually. The right side of Figure 4 gives an output example of our model, we can see that the license plate region has been found, and the front-rear region of the owner car is also given along with its pose information.

This section includes the backbone feature extraction network design, and the head network architecture with its function and design inspiration, which is done in one-stage manner. The last part of this section will be a brief discussion for anchor-free method.

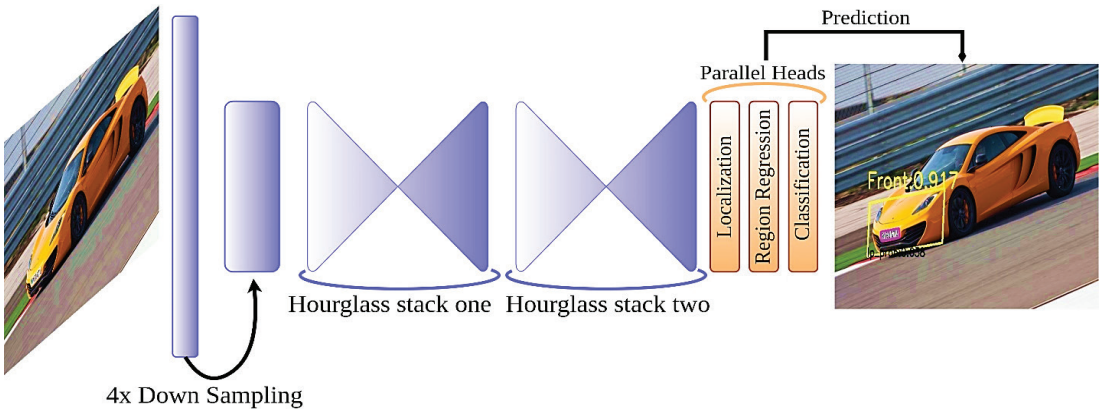

Figure 4. The whole pipeline of our model

3.1 Backbone Network Design

As discussed in section 2.1.1, a one-stage classifier has a trade-off between speed and performance, and the performance will descend significantly especially for small vehicles. License plates, in general situations, are small vehicles (imaging the scale difference between license plates and vehicles), so the feature extraction ability of the backbone network architecture becomes considerable, the spatial information needs to be fruitful to avoid missing detections for small vehicles. Inspired by the adoption of Hourglass Network in recent one-stage detectors (H. Law, 2018), we introduced Hourglass Network as our backbone architecture and found it performed much better than a normal straight forward DCNN, a comparison between their performance.

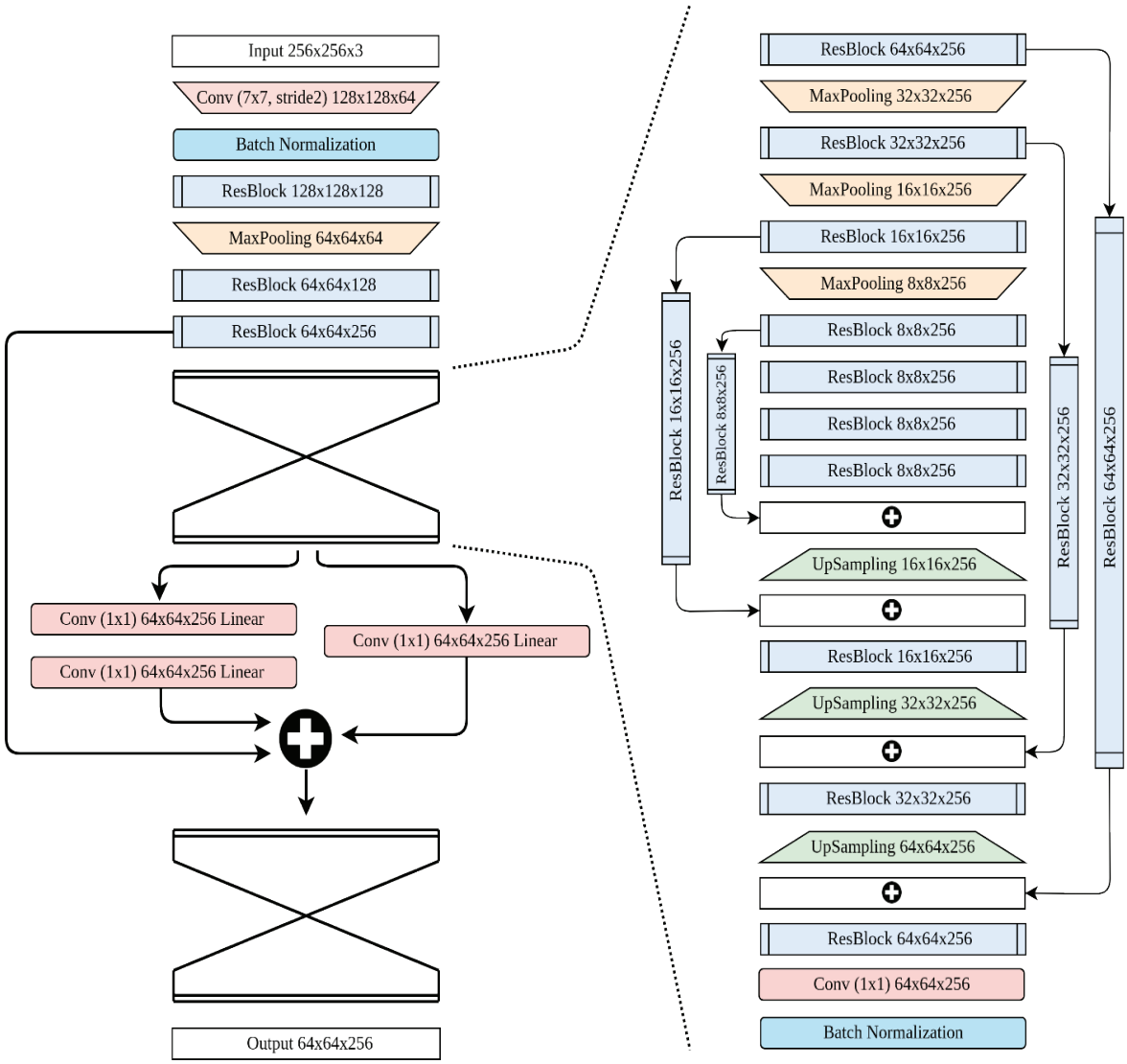

Figure 5. Backbone network. A two-stack Hourglass Network

3.2 Head Network Design

Beyond the backbone network, we designed four parallel DCNNs as our head networks, each of them handles different tasks, Figure 6 gives a clear comprehension of the architecture, and we will illustrate each of the head networks.

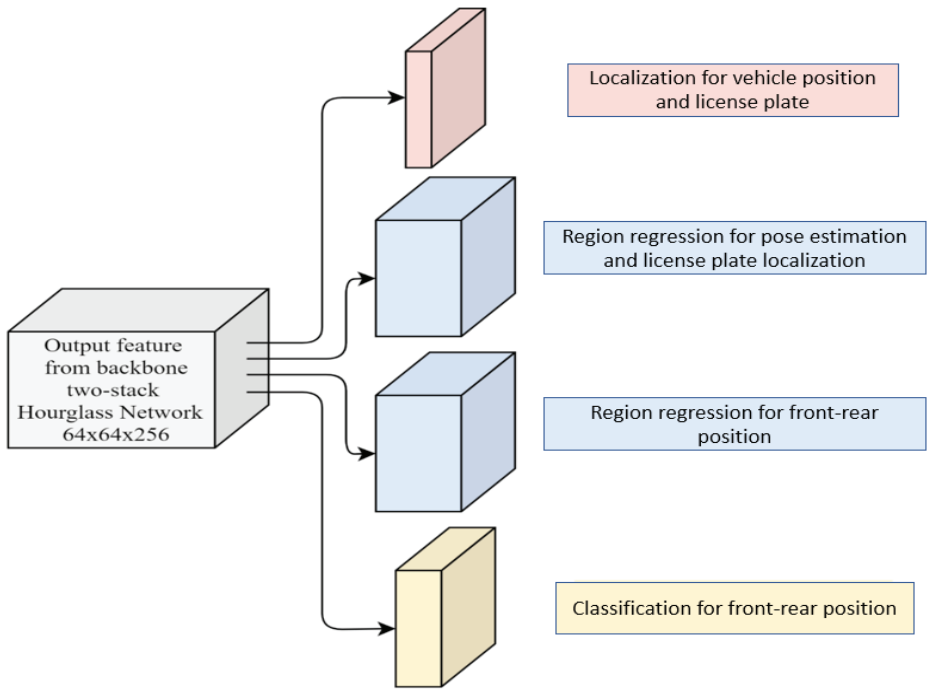

Figure 6. Head network, there are four parallel DCNN branches, the top localization branch, the middle two region regression branches, and the final classification branch

1. **Localization for vehicle position and license plate:** The localization head is done by a deep convolutional neural network with filter size 3x3, the output will be a feature map of size 64x64x1, for each pixel, there is only one output channel which indicates the probability of containing a license plate. Since we did not include the background class in our design, it serves as a single-class problem; the sigmoid activation function fits our requirement and became our choice for activation function.

2. **Region regression for pose estimation and license plate localization:** The region regression part is done by a convolutional neural network with filter size 3x3, and the output feature map has a size of 64x64x8, there are eight output channels for each pixel, the output channels handle the region proposal for vehicle pose estimation and license plates localization. Figure 7 explains how these values work with region proposal, first, we will have four pairs of unit vectors in the same arrangement of four quadrants, r_x and r_y then serve as scalars to expand these unit vectors, after expanding a pair of unit vectors, do the vector addition of the horizontal vector and vertical vector in the pair to obtain a destination point. After performing the same procedure on the four pairs of unit vectors, we will obtain four points, and these points will then be the vertices of a quadrilateral, which indicate the region of a license plate. Since we need the expanding factor for unit vectors, exact scalars must be obtained, so we used linear activation function (equivalent to no activation function) for the DCNN layer.

3. **Region regression for front-rear:** The region regression for the car's pose (front-rear) uses exactly the same method used in region regression for license plate, but the vertices now indicate the front-rear region of a car.

4. **Classification for front-rear:** As a multi-class classification task, there are three classes here, front class, rear class, and background class. The classification process is done by a convolutional neural network with filter size 3x3, and the output feature map has a size of 64x64x3, each channel in each pixel represents the probability for front class, rear class, or background class, respectively. Here we used softmax activation function for its compatibility for multi-class classification.

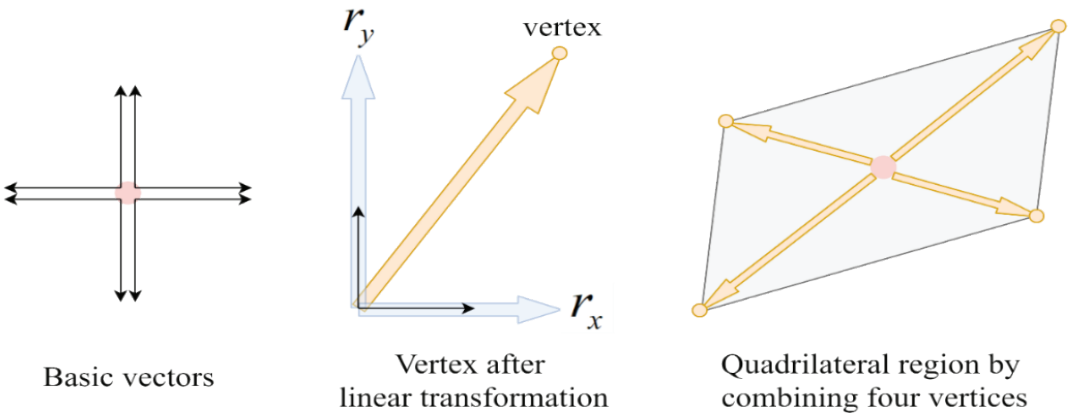

Figure 7. Region regression method

3.3 Training Details

The purpose of this section is to describe the whole construction of our training process, including the online data augmentation method, the transfer learning method used for avoiding unstable training state, and the adjustment of the hyperparameters for different periods of iterations. We used LISA dataset for training and testing. Before the training images were fed into the model, several augmentation methods were taken randomly. These methods are listed in Table 1, the Chance column in the table with value 50% means the augmentation was not taken for every time, but only half of the chance. Each image performed each of the methods for one time with random order. Online data augmentation preserves the diversity of training data. In every augmentation method, we also used random parameters, meaning that after performing all of the methods, a single original image will be transformed into countless augmented images, this expands the training data scale and prevents the model from overfitting too early, Figure 8 gives some augmented data examples of a single image.

Table 2. Augmentation for Training Data

Augmentation type	Parameter	Chance
Horizontal scale	0.5 ~ 1.0	50%
Vertical scale	0.5 ~ 1.0	50%
Shear angle	-90 ° ~ +90 °	50%
Rotation angle	-25°~+25°	50%
Perspective transform	Standard deviation: 0.05 ~ 0.1	50%
Flip	Only horizontal flip	50%
Hue and Saturation	-50 ~ +50 (in scale of 255)	100%
Overall scale	0.2 ~ 1.0	100%

Figure 8. Samples of augmented data

4. RESULTS

The training and testing process were on operating system Ubuntu 16.04 LTS with CPU Intel i7-6500 3.20GHz, GPU GeForce GTX1080, DRAM 16GB. We built the system with python 2.7, and the deep learning frameworks are Tensorflow and Keras. The codes are all available online. We also made the codes available for Windows and python 3.7. Figure 9 shows some examples of the classification results on the LISA dataset, the text above the bounding quadrilaterals of car's front-rear gives the classification results, Front, Rear, or Unknown for background class. The number followed by class is the output of the Softmax activation function. The license plate probability is also written at the bottom of the bounding quadrilateral. The results shown in Figure 9 are done by multi-scale testing with input dimensions 256 and 512, which is also the testing dimension yielding the highest mAP score on the LISA dataset.

Figure 10 shows the detection results on the Multiple Cars Scene dataset, this dataset is quite more challenging than the LISA dataset since some of the license plates are relatively small in the images, making it hard to detect with low input dimension, we used multi-scale testing with dimensions 256, 512 and 1024 for those visualization results since it obtained the best mAP on the Multiple Cars Scene dataset.

Figure 9. Classification and pose estimation results of LISA dataset

Figure 10. Classification and pose estimation results of Multiple Cars Scene dataset

Figure 11 gives some examples of the detection missing issue by vehicle detection-based method. By utilizing our proposed method, we successfully avoided this issue and found all of the license plates inside the image. This indicates that our method is more reliable when the amount of vehicle inside an image becomes larger since the overlapped situation among vehicles will increase as well.

Figure 12 and Figure 13 visualize the vehicle pose estimation with license plate localization probability for each pixel and the classification ability of the model by plotting the distribution of those high probability pixels. Our DCNN model can locate the license plate within a precise region and tell all the possible pixels for the car's front and rear.

Figure 11. Comparison between vehicle pose estimation and license plate localization (left column) and proposed method (right column). On the left side, pose estimation and license plate localization missing appeared due to the false regression of license plate; on the other hand, our proposed method can avoid those cases and find all of the license plates

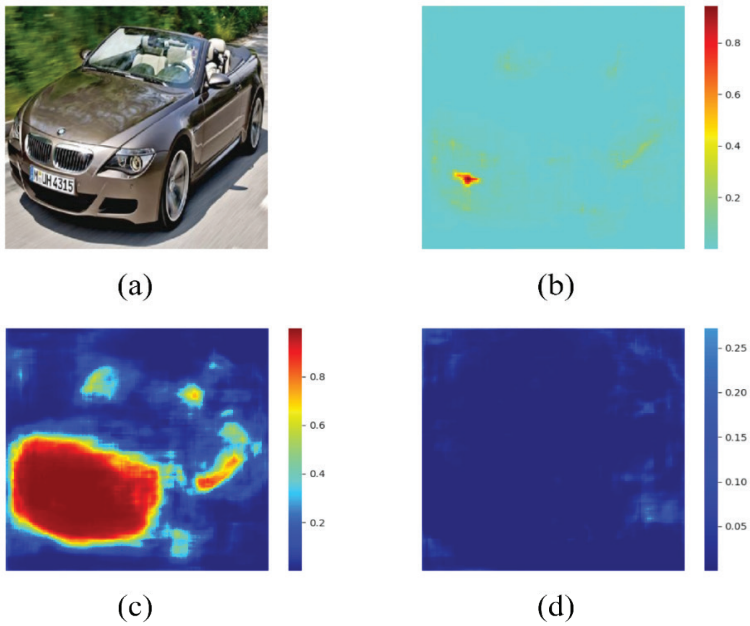

Figure 12: Heatmap of LISA dataset. (a): Input image (b): Vehicle pose estimation with license plate probability (c): Front probability (d): Rear probability

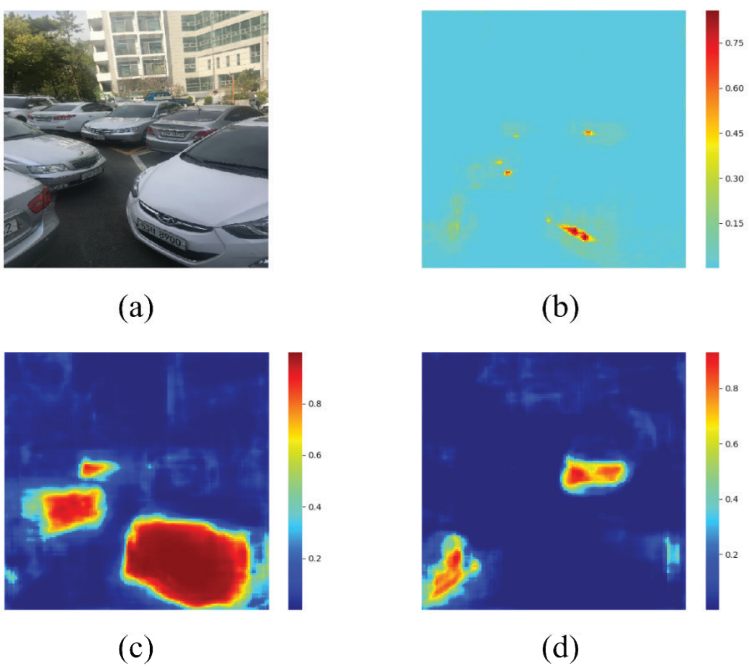

Figure 13. Heatmap of Multiple Cars Scene dataset. (a): Input image (b): Vehicle pose estimation with license plate probability (c): Front probability (d): Rear probability

The learning states for each functional head are shown in Figure 14. The states are the results of single-scale testing with a dimension of 512. For the vehicle pose estimation and license plate mAP50 performance in Figure 14, the model learned fast in the early 300k iterations, after that, the learning state became steady and tended to be overfitting after 600k iterations. The training settings modifications at 330k and 434k fine-tuned the model slightly to get mAP50 beyond 80, but the other datasets did not significantly benefit the mAP for the LISA dataset since the obliqueness levels in the LISA dataset is not that heavy as the ones in the dataset.

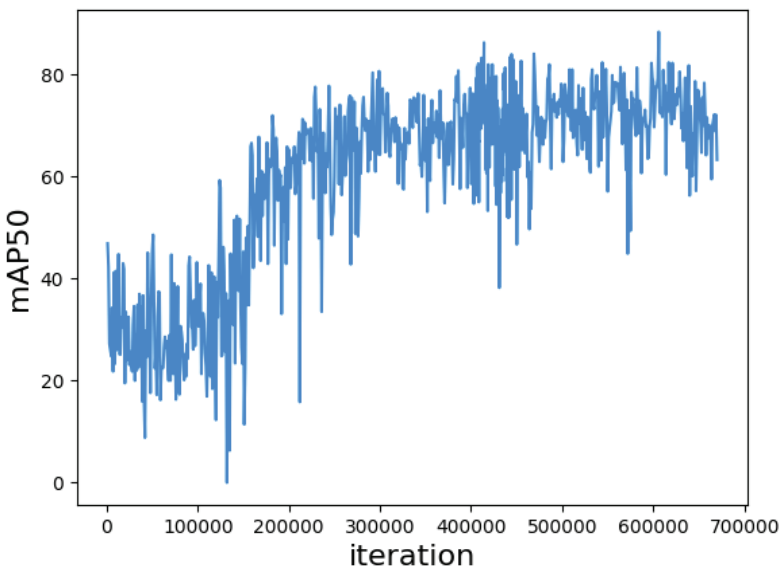

Figure 14. Vehicle pose estimation and license plate localization mAP50 to iteration on LISA dataset

5. CONCLUSION

The proposed DCNN based vehicle pose estimation with license plate locatization model showed the ability to detect license plates under different vision angles. Performance estimation on the dataset with oblique vehicles outperformed the existing commercial systems OpenALPR (K. Simonyan,2015) and SightHound (K. He, 2016), reached mAP/mAP50 of 40.8/90.1. Our system also detects bounding quadrilateral instead of bounding rectangles, yielding a more precise indication for vehicle pose estimation and license plate localization compared to conventional systems. Another main contribution is providing the vehicle information while performing vehicle pose estimation and license plate localization detection, we called this kind of information contextual information, which provides the relation comprehension between the vehicle pose estimation and license plate localization and the vehicle, we got the pose classification accuracy 98.8% and average IoU 71.3%. Applications like traffic scene analysis, we may utilize contextual information for enhancing the interpretation of the vehicle pose estimation and license

plate localization. Since we have obtained the area of the owner car, by further analyzing, we can get; for instance, car brand, model, and color information. In addition, some parking lots tell users to park their cars in a consistent direction (e.g., back-in parking only), the pose information given by our system might help the management of those parking lots.

6. REFERENCES

Du, S., M. Ibrahim, M. Shehata, and W. Badawy. 2013. Automatic License Plate Recognition (ALPR): a state-of-the-art review, IEEE Transactions on Circuits and Systems for Video Technology 23(2), pp. 311-325.

Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J. and Zisserman, A. 2012. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results, available in: http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

Fu, C.-Y., W. Liu, A. Ranga, A. Tyagi and A. C. Berg. 2017. Dssd: Deconvolutional single shot detector, arXiv preprint arXiv:1701.06659.

Girshick, R., J. Donahue, T. Darrell, and J. Malik. 2014. Rich feature hierarchies for accurate vehicle detection and semantic segmentation, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Girshick, R. 2015. Fast R-CNN," in IEEE International Conference on Computer Vision (ICCV).

He, K., X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Law, H., and J. Deng. 2018. CornerNet: detecting vehicles as paired keypoints, in The European Conference on Computer Vision (ECCV).

Lin, T., M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, and C.L.Z.P. Dollár. 2014. Microsoft COCO: Common Vehicles in Context, in The European Conference on Computer Vision (ECCV).

Lin, T.-Y., P. Goyal, R. Girshick, K. He, and P. Doll´ar. 2017. Focal loss for dense vehicle detection, in IEEE International Conference on Computer Vision (ICCV).

Liu, W., D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. Berg. 2016. SSD: single shot multibox detector, in The European Conference on Computer Vision (ECCV).

Newell, A., K. Yang, and J. Deng. 2016. Stacked hourglass networks for human pose estimation, in The European Conference on Computer Vision (ECCV).

Redmon, J., and A. Farhadi. 2018. YOLOv3: an incremental improvement, arXiv preprint arXiv:1804.02767.

Ren, S., K. He, R. Girshick, and J. Sun. 2015. Faster R-CNN: towards real-time vehicle detection with region proposal networks, in Conference on Neural Information Processing Systems (NIPS).

Silva, S., and C. Jung. 2017. Real-time Brazilian license plate detection and recognition using deep convolutional neural networks, in 14th IAPR International Conference on Document Analysis and Recognition (ICDAR).

Simonyan, K., and A. Zisserman. 2015. Very deep convolutional networks for large-scale image recognition, in International Conference on Learning Representations (ICLR).

Sivaraman, S., and M.M. Trivedi. 2010. A General Active Learning Framework for On-road Vehicle Recognition and Tracking, IEEE Transactions on Intelligent Transportation Systems.

Web Sources

Available Online: https://www.openalpr.com.

Available Online: https://www.sighthound.com/press/sighthound-ai-software-now-reads-license-plates.