# Exploring Various Methods of Parking Space Detection in Image

Yuan Chen, Fan Wu, Chu-An Hsiao, Daolun Chen

Department of Computer Science
Rice University
6100 Main St, Houston, TX 7700

{yc149, fw21, ch86, dc59}@rice.edu

## Abstract

In this work we investigate the effect of the convolutional network depth on its accuracy in the large-scale image recognition setting. Our main contribution is a thorough evaluation of networks of increasing depth using an architecture with VGG(Simonyan et al, 2014) and ResNet(Kaiming et al., 2015) on the prior-art configurations that can be achieved by pushing the depth of the number of layers.

We also show that our representations generalize well to other datasets, where they achieve state-of-the-art results. Our code is available on https://github.com/fw21/COMP576
to facilitate further research on the use of deep visual representations in computer vision.

## 1. Background

As a graduate student who live off campus, it is difficult for students to find parking spot in the campus, We believe that, with our tool to identify available parking lots, we believe it will improve students and faculties life quality and make the best use of parking space

The goal for our research was to take any static image or video of a parking lot and be able to automatically detect whenever a parking space was available or occupied. that's used for image analysis. Combining it with our knowledge of CNN and RNN, we believe we will make it more efficient and intuitive.

### 1.1 Recurrent Neural Network

A recurrent neural network (RNN) is a type of artificial neural network that is used to process sequential data. RNNs have the ability to use information from previous input in the current input, which allows them to learn patterns and dependencies in data that has a temporal or sequential structure. This makes them well-suited for tasks such as language modeling, machine translation, and speech recognition.

### 1.2 ConvolutionalNeural Network

A convolutional neural network (CNN) is a type of deep learning neural network that is commonly used in image and video recognition tasks. CNNs are designed to take advantage of the 2D structure of image data by using multiple filters to learn different features of the data. This allows them to learn complex patterns and features in the data, which allows them to perform well on tasks such as image classification, object detection, and facial recognition.

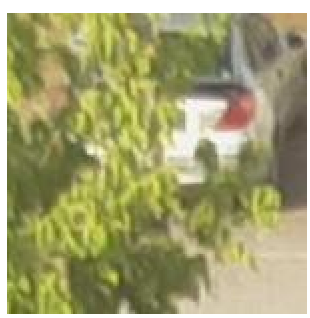
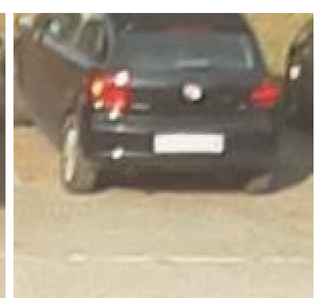
In a classic document on ECCV in 2014: "Visualizing and Understanding Convolutional Networks", The purpose of this document is to use feature visualization to tell us how to see that our accuracy has indeed improved from the perspective of visualization, and the features we learned by designing CNN are indeed more powerful.

## 2. Dataset Overview

This project used the CNRPark+EXT dataset for model training and testing, which contains images of vacant or occupied parking spaces. We utilized CNT-EXT-Patches-150x150(Amato et al. 2017) as training data, and CNRPark-Patches-150x150 as testing dataset. The following are the detailed information regarding CNRPark-Patches- 150x150 and CNT-EXT-Patches-150x150.

CNRPark-Patches-150x150: 12,584 parking space images taken in July 2015 from 2 different cameras.

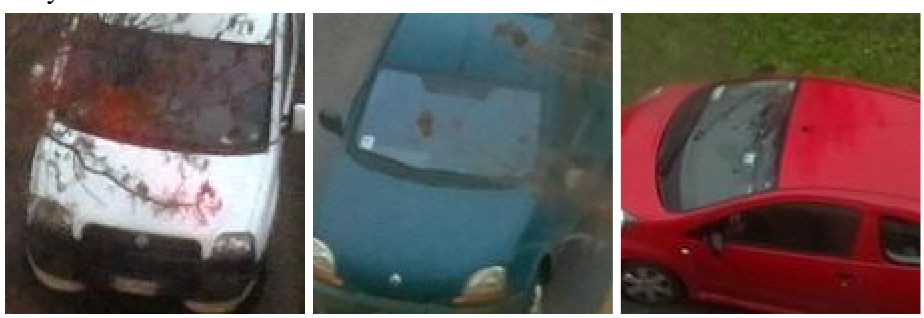
- CAMERA: the camera that took the image (A or B)
- CLASS: the state of the parking space (free or busy)
- YYYYMMDD_HHMML: the date and time image collected
- SLOT_ID: local ID for the monitored slot given by each camera
- Sample Images
  - Busy

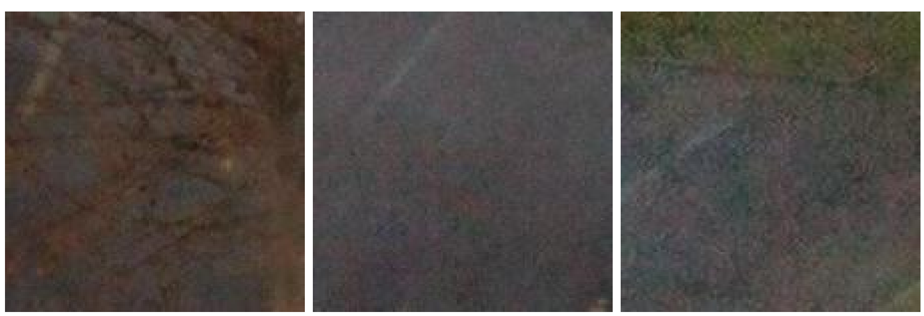
- Free

<u>CNT-EXT-Patches-150x150</u>: 144,965 parking space images taken between Nov. 2015 to Feb. 2016 from 9 cameras.

- CAM_ID: the number of the camera (from 1 to 9)
- WEATHER: the weather when the image was taken (Sunny, Overcast, or Rainy)
- CAPTURE_DATE: the date image was captured, format: YYYY-MM=DD
- CAPTURE_TIME: the time image was captured, format: HH.MM
- SLOT_ID: global ID for the monitored slot
- Label: 0 for free, 1 for busy
- Sample Images
    - Busy

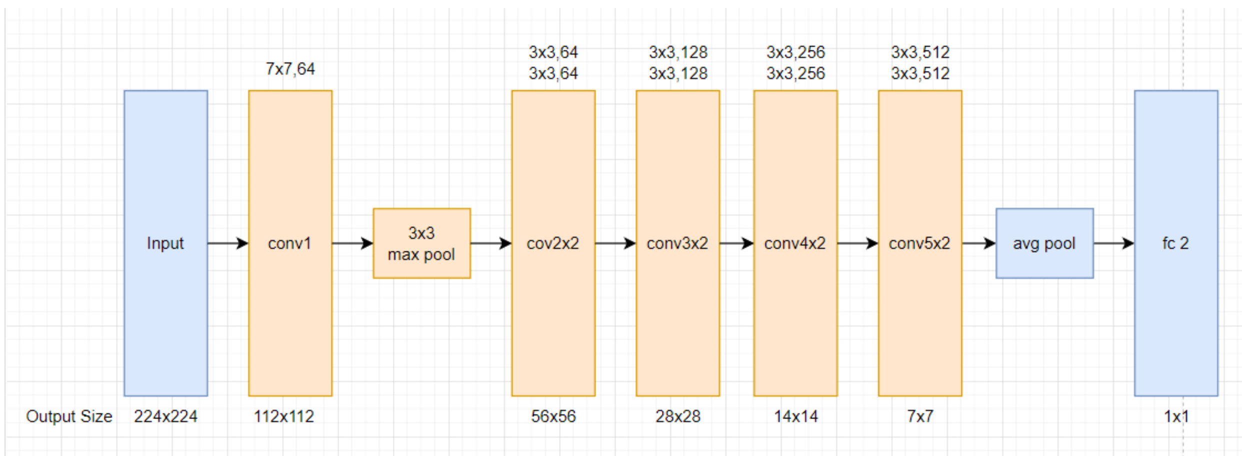


    - Free



# 3. Model Building and Optimization

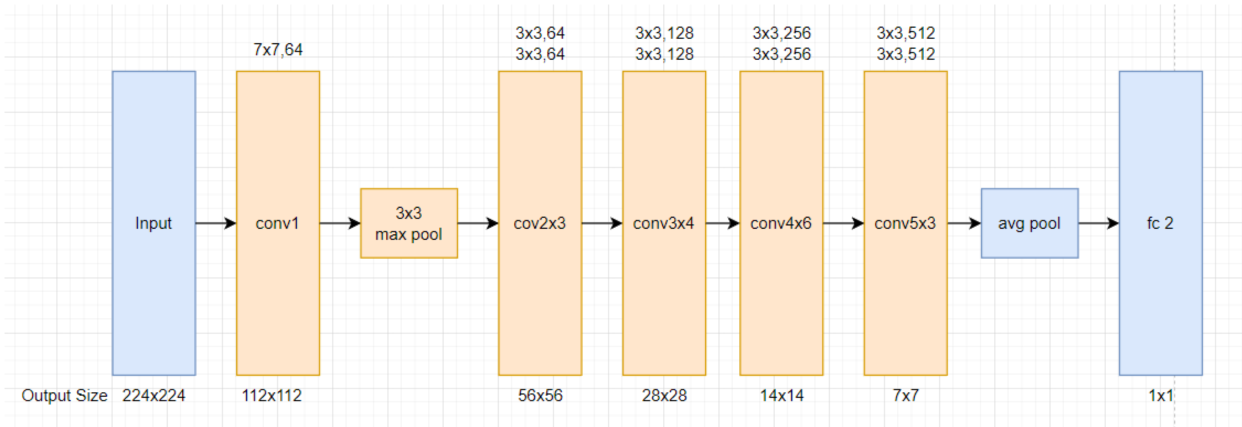
## 3.1 ResNet18(Kaiming et al., 2015)




ResNet is a convolutional neural network architecture developed by Microsoft Research. It is a variant of the ResNet architecture, which was introduced by Kaiming He et al. in their 2015 paper "Deep Residual Learning for Image Recognition." It is a relatively shallow architecture, with 18 convolutional layers. It is characterized by the use of "shortcut" or "skip" connections, which allow the network to learn residual functions. This makes it possible for the network to learn very deep networks without encountering the problem of vanishing gradients.

ResNet18 is commonly used for image classification and object detection tasks. It is trained on large datasets such as ImageNet, which contains millions of labeled images in more than 1,000 classes. This allows the network to learn to recognize a wide variety of objects and scenes.

## 3.2 **ResNet34**

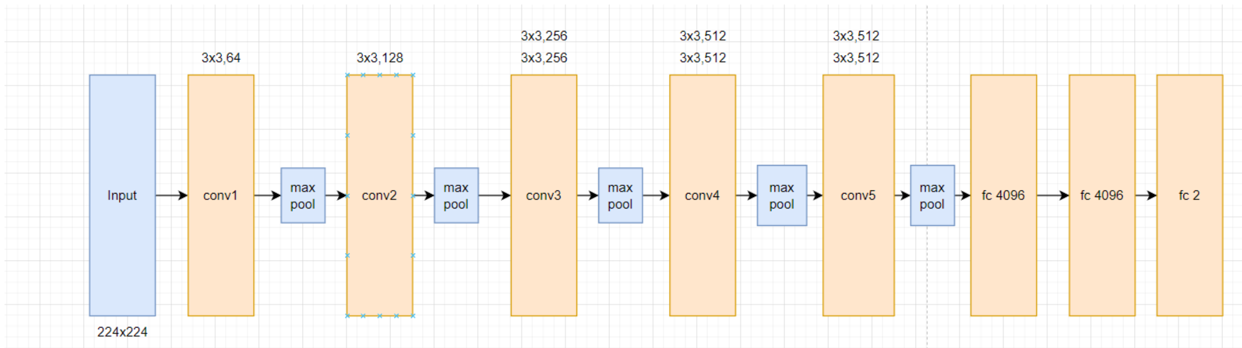



ResNet34 is a 34 layers network which is commonly used for image classification and object detection tasks.

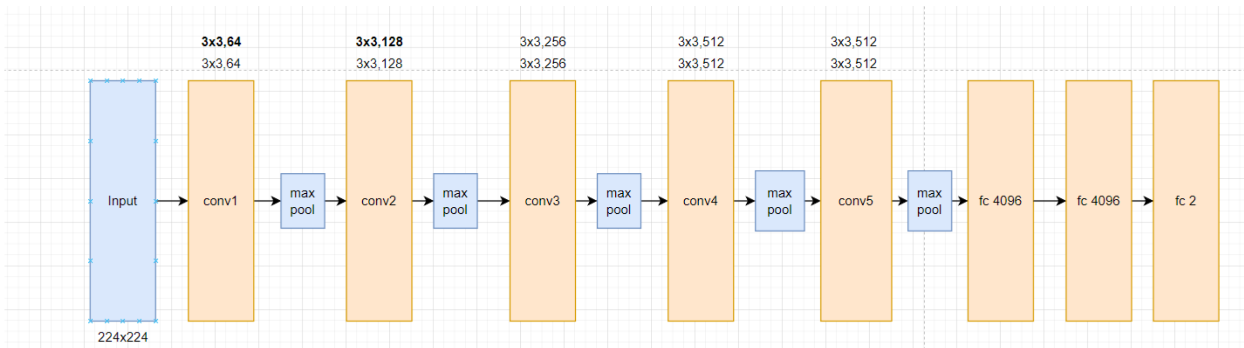
## 3.3 Vgg11(Simonyan et al, 2014)

VGG11 is a convolutional neural network architecture developed by the Visual Geometry Group (VGG) at the University of Oxford. It is a variant of the VGG network, which was developed by Karen Simonyan and Andrew Zisserman and won the 2014 ImageNet Large Scale Visual Recognition Challenge (ILSVRC). VGG11 has 11 convolutional layers, while VGG13 has 13 convolutional layers, VGG11 also has fewer parameters than the other VGG architectures, which makes it faster to train and less prone to overfitting.

Like other VGG networks, VGG11 is characterized by its use of small convolutional filters (3x3) and a deep network architecture. This allows the network to learn a rich set of features from the input data, which can be used for a wide range of tasks, such as image classification, object detection, and segmentation. VGG11 is typically trained on the ImageNet dataset, which contains millions of labeled images in more than 1,000 classes. This allows the network to learn to recognize a wide variety of objects and scenes.




## 3.4 Vgg13

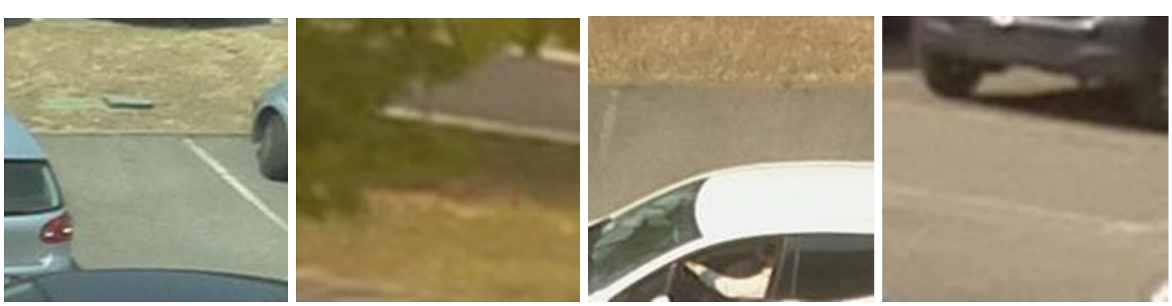



VGG13 is a variant of the VGG network. VGG13 has 13 convolutional layers,which makes it faster to train and less prone to overfitting.

# 4. Challenges

## 4.1 Dirty Data

The first challenge was dirty data. In the dataset there were wrongly labeled data, usually caused by distraction of other nearby objects like trees and cars passing by. For example, the following images were labeled as busy but actually are free. We can see that because of the bad view angle of cameras, objects appearing in images can be a distraction. But in order to improve the robustness of our model, manual data cleaning was performed.



Also, here are some images marked as free but were actually busy or couldn't be determined (covered by tree).

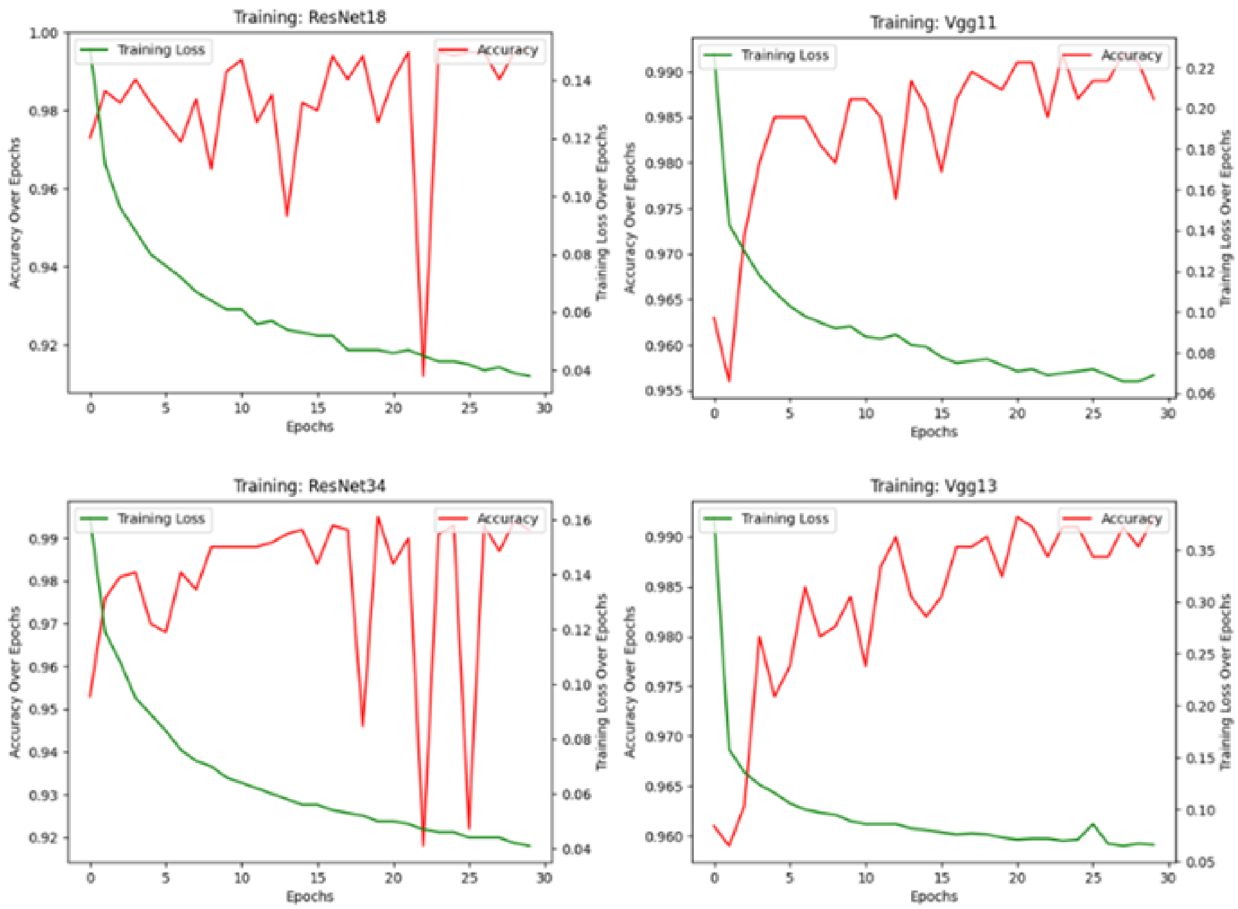
Our manual data cleaning changed wrong labels and removed undetermined images.

### 4.2 Dataset

The other challenge comes from the size of the dataset. Original CNRPark dataset can't be divided evenly into two of our classification labels, and more images are marked as busy than as free. From the first couple trials of training, we also noticed that the accuracy rate was very high and reached 97%, while when doing the prediction, the accuracy was not so ideal as the training. There could be overfitting due to a large number of similar images in the dataset which caused this in the training and also made the training process more time wasting. In our project, considering that we divided the original dataset into training set and validation set, the problems mentioned above can be mitigated by trimming the original dataset. We removed busy images from the training set into the validation set. And we kept a 3:7 portion between the sizes of validation and training set.

## 4. Training

In this project, we used Pytorch to implement our models and deployed them on Google Colab. Each model was trained for 30 epochs. Training loss and accuracy were recorded over epochs for plotting figures. The figure below is the visualized training process of all four models, there are two curves for each model, one for training loss and one for training accuracy.

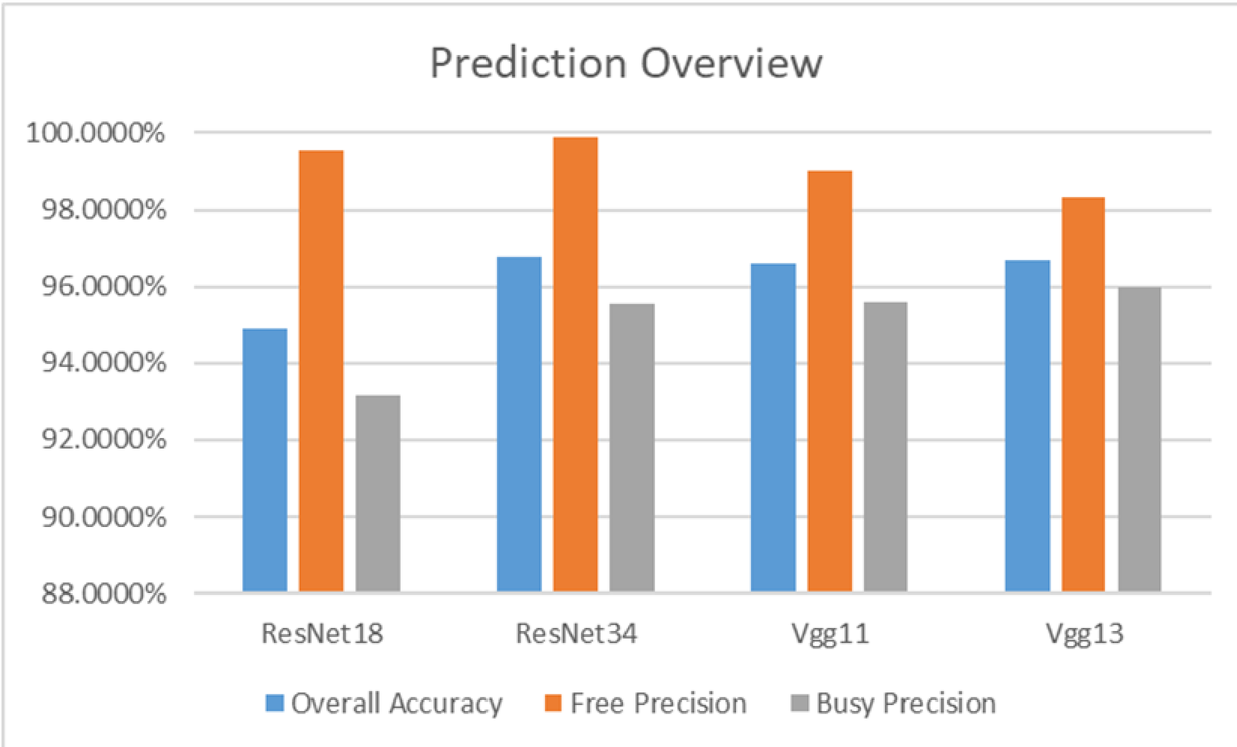



We can see though there was fluctuation when training ResNet, but the training loss still kept decreasing and accuracy rate generally maintained a very high value of more than 0.9. And, among four models, the training loss decreased most smoothly for ResNet34, according to the figure.

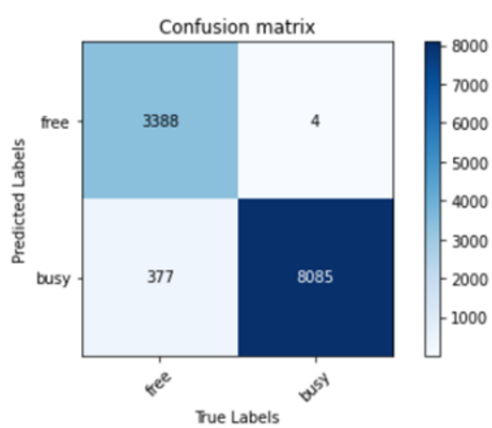
# 5. Results

## Prediction Results

After the training, we used four models to perform predictions respectively. Three types of data were recorded: overall accuracy, free label precision rate and busy label precision rate. The bar chart presented below includes all three statistics for models' prediction.




Among them, as for precision rate of free label, ResNet34 performed best; as for busy label precision rate, Vgg13 was the best. The ResNet performed best as for overall prediction accuracy rate, slightly overperformed other two models (Vgg11, Vgg13) and had a significant improvement compared with ResNet18, reaching an accuracy rate of 0.9679. Considering the overall accuracy, ResNet34 was chosen as our final solution, with hyper parameters as listed: ReLU activation, Adam optimization of learning rate 0.0003. However, no matter what model was chosen, there was always a significant difference between precision rates of busy and free labels. In order to analyze this problem, the confusion matrix of the model ResNet34 is shown below.




the model accuracy is 0.9678589505652101

| | Precision | Recall | Specificity |
|---|---|---|---|
| free | 0.999 | 0.9 | 1.0 |
| busy | 0.955 | 1.0 | 0.9 |

From the confusion matrix, there were 4 images with a busy parking lot predicted to be free and 377 images with a free parking lot but predicted to be busy. This contrast provides a simple and direct answer to the issue in the figure of prediction overview. In order to find the root cause of this problem, we dig into the original images in the dataset. Here are some misclassified images.

Predict Label: Free True Label: Busy

Two images above were predicted as free and there are trees that overlap with parked cars which also occupy the most space of images. And the other type of wrong prediction is like the images below.

Predict Label: Busy True Label: Free

Images shown above are affected by visual distraction. Trees appear in the first two images, and the shade in the second picture. Our model regarded them as something that occupied the parking lot.

Predict Label: Busy True Label: Free

Next two images, cars misled the prediction. In the third picture, the car and shade overlapped with the free lot. As for the last picture, we believe if a clearer mark was made, our model could better recognize the lot and make the right prediction.

More cases were found where free parking lots were predicted as busy. Mainly because of the visual distraction from trees, cars and shade. Any kind of bad view angle may cause such an issue. And from the observation, we also believed that the prediction accuracy would drop if trees appear more in images.

## 6. Future Work

The next step of our plan is to use our trained model to predict parking lot vacancy based on videos recorded by security cameras. By sampling snapshots of the video, the model can make predictions for parking lots. Before the experiments on videos, here are two points we need to improve: accuracy and efficiency.

### 6.1 Accuracy Improvement

As for prediction accuracy rate, here are some future works to improve it. First, in real world application, from the perspective of a parking lot, a clearer mark can be made to notify cameras and model the exact boundary of the predicted lot. Second, better view angle can be provided for security cameras, for example a higher installation spot can mitigate the problem of overlapping objects (a top view can be best). Third, more cameras should be deployed to make a joint prediction because setting cameras from different angles reduces the amount of overlapping.

### 6.2 Efficiency Improvement

Currently, it costs around 30 seconds to predict 11,854 images using the Colab platform. Considering deployment on security cameras, on real time embedded systems with lower memory and CPU processing capacity, simplification should be made. We believe there is a method to reduce the number of total neurons of the network because our task, compared with the original task of ResNet34, was reduced to binary classification (busy, free). So, possible simplification and optimization will be conducted to our model in the future.

## 7. Conclusion

In this project, we trained four different models (ResNet18, ResNet34, Vgg1, Vgg13) on the CNRPark+EXT dataset. Among the prediction results, ResNet34 performed best, of more than 96.79% accuracy rate, but still needs to be optimized and simplified. Although we reach the end of the semester, we believe that other people could benefit from building up our work. We believe that we could improve the accuracy of our task, especially for Rice University students and faculty. Training and using Rice University parking lot data would serve our goal and benefit for students.

## References

[1] Russakovsky, Olga, et al. "Imagenet large scale visual recognition challenge." *International journal of computer vision* 115.3 (2015): 211-252.

[2] Amato, Giuseppe, et al. "Deep learning for decentralized parking lot occupancy detection." *Expert Systems with Applications* 72 (2017): 327-334.

[3] Amato, Giuseppe, et al. "Car parking occupancy detection using smart camera networks and deep learning." *2016 IEEE Symposium on Computers and Communication (ISCC)*. IEEE, 2016.

[4] “UFPR.” [Online]. Available: https://www.inf.ufpr.br/lesoliveira/download/pklot-readme.pdf. [Accessed: 16-Nov-2022].

[5] “Barry street dataset: Segmentation of the individual 30 parking spaces ...” [Online]. Available:https://www.researchgate.net/figure/Barry-Street-dataset-segmentation-of-the-individual-30-parking-spaces-conditions-namely_fig3_323796590. [Accessed: 16-Nov-2022].

[6]Olgarose, “Olgarose/Parking Lot: Parking Space Detection in opencv,” *GitHub*. [Online]. Available: https://github.com/olgarose/ParkingLot. [Accessed: 15-Nov-2022].

[7]“CNRPARK+EXTA dataset for visual occupancy detection of parking lots Giuseppe Amato, Fabio Carrara, Fabrizio Falchi, Claudio Gennaro, Claudio Vairo,” *CNR Parking Dataset - Dataset for visual occupancy detection of parking lots*. [Online]. Available: http://cnrpark.it/. [Accessed: 15-Nov-2022].