# Parking Lot Vacancy Detector – Report

## 1. Creating training dataset

- Utilised the xml ElementTree parser for parsing the given xml file.

- Located the directory containing training images(.jpg) and corresponding xml file.

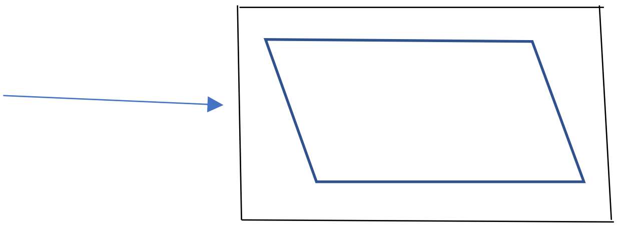
- Extracted the individual elements of the xml file like id , occupied status , individual coordinate point for the location of various cars in the form of (x,y) to generate positive training images.

- Generated positive training images by cropping the actual generated image based on the ground truth file

Example:

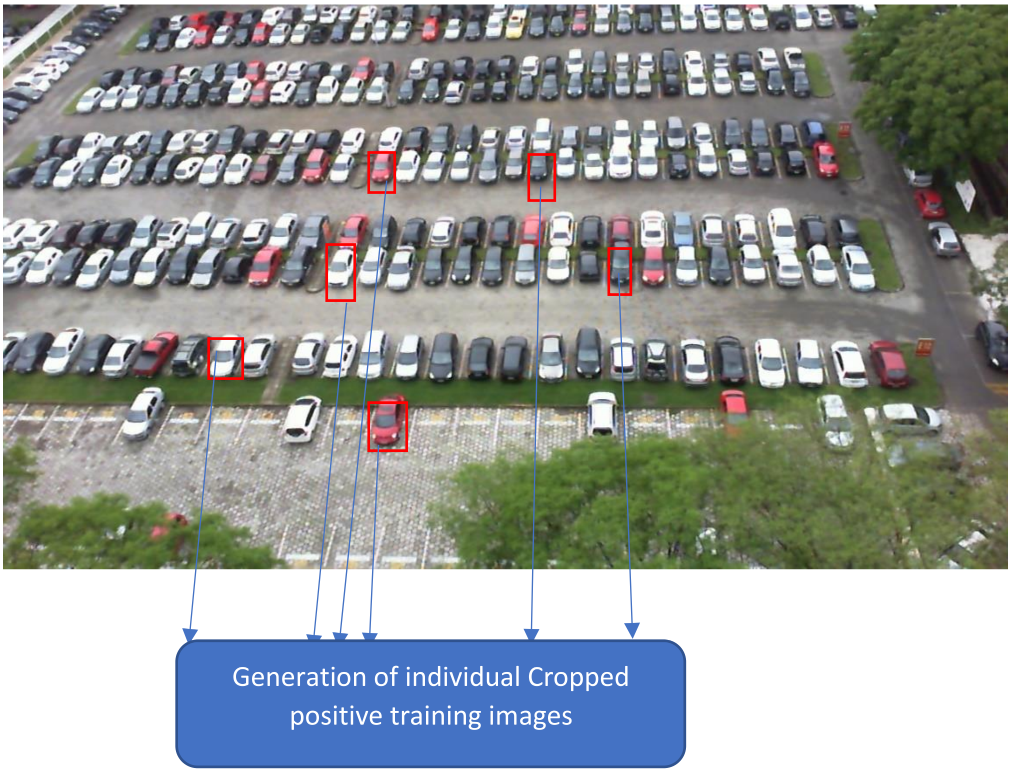
Coordinates from the ground truth file(*.xml)

```
<point x="278" y="230" />
<point x="290" y="186" />
<point x="324" y="185" />
<point x="308" y="230" />
```



- The cropping is done to the scale of 5 along x axis and to the scale of 10 along y axis

**crop_img = img[ymin-10:ymax+10, xmin-5:xmax+5]**



Generation of individual Cropped positive training images

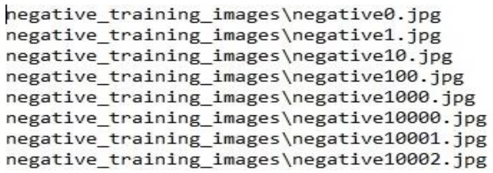
<u>Cropped Positive training images</u>: Total number = 1,84,000

- Downloaded the negative images dataset from the internet

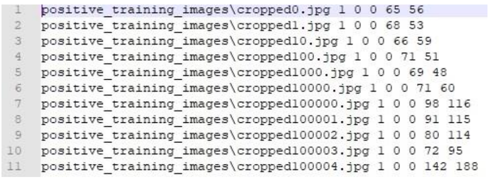
  <u>Negative training images</u>: Total number = 90,000

- Saved the cropped positive training images to a separate folder and the negative training images to a separate folder.

- Created 2 files namely “bg.txt” and “info.dat” containing the path of all the positive and negative training images.



```
negative_training_images\negative0.jpg
negative_training_images\negative1.jpg
negative_training_images\negative10.jpg
negative_training_images\negative100.jpg
negative_training_images\negative1000.jpg
negative_training_images\negative10000.jpg
negative_training_images\negative10001.jpg
negative_training_images\negative10002.jpg
```



```
positive_training_images\cropped0.jpg 1 0 0 65 56
positive_training_images\cropped1.jpg 1 0 0 68 53
positive_training_images\cropped10.jpg 1 0 0 66 59
positive_training_images\cropped100.jpg 1 0 0 71 51
positive_training_images\cropped1000.jpg 1 0 0 69 48
positive_training_images\cropped10000.jpg 1 0 0 71 60
positive_training_images\cropped100000.jpg 1 0 0 98 116
positive_training_images\cropped100001.jpg 1 0 0 91 115
positive_training_images\cropped100002.jpg 1 0 0 80 114
positive_training_images\cropped100003.jpg 1 0 0 72 95
positive_training_images\cropped100004.jpg 1 0 0 142 188
```

- Created *.vec file containing nearly 80,000 samples of positive images using the command:

  **opencv_createsamples -info info.dat -vec positive3.vec -num 80000 -w 24 -h 24**

## 2. Training the cascade classifier

- Now we train the classifier using the following command:

  **opencv_traincascade -data trial “folder” -vec pos.vec -bg bg.txt -numPos “ “ -numNeg “ “ -numStages “ “ -featureType LBP/HAAR**

  **-w 24 -h 24**

<u>Sample training scenarios</u>

A. Training Data – 1 :
   - Number of Positive samples – 45,000(initial value)
   - Number of Negative samples – 50,000
   - Number of stages – 10
   - Feature Type – LBP

B. Training Data – 2 :
   - Number of Positive samples – 2,700(initial value)
   - Number of Negative samples – 3,000
   - Number of stages – 5
   - Feature Type – HAAR

C. Training Data -3 :
   - Number of Positive samples – 67,500(initial value)
   - Number of Negative samples – 75,000
   - Number of stages – 10
   - Feature Type – LBP

D. Training Data-4 :
   - Number of Positive samples – 9,000(initial value)
   - Number of Negative samples – 10,000
   - Number of stages – 8
   - Feature Type – HAAR

## 3. Car detection and Accuracy

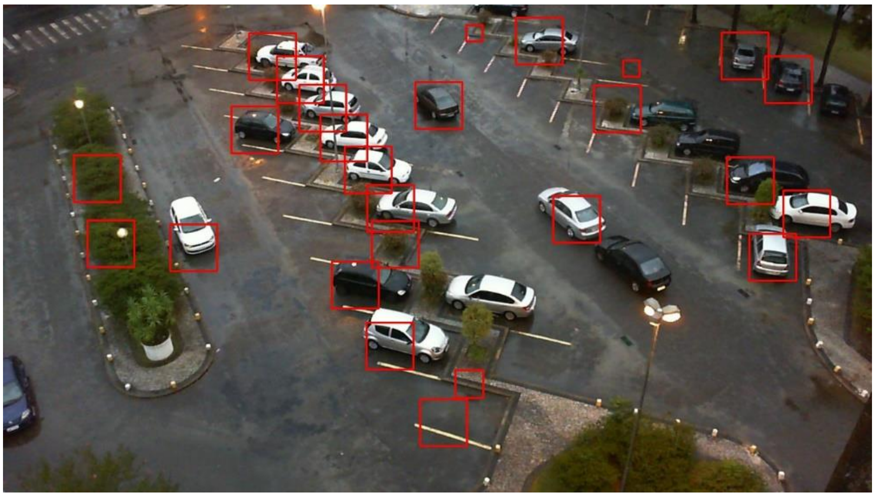
- Now after training phase, **cascade**(.xml) file is generated along with various stages as per the command
- Test images are the ones from rainy or cloudy folders as given before.
- Written a python file **testing.py** in which the testing image is loaded and passed to **CascadeClassifier::detectMultiScale()**
- Now the function **CascadeClassifier::detectMultiScale()** contains parameters like:
- **Note**: ALL parameters used are subject to each individual image.
    - I. **scaleFactor -** tried between 1.1 and 1.9
    - II. **minNeighbors -** tried from 6 to 20
    - III. **minSize =** (20,20)
    - IV. **maxSize =** (75,75)
- For detecting cars, I have retrieved the corresponding xml file of the passed testing image
- The function **CascadeClassifier::detectMultiScale()** returns a list of 4 elements – [x,y,w,h]

- Initially extracted the “X” , “Y” points from the corresponding “.xml” file
- Now generated 4 values Xmin , Xmax , Ymin , Ymax from a pre defined list containing above “X” and “Y”
- Calculated the area of bounding rectangle from xml file and also the area of detected rectangle.
- In order to associate these 2 rectangles for detection , used the midpoint of bounded rectangle.
- Now initialised a **threshold** value(default=60) and checked whether the ratio of the above calculated areas falls above the threshold.
- If yes, then it is a True Positive and else it is a False positive.

**Accuracy** = (TP + TN) / (TP + FP + TN + FN)

## 4. Results and Parking lot analysis

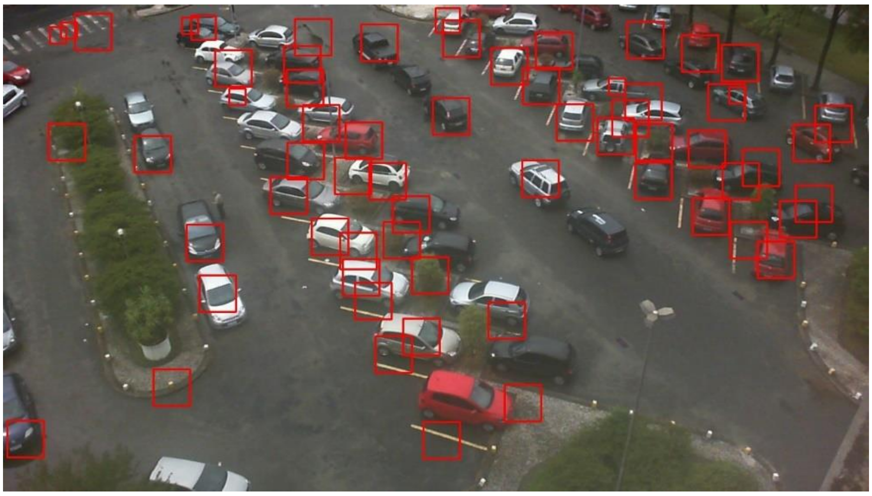
I. Pos_samples = 45k , neg_samples =50k , LBP , stages = 10



Number of cars detected : 17

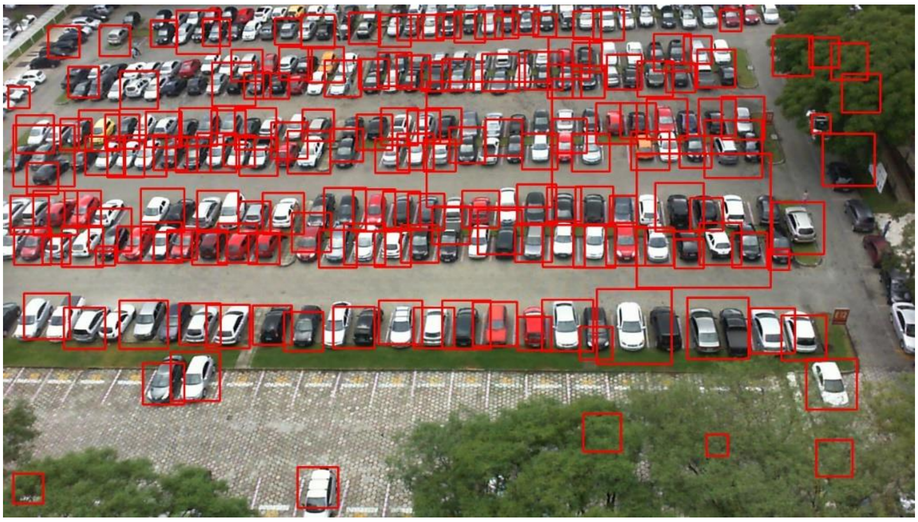
Number of vacant spots detected : 23

II. Pos_samples = 2.7k , neg_samples = 3k , HAAR , stages =5



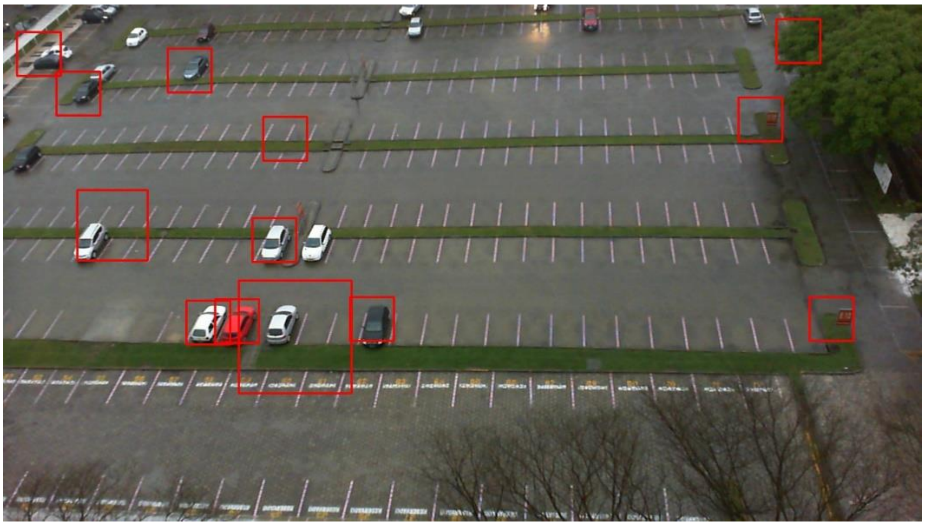
Number of cars detected : 26
Number of vacant spaces detected : 14

III. Pos_samples = 67.5k , neg_samples = 75k , LBP , stages =10



Number of cars detected : 81
Number of vacant spaces detected : 19

IV. Pos_samples = 9k , neg_samples = 10k , HAAR , stages =8



Number of cars detected : 10

Number of vacant spaces detected : 88

| Test Image | Classifier Feature | Training | | | TP | FP | Accuracy |
|---|---|---|---|---|---|---|---|
| | | Stages | No. of Positives | No. of Negatives | | | |
| 2013-04-12_17_55_13.jpg | LBP | 10 | 45k | 50k | 17 | 23 | 85.0 |
| 2013-02-26_13_04_33.jpg | HAAR | 5 | 2.7k | 3k | 26 | 0 | 65.0 |
| 2012-11-10_11_12_51.jpg | LBP | 10 | 67.5k | 75k | 81 | 19 | 93.0 |
| 2012-09-21_07_05_12.jpg | HAAR | 8 | 9k | 10k | 12 | 16 | 92.0 |
| 2012-12-12_10_00_05.jpg | HAAR | 5 | 2.7k | 3k | 18 | 0 | 64.2 |
| 2012-12-08_18_25_14.jpg | HAAR | 8 | 9k | 10k | 0 | 6 | 78.5 |
| 2012-12-22_12_00_08.jpg | HAAR | 5 | 2.7k | 3k | 1 | 4 | 85.7 |
| 2012-12-22_12_00_08.jpg | HAAR | 8 | 9k | 10k | 1 | 5 | 82.2 |
| 2013-01-18_17_05_13.jpg | HAAR | 5 | 2.7k | 3k | 4 | 4 | 82.0 |
| 2013-01-18_17_05_13.jpg | HAAR | 8 | 9k | 10k | 0 | 0 | 82.0 |
| 2013-03-05_08_20_02.jpg | HAAR | 5 | 2.7k | 3k | 24 | 0 | 60.0 |
| 2013-03-05_08_20_02.jpg | HAAR | 8 | 9k | 10k | 29 | 0 | 72.5 |
| 2013-03-19_07_25_01.jpg | HAAR | 5 | 2.7k | 3k | 18 | 8 | 65.0 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2013-03-19_07_25_01.jpg | HAAR | 8 | 9k | 10k | 16 | 2 | 75.0 |
| 2012-12-17_09_50_05.jpg | HAAR | 5 | 2.7k | 3k | 17 | 0 | 60.7 |
| 2012-12-14_09_20_04.jpg | HAAR | 8 | 9k | 10k | 21 | 0 | 75.0 |
| 2012-12-08_18_25_14.jpg | HAAR | 5 | 2.7k | 3k | 0 | 0 | 100.0 |
| 2012-12-08_19_30_16.jpg | HAAR | 8 | 9k | 10k | 0 | 0 | 100.0 |
| 2013-01-16_09_05_04.jpg | HAAR | 5 | 2.7k | 3k | 1 | 3 | 85.7 |
| 2013-01-16_11_15_07.jpg | HAAR | 8 | 9k | 10k | 9 | 6 | 75.0 |
| 2013-02-26_13_19_33.jpg | HAAR | 5 | 2.7k | 3k | 23 | 0 | 60.0 |
| 2013-02-26_13_19_33.jpg | HAAR | 8 | 9k | 10k | 33 | 0 | 85.0 |
| 2012-09-12_10_11_12.jpg | HAAR | 5 | 2.7k | 3k | 59 | 4 | 60.0 |
| 2012-09-12_10_16_27.jpg | HAAR | 8 | 9k | 10k | 82 | 0 | 87.0 |
| 2012-10-11_07_36_48.jpg | HAAR | 5 | 2.7k | 3k | 12 | 14 | 74.0 |
| 2012-10-11_07_41_49.jpg | HAAR | 8 | 9k | 10k | 21 | 6 | 80.0 |
| 2012-11-11_12_49_03.jpg | HAAR | 8 | 9k | 10k | 3 | 4 | 96.0 |
| 2012-11-11_12_54_03.jpg | LBP | 10 | 45k | 50k | 4 | 1 | 99.0 |
| 2012-11-11_12_44_02.jpg | LBP | 10 | 67.5k | 75k | 2 | 3 | 97.0 |
| 2013-01-17_07_50_03.jpg | LBP | 10 | 45k | 50k | 0 | 1 | 96.4 |
| 2013-01-17_07_55_03.jpg | LBP | 10 | 67.5k | 75k | 0 | 2 | 92.8 |
| 2012-11-08_07_05_26.jpg | LBP | 10 | 45k | 50k | 8 | 4 | 96.0 |
| 2012-12-07_16_42_25.jpg | LBP | 10 | 67.5k | 75k | 24 | 0 | 92.8 |
| 2012-12-15_07_30_02.jpg | LBP | 10 | 45k | 50k | 0 | 2 | 92.2 |
| 2012-12-15_07_35_02.jpg | LBP | 10 | 67.5k | 75k | 0 | 1 | 96.4 |
| 2012-10-26_06_59_26.jpg | LBP | 10 | 45k | 50k | 4 | 0 | 93.0 |
| 2012-10-26_07_04_26.jpg | LBP | 10 | 67.5k | 75k | 13 | 0 | 99.0 |
| 2012-10-26_09_14_33.jpg | LBP | 10 | 45k | 50k | 95 | 0 | 99.0 |
| 2012-10-26_09_19_33.jpg | LBP | 10 | 67.5k | 75k | 86 | 0 | 90.0 |
| 2012-10-28_06_31_45.jpg | LBP | 10 | 45k | 50k | 1 | 0 | 100.0 |
| 2012-10-28_06_36_45.jpg | LBP | 10 | 67.5k | 75k | 1 | 0 | 100.0 |
| 2012-09-28_17_26_29.jpg | LBP | 10 | 45k | 50k | 53 | 0 | 90.0 |
| 2012-09-28_17_31_29.jpg | LBP | 10 | 67.5k | 75k | 49 | 8 | 80.0 |
| 2012-09-28_17_56_30.jpg | LBP | 10 | 45k | 50k | 31 | 2 | 81.0 |
| 2012-09-28_18_01_30.jpg | LBP | 10 | 67.5k | 75k | 46 | 1 | 97.0 |
| 2012-11-09_11_56_47.jpg | LBP | 10 | 67.5k | 75k | 56 | 2 | 88.0 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2012-11-09_12_06_47.jpg | LBP | 10 | 7.2k | 8k | 39 | 1 | 79.0 |
| 2013-01-22_12_40_08.jpg | LBP | 10 | 7.2k | 8k | 21 | 3 | 78.5 |
| 2013-03-13_13_05_08.jpg | LBP | 10 | 7.2k | 8k | 27 | 1 | 70.0 |
| 2012-11-10_11_12_51.jpg | LBP | 10 | 7.2k | 8k | 63 | 0 | 75.0 |