final report on

VACANT PARKING SPACE ESTIMATION WITH BACKGROUND MODELLING AND DEEP LEARNING

Prof. Ching-Chun Huang
Dr. Hung-Nguyen Manh
Student: Nhan Tran

Contents

1 INTRODUCTION

In crowded cities nowadays, it becomes challenging to find a vacant parking space due to the increasing of private vehicles and limited spaces. As a result, many parking management systems are promised to solve the problem by a substantial amount. Conventional parking management systems utilize a sensor network to sense the occupancy of parking spaces. Though the sensor-based solutions provide an instant way to build up the parking services, the signal received by a sensor might be interrupted in dynamic and cluttered environments. In recent years, deep learning becomes a powerful tool in many computer vision tasks such as classification, detection, segmentation, etc. For this reason, computer vision-based parking management systems caught more attentions as the image sensors capture the environment in a similar way to human.

These systems formulated the finding of vacant parking space as a classification problem. Similar to many other vision-based approaches, an actual parking management system faces many challenges such as perspective distortion, lighting variation, object occlusion and size variation.

In this project, we consider the vacant parking space detection as a density estimation problem. Different from previous works, we make use of the number of available vehicles in a parking lot and then infer the vacant spaces. Other information about location of each occupied space is not used during training. We trained a deep neural network which predicts the density map, each provides in formation about where a parking space is likely to be occupied. We incorporate the result of background modelling into the deep neural network and see that it helps the learning process.

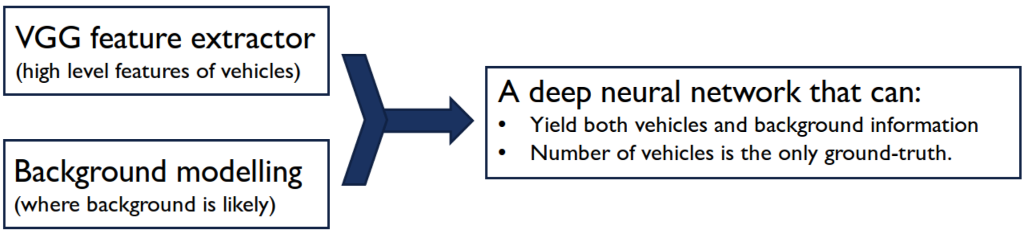

Figure 1: Combining background modelling and features extracted from deep neural network to infer the vacant parking space.

2 RELATED WORKS

In this section, we will address methods that utilize supervised deep learning to solve the vacant parking space problem. Other methods based on hand-crafted features will not be discussed in the scope of this report.

Compared to hand-crafted based approaches, deep learning based approaches show their flexibility and feature generalization. Valipour et al. (2016) proposed to manually crop image regions of interest, which contain the appearance of parking spaces. The cropped and normalized image regions are fed directly into a conventional CNN network for classification. The extracted features are more robust to the lighting and weather variations.

Figure 2: Training images from PKLot dataset.

Figure 3: Valipour et al. (2016) considered the vacant parking space detection as a classification problem.

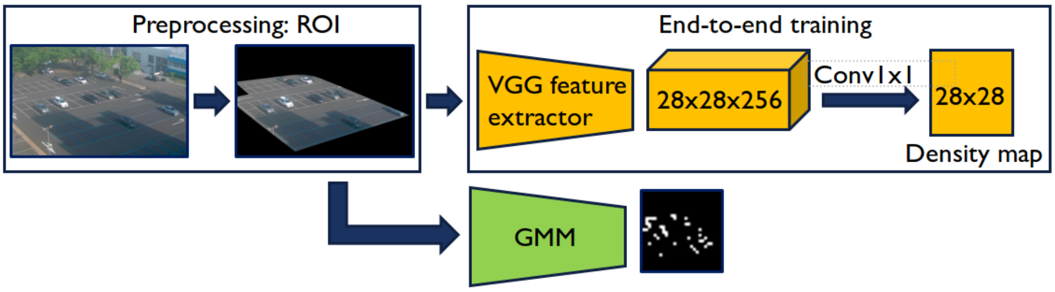

Figure 4: The proposed network structure.

Though the trained network can perform well in most cases, it tends to heavily depend on the diversity of dataset to have a well generalized network. On the one hand, labelling each and every parking space is tedious and time consuming. On the other hand, we usually do not have enough data to train the classifier. This brings us the idea of considering this no longer a classification problem but a density estimation problem. We no longer need the label for each parking space during training. The only cue used for training is the number of available vehicles in a parking lot which usually known beforehand.

3 METHODOLOGY

In this section, we explain in detail about the network architecture and the loss function formulation later used for training. Fig. 4 shows the overall network structure.

The network is composed of two components: a feature extractor based on VGG and a background modelling. We will discuss about these components in detail next.

3.1 VGG Feature Extractor

We adapt VGG16 network for the task of feature extraction. The VGG16 architecture features a simple and uniform design throughout the network. It is always a good idea to start with simple design and yet effective. The feature extractor network is clearly shown in Fig. 5.

We removed the classifier as well as some convolutional and pooling layers at the end of VGG16 network. We reasoned that the later feature maps are too abstract for recognizing vehicles and hence are no longer necessary. We only keep the earlier convolutional and pooling layers and initialized with pre-trained weights. As a result,

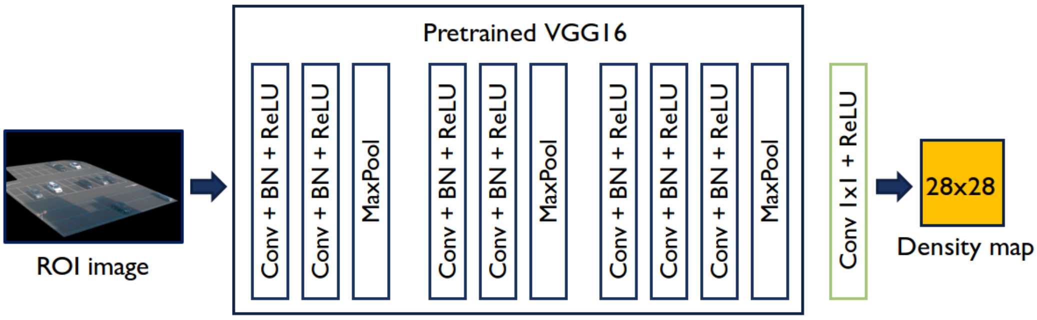

Figure 5: The feature extractor network based on VGG16 with batch normalization.

the output feature map has the shape of $28 \times 28 \times 256$. We append a convolutional layer to transform this feature map to a one dimensional density map with the size of 28×28. Finally, a non-linearity, in this case ReLU is added. The density map is expected to infer where a vehicle is likely to be in the parking lot.

3.2 Background Modelling

Background modelling is the task of telling where in the image the background is likely to be. We perform background modelling using Gaussian mixture models (GMMs) for their capacity and statistical nature.

A Gaussian mixture model attempts to find a mixture of multi-dimensional Gaussian probability distributions that are best model any input dataset. Though GMM is often categorized as a clustering algorithm, fundamentally it is an algorithm for density estimation. That is to say, the result of a GMM fit to some data is technically not a clustering model, but a generative probabilistic model describing the distribution of the data. An example where GMM is used for density estimation is shown in Fig. 6.

We utilize the built-in APIs from Scikit-learn framework to perform background modelling on parking lot images. Particularly, we use custom functions provided by Scikit-learn to fit a GMM model which takes background patches as inputs. We then use the fitted GMM model to predict a score for each pixel in a parking lot image. The score will tell how much that particular pixel is likely to be background. Using this approach we can predict a binary map where black color denotes background and white color denotes non-background. The trickiest part is mostly the selection of background patches which later used to fit the GMM model. In practice, the background selection should cover the background in several conditions such as lighting variation, object occlusion, weather variation, etc. Fig. 7 shows our background

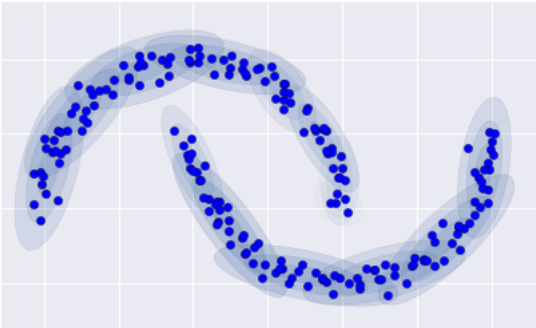

Figure 6: GMMs as density estimation example

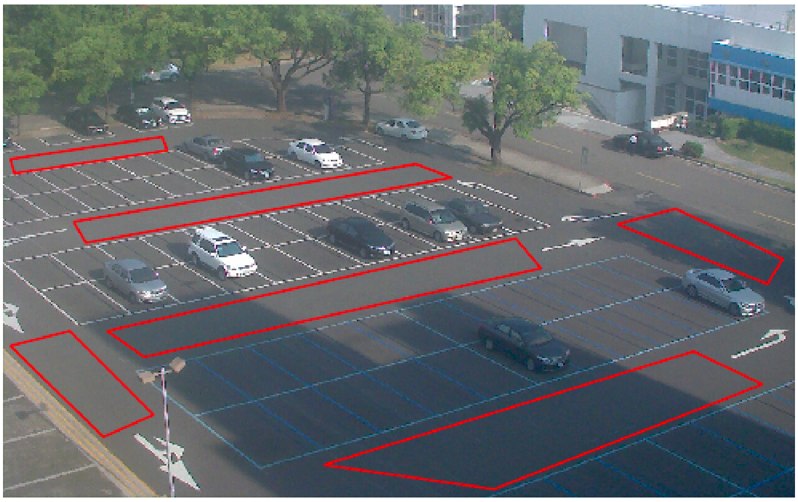

Figure 7: Background selection to fit GMM

selection though it might not yield the best result.

The number of Gaussian components to fit a GMM is also a crucial parameter to make sure GMM can capture a good estimation of the background. In Fig. 8 we showed how different number of Gaussian components can affect the final result.

Though background modelling with GMMs yield acceptable results, it is not a perfect solution. The necessary of thresholding makes the background modelling results affected by human decision. Another reason is that GMMs perform worse at regions where vehicle color is very similar to background color.

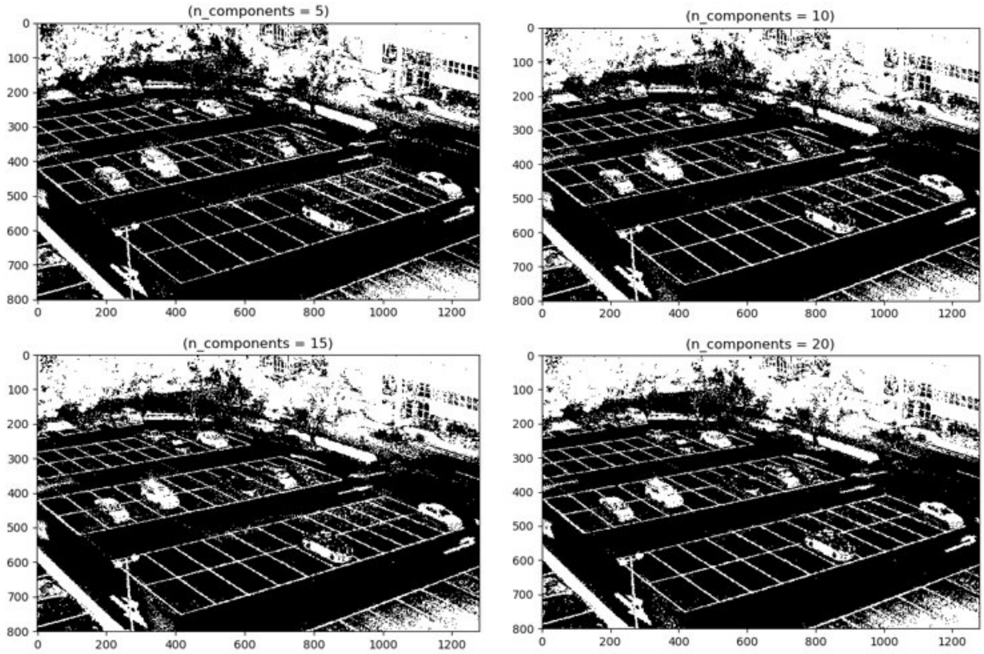

Figure 8: How different number of Gaussian components affect the background prediction

3.3 Loss Function Formulation

In this section, we formulate the loss function which will be used later during back-propagation of the deep neural network.

$$loss_{total} = \lambda \times loss_{count}(sum(I_{pred}, n)) + \beta \times loss_{gmm}(I_{pred}, I_{gmm})$$

where,

- I_{pred} is the predicted density map (where a vehicle is most likely in a parking lot).
- n is the number of available vehicles in a parking lot (ground-truth).
- I_{gmm} is the density map yielded by GMM.
- λ and β are hyper-parameters.

The total loss is composed of two components: $loss_{count}$ and $loss_{gmm}$. The former calculates the difference between the predicted number of vehicles and ground-truth while the later calculates the difference between the predicted density map and GMM density map. In this setting we use Mean Square Error (MSE) loss but other losses can be easily adapted.

4 TRAINING SETTING

Dataset

The dataset that we use for training is composed of 348 images of parking lot captured in several lighting conditions and parking status. We used *Adam* optimizer at a learning rate of $1e-4$. We trained the deep neural network for 300 epochs in several settings of λ and β.

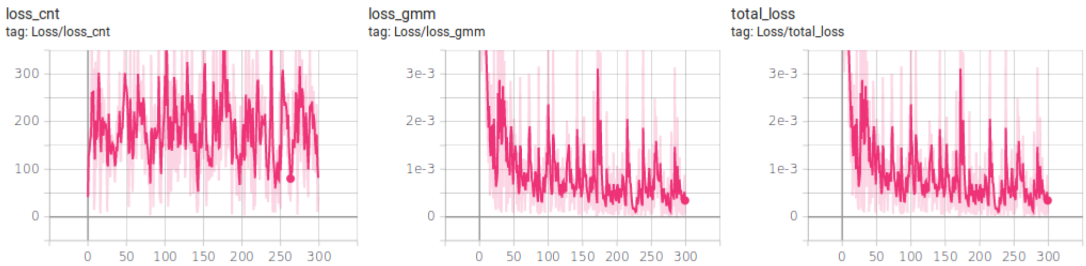

Figure 9: $\lambda = 0; \beta = 1$

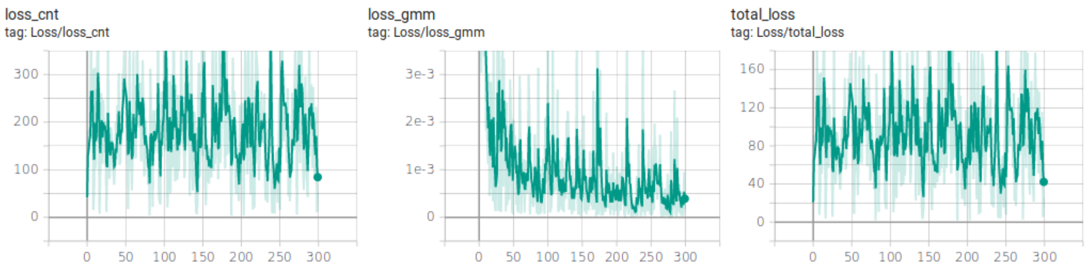

Figure 10: $\lambda = 0.5; \beta = 1$

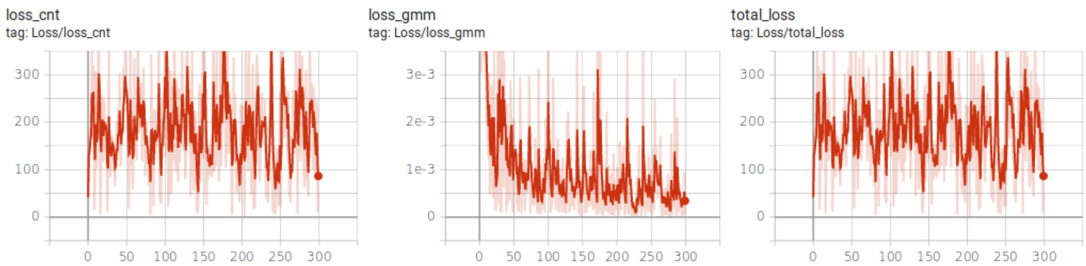

Figure 11: $\lambda = 1; \beta = 1$

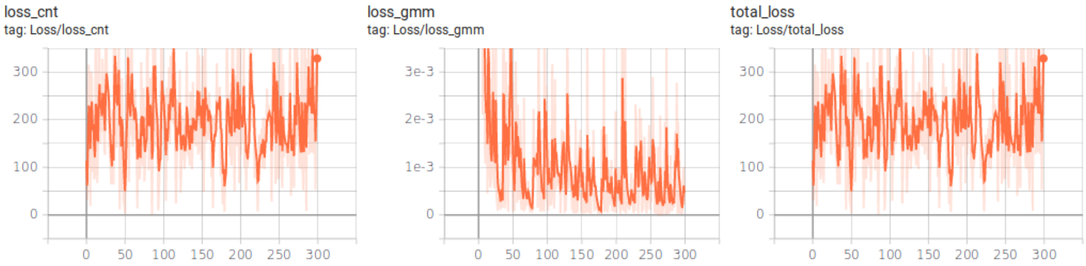

Figure 12: $\lambda = 1; \beta = 0.5$

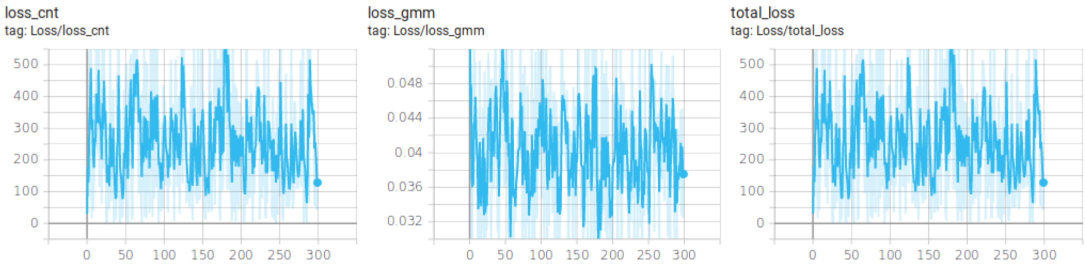

Figure 13: $\lambda = 1; \beta = 0$

We can see that $loss_{count}$ is not likely to converge while $loss_{gmm}$ could converge after substantial amount of epochs as well as β is non-zero. We argue that the number of available vehicles does not provide enough cues for the network to learn a representation of density map.

5 CONCLUSION

In this project, we formulated the finding of vacant parking space no longer a classification problem but a density estimation problem. We proposed to incorporate the background modelling predictions, which used Gaussian mixture models ($GMMs$), with features extracted from a family of VGG networks. We trained a deep neural network to predict a density map which can tell where the vehicle is likely to be in the parking lot image. Each network component is assigned with one specific loss which later constitutes the total loss function. During training, the loss associated with GMM density map tends to converge after some epochs while the loss associated with the number of available vehicles shows no evidence of convergence. This implies that using only the number of vehicles could not provide enough information for the network to learn a representation of desired density map.

We hope that future works can exploit the pros and cons of this project to solve the aforementioned challenges.

References

Scikit-learn. Gaussian mixture. https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html#sklearn.mixture.GaussianMixture.score_samples. Accessed on 2019-10-05.

Valipour, S., Siam, M., Stroulia, E., and Jägersand, M. (2016). Parking stall vacancy indicator system based on deep convolutional neural networks. *CoRR*, abs/1606.09367.