TECHGIG : CODE GLADIATORS

# TEAM : AVIATO

Jio Artificial Intelligence Hackathon

# EXECUTIVE SUMMARY

## IDENTIFYING PARKING SPACES FOR SMART CITIES

We developed a 2 way automated system for empty parking space detection using complete lot description images and their sectioned individual space images that were either filled or empty. The binary sectional classifier demonstrated an accuracy of 99.97% thus depicting the generalizability of the model on subsequent new datasets from stationary camera feeds, whereas the empty space counter generated an accuracy of 80.83% on the test set.

# Problem At Hand

Based on a report from researchers at UCLA, people spend around 8 minutes on average searching for a parking spot. Identifiable parking lot spaces not only help environmentally (by helping people find parking spots faster), but also from a security perspective by preventing car thefts.

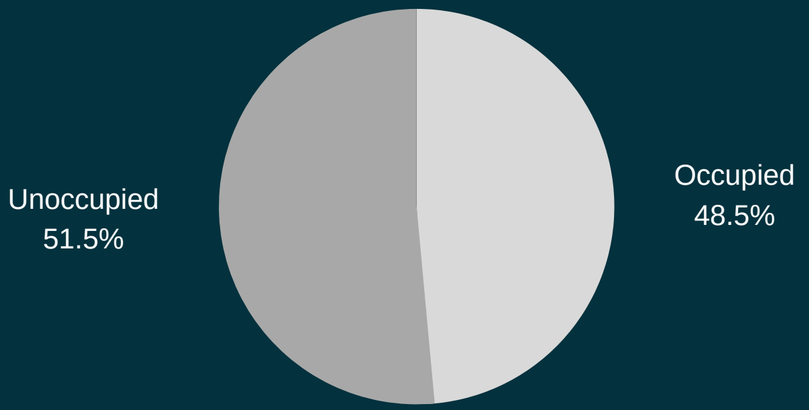
Hence, image based methods for identifying available spaces for parking are the only convenient solution, because it's alternative of installing tracking devices is fairly cost intensive. Using Deep Learning methodologies help us overcome this problem by building sustainable models for detecting parking spaces.

# APPROACH DESCRIPTION

- DATASET CREATION
- METHODOLOGY
- IMPLEMENTATION
- RESULTS

# DATASET COLLECTION

We used PKLot dataset (1) - A Robust Dataset for Parking Lot Classification. It contains 12,417 images of parking lots and 695,899 images of parking spaces segmented from them, which were manually checked and labeled. There were 3 zones of capture, namely UFPR04, UFPR05 and PUCPR. Each parking lot image is annotated with the date, time and current weather conditions (either sunny, cloudy, or rainy). Perspective correction has been applied on individual segments, and all are rotated to a vertical orientation.




SEGMENTED PARKING SPACES

(1) - http://web.inf.ufpr.br/vri/databases/parking-lot-database/

# METHODOLOGY

## 2 WAY MODEL - REGULAR CNN AND SPACE COUNTER

Regular CNN Model was used for bottom-up binary classification system, where we feed each individual parking space image to a CNN and it chooses whether that space is vacant or occupied

The Space Counter is a multinomial classifier that takes in an image of the entire parking area and returns the number of available parking spots in that image, up to a predetermined max value n.

## CNN MODEL DESCRIPTION

The Convolutional Neural Network uses a k*k filter also called a feature detector. The entire model has multiple convolutional layers each of which have multiple filters. The bottom-most layer receives the raw image, and the filters produces a 2D array with a dot product, which is called the activation map. The activation maps of all filters in a layer are then joined to form the activation volume. This output is now passed into an activation layer (ReLu or tanh) and further to a maxpool to scale down the activation volume, this now becomes the input for the next layer and the cycle goes on depending on the CNN size. Finally we feed into a fully connected layer and then into a softmax classifier to convert into class probabilities.

# IMPLEMENTATION

## TECHNIQUES

- Batch Normalization using mini-batches
- Adam update (Alternative for SGD) for allowing our models to achieve the same training loss in less training epochs.
- Saliency Maps to learn which spatial regions in the image contributed most heavily to that score.

## BINARY CLASSIFIER

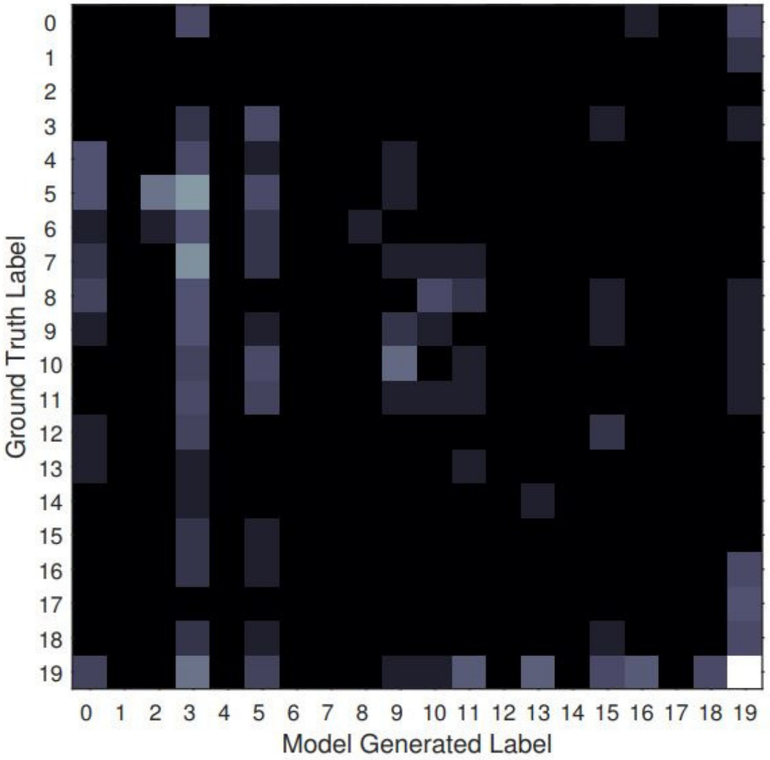
For binary classification model, we used 3 convolutional layers, each with a depth of 10, 20, and 30, respectively, and a filter size of 5x5. Each such layer was followed by a batch normalization layer and a ReLu activation function. After that, a max pooling layer with a filter size of 2x2 was added. The output of the convolutional layers was then fed into 3 fully-connected layers with 30, 20, and 10 neurons, respectively. Here each of the layers was directly fed into a ReLu activation. Finally, we used a Softmax layer to compute class scores. The images were resized to 48x64 before training. Furthermore, all the images were normalized so the mean of each color channel was 0.

## SPACE COUNTER

Architecture is similar to CNN apart from the input size and the class labels. Because it uses images of the entire parking lot, a much larger input size is required. There are now n + 1 labels where n is the predetermined maximum count. The images were resized to 256x128 pixels and normalized in the same way as the pre-segmented parking spots.

# RESULTS

- As expected, the binary classifier performed very well on the classification task and scored an accuracy of 0.9997 on the test set, even using a simple architecture with only 3 convolutional and fully-connected layers with a fairly small amount of neurons.
- It is also robust to overfitting, scoring over 99% for the training, validation and test set, and converges within one training epoch
- Test accuracy for Space Counter was 0.8084. The vast majority of the test data about 1,000 out of the 1,200 tested - was in the small box on the bottom right, which counts when the model guessed there were ”19 or more” empty spaces and was correct. In essence, the space counter is still quite accurate at noting when a parking lot has a large number of empty spaces and when it is almost full.




Comparison between the ground truth empty space counts and the counts. A lighter box indicates that the (ground truth, model choice) pair occurred more often.

# TECH STACK USED

## TORCH FOR LUAJIT

We used the Torch (http://torch.ch/) framework to build and train our CNNs and DeepMind's torch-hdf5 package to process the compressed HDF5 dataset

## PYTHON

For every other work, including compressiing data to HDF5, spot counting, cross validation, generalization and generating stats, we used Python 3. Python codes were powered by the NumPy library.

# Why is our team the best?

## EXPERIENCE

We are 2 junior undergraduates from IIT Kanpur, who have had experience in building AI models for solving cutting edge problems and also worked on applications of ML through research projects and competitions.

## COMPREHENSIVE APPROACH

We worked around solving both ends of the problem, classifying segmented individual spaces and counting empty spots from a total overview. This makes our approach comprehensive and solves the bigger picture

## ACCURATE PREDICTIONS

Our models have been proven to be generalizable and have shown a fair amount of accuracy for both cases, which work wel for easy integration and deployment for modern systems

## EASY DOCUMENTATION

Often great tech solutions lack readability and proper documentation for hassle free deployment. We have made sure you understand every piece of our approach and solution.

# ASSOCIATED FILE ATTACHMENT DESCRIPTION

## COMPLETE CODE WITH LUA AND PYTHON SCRIPTS

All the files required to reproduce the results, is in the code folder.

## README.TXT

Readme file.

# Thank You!

# References :


- Ioffe, Sergey and Christian Szegedy (2015). “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: CoRR abs/1502.03167. URL: http://arxiv.org/abs/ 1502.03167
- Kingma, Diederik and Jimmy Ba (2014). “Adam: A method for stochastic optimization”. In: arXiv preprint arXiv:1412.6980.
- Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton (2012). “ImageNet Classification with Deep Convolutional Neural Networks”. In: Advances in Neural Information Processing Systems 25. Ed. by F. Pereira et al. Curran Associates, Inc., pp. 1097–1105. URL: http : / / papers . nips . cc / paper / 4824 - imagenet - classification - with - deep - convolutional-neural-networks.pdf.
- Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman (2013). “Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps”. In: CoRR abs/1312.6034. URL: http://arxiv.org/ abs/1312.6034.
- The HDF Group (1997-). Hierarchical Data Format, version 5. http://www.hdfgroup.org/HDF5/.

# FUTURE ADDITIONS

”Segmentation Pipeline” for direct segmenting, rotation correcting and orienting  full scale stationary parking images to be sent into a binary classifier for a better overall predicitons

Using Satelite Imagery for better angles and more training and testing data

Using Mask-RNN for Parking Slot Slicing