Parking Vacancies Detection

Course: Big Data Content Analytics

Program: MSc Business Analytics

Names: Synefakis Konstantinos & Mantziaris Nikolaos

Description

Parking Space is a serious problem of 21st century. Finding ways to inform the inhabitants of a city or a space if there is any empty space to park the car is crucial. It is interesting because you can substitute the expensive sensors for each parking spot with cameras.

Business-wise there are already several startups that try to fulfill this growing need. Our project can lead to a startup that will deploy the model on cloud and offer its services to municipalities in order to help them with public park spaces and even with private parking places.

Mission

There is already some type of work done by the creators of the dataset. They used this dataset and ParkLot and with the use of AlexNet and BVLC Caffe framework they used CNNs to try to tackle this.

We could not find a legal way to access the papers so we don't have much knowledge about what type of implementation they did. We decided to use CNNs with Tensorflow environment plus some already created models with fastAI called Mask-R-CNN and Resnet50 which is a 50 layer Residual Network.

Data

Our Data is found in the site cnrpark.it. We used CNRPark-Patches for our model and the CNR-EXT-FULLIMAGE for the ready made model. We used some data augmentation in order to prevent overfitting like random flipping and ZCA whitening.

Our dataset contains 150x150 images of

- <CAMERA> can be A or B,
- <CLASS> can be free or busy,
- YYYYMMDD_HHMM is the zero-padded 24-hour capture datetime,
- <SLOT_ID> is a local ID given to the slot for that particular camera

Full frames of the cameras belonging to the CNR-EXT subset. Images have been down sampled from 2592x1944 to 1000x750 due to privacy issues.

Files follow this organization:

FULL_IMAGE_1000x750/<WEATHER>/<CAPTURE_DATE>/camera<CAM_ID>/<CAPTURE_DATE>_<CAPTURE_TIME>.jpg.

Where:

- <WEATHER> can be SUNNY, OVERCAST or RAINY,
- <CAPTURE_DATE> is the zero-padded YYYY-MM-DD formatted capture date,
- <CAM_ID> is the number of the camera, ranging 1-9,
- <CAPTURE_TIME> is the zero-padded 24-hour HHMM formatted capture time.

The archive also contains 9 CSV files (one per camera) containing the bounding boxes of each parking space with which patches have been segmented. Pixel coordinates of the bounding boxes refer to the 2592x1944 version of the image and need to be rescaled to match the 1000x750 version.

Methodology

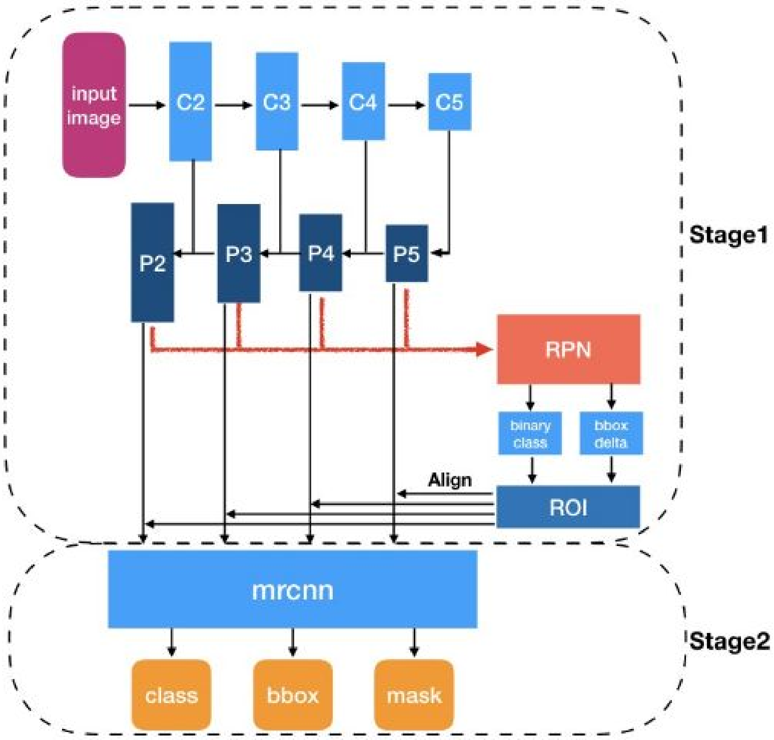
We will use CNN,Mask-R-CNN and ResNet 50 models.

Our CNN consists of

```
=================================================================
conv2d_1 (Conv2D)            (None, 150, 150, 32)      2432
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 75, 75, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 75, 75, 64)        18496
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 37, 37, 64)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 37, 37, 96)        55392
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 18, 18, 96)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 18, 18, 96)        83040
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 9, 9, 96)          0
_________________________________________________________________
flatten_1 (Flatten)          (None, 7776)              0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               3981824
_________________________________________________________________
activation_1 (Activation)    (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 1026
=================================================================
Total params: 4,142,210
```

We use ReLu activation and only in the last dense layer we use Softmax for activation. We also used Adam Optimizer with lr = 0.001, with categorical cross entropy as loss and the accuracy for metric.

Mask-R-CNN consists of :



First stage

A lightweight neural network called RPN scans all FPN top-bottom pathways (hereinafter referred to feature map) and proposes regions which may contain objects. That's all it is.

While scanning the feature map in an efficient way, we need a method to bind features to its raw image location. Here come the anchors. Anchors are a set of boxes with predefined locations and scales relative to images. Ground-truth classes(only object or background binary classified at this stage) and bounding boxes are assigned to individual anchors according to some IoU value. As anchors with different scales bind to different levels of feature map, RPN uses these anchors to figure out where of the feature map 'should' get an object and what size of its bounding box is. Here we may agree that convolving, downsampling and upsampling would keep features staying the same relative locations as the objects in original image and wouldn't mess them around.

Second Stage

Another neural network takes proposed regions by the first stage and assign them to several specific areas of a feature map level, scans these areas, and generates objects classes (multi-categorical classified), bounding boxes and masks. The procedure looks like RPN. Differences are that without the help of anchors, stage-two used a trick called ROI Align to locate the relevant areas of feature map, and there is a branch generating masks for each object in pixel level. Work completed.

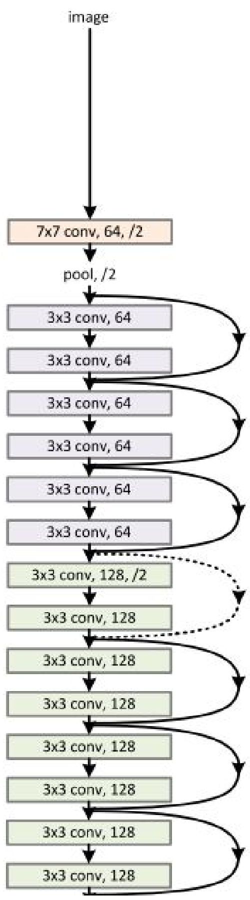
Mask-R-CNN uses SGD as optimizer with lr 0.001

ResNet 50

In General

In a deep convolutional neural network, several layers are stacked and are trained to the task at hand. The network learns several low/mid/high level features at the end of its layers. In residual learning, instead of trying to learn some features, we try to learn some residual.

Residual can be simply understood as subtraction of feature learned from input of that layer. ResNet does this using shortcut connections (directly connecting input of nth layer to some (n+x)th layer. It has proved that training this form of networks is easier than training simple deep convolutional neural networks and also the problem of degrading accuracy is resolved.

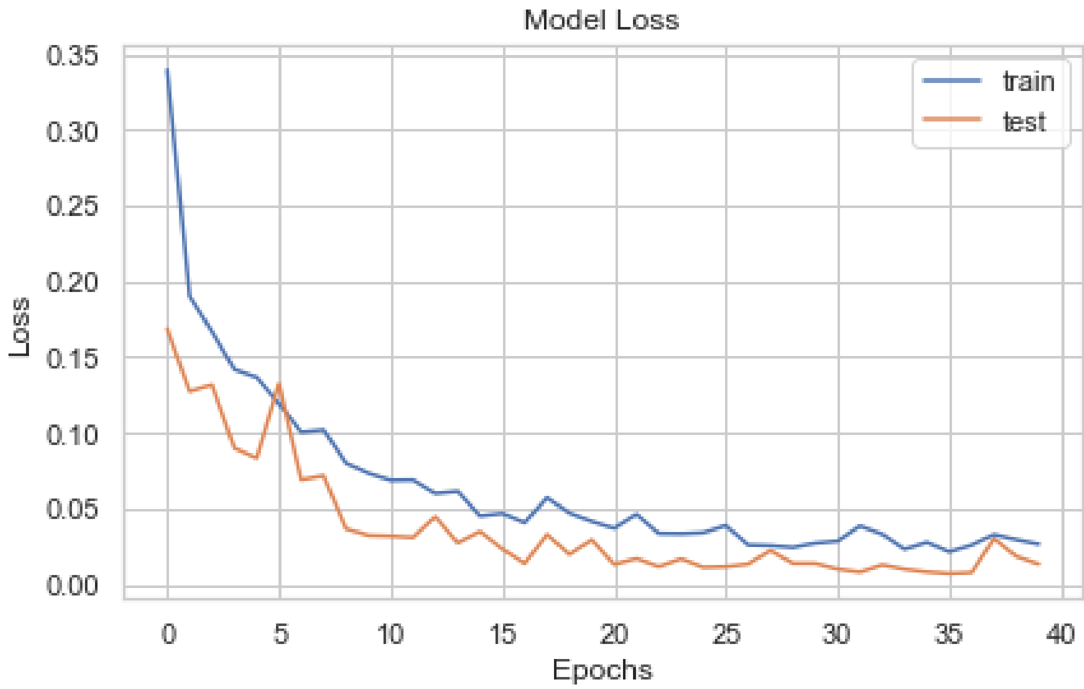


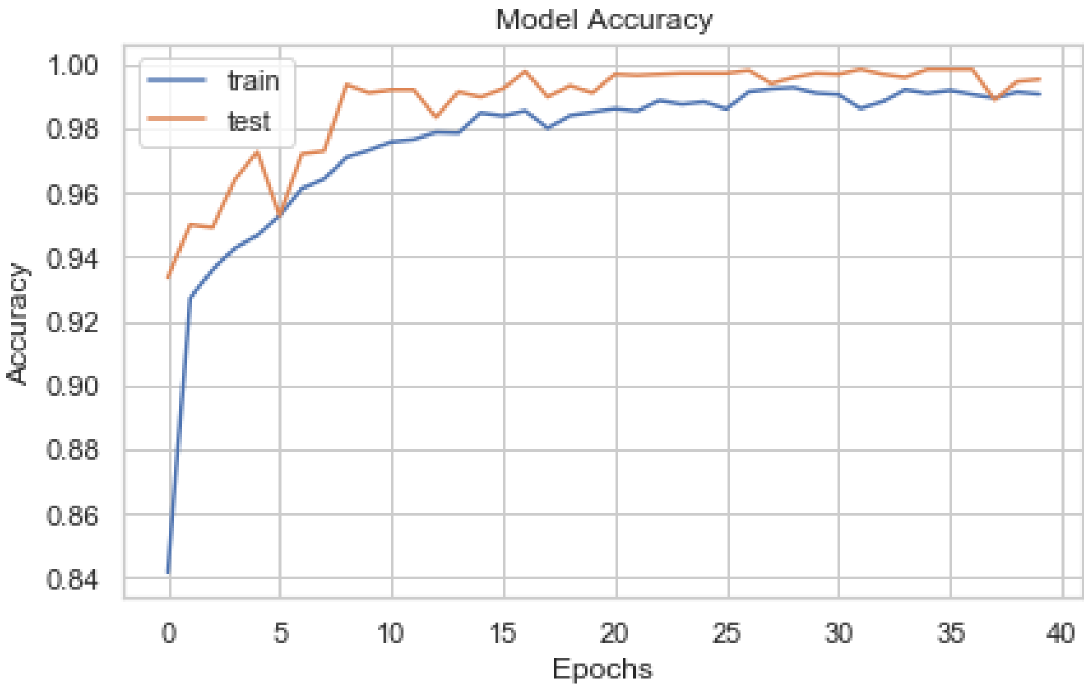
Our Mask-R-CNN model is trained in COCO dataset and afterwards we just use it in our own dataset.

Results

Our results are based mostly on the visual part of the models since we used pretrained data but we will show the results of our model.

Our CNN:



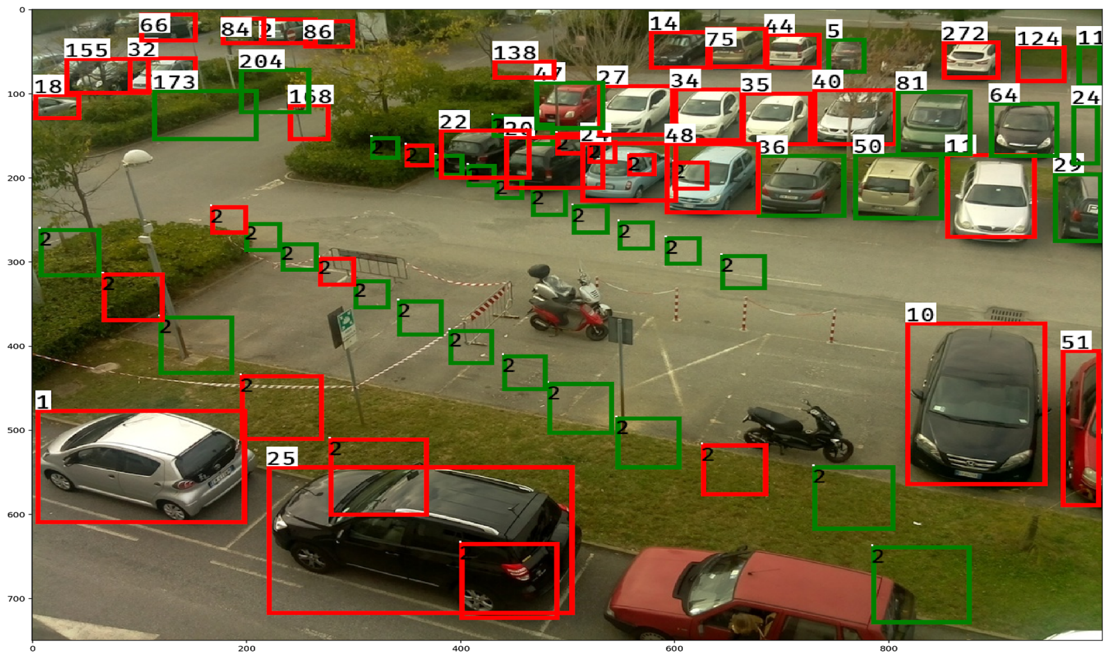


The Resnet50 that is fully trained with all available data has a confusion table

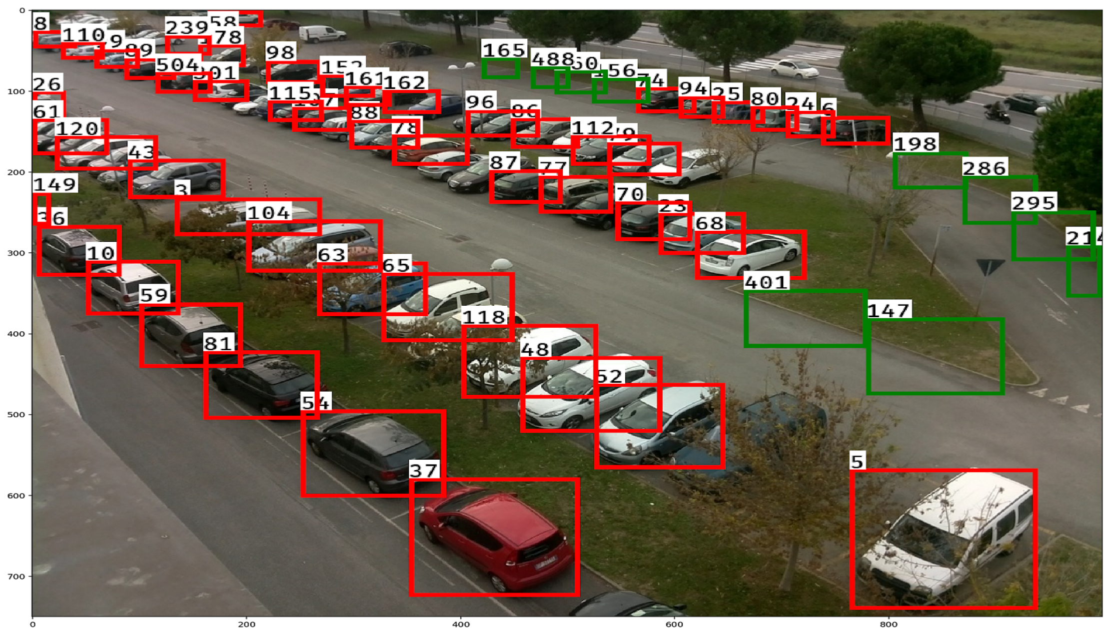
13941	80
59	15813

Image Results

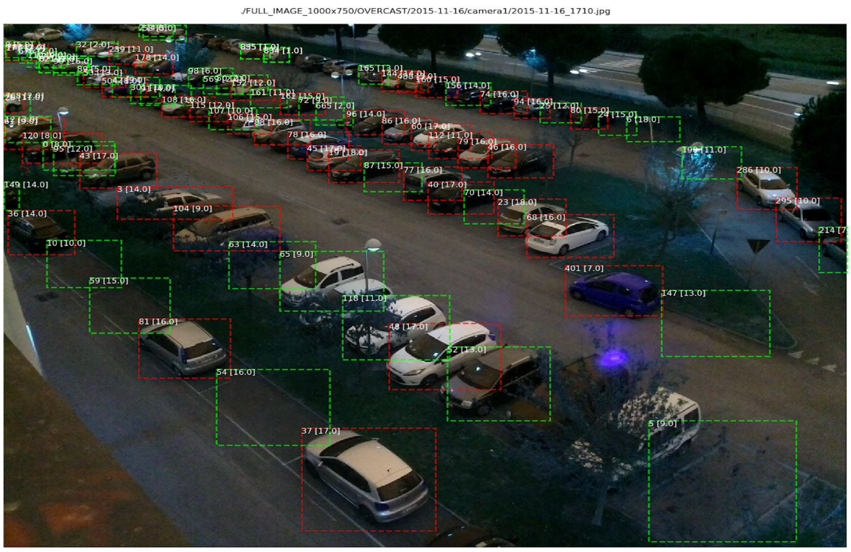
Our CNN



Resnet50



Mask-R-CNN



Other Notes

The biggest challenge was to find the proper park-spaces of the available parking and make an acceptable model. Generally, our model is barely acceptable but it is a first step in order to the more complex solutions.

We needed 34 minutes for the Resnet50 model, 8 minutes for Mask-R-CNN and 2 minutes for our own CNN.

We used a PC with Intel i7 5820k 4.8ghz 6 cores, NVIDIA GTX 980Ti,16GB RAM & one laptop Dell Precision i7 8850 2.59ghz 6 cores, 4 GB NVIDIA Quadro P1000, 32DB RAM

Member/Roles

1. **Konstantinos Synefakis**
    - Computer Engineering and Informatics
    - Core design Philosophy of CNN, adapting the pretrained models to our needs
2. **Mantziaris Nikolaos**
    - Business Administration - AUEB
    - Research, data ETL processes, presentation's design and authorship

Bibliography

http://www.sciencedirect.com/science/article/pii/S095741741630598X

http://ieeexplore.ieee.org/abstract/document/7543901/

https://arxiv.org/abs/1703.06870

https://github.com/matterport/Mask_RCNN

http://www.arxiv.org/abs/1512.03385

Timeplan

When we were thinking of that project, we couldn't realise of the difficulties we are going to face. We were not experienced enough in image processing and we had wrong opinion on how much data we were going to need in order to properly train the models we used.

Not only this, there were not similar projects or solutions on the web, except the model from COCO dataset, so we couldn't have access to more information. This thing, along with the fact of lack of time, made our job really difficult and we were re-planning our time again and again.

We could say that 60%-70% of our time was used for training and creating the model, while the rest of the time was used for research, studying on image processing and ETL of data.

Contact Person

Synefaki Konstantinos - (+30 698 1481295) - synefakis.k@gmail.com

Comments

We are not that content with the result. The resizing for the parking slots does not always work and also due to a lack of time we would like to be able to train the model with million more data.

We had serious issues finding a way to mark the spots and then classify them. Towards the end I came up with the solution otherwise we would just be able to classify if an image has a car parked or not. If we had more data and knowledge on how to handle images we could create a better performing classifier instead of relying on pretrained solutions.

Also the algorithm that tries to mark the space is kinda wrong since it recognizes some spots incorrectly .