AUTOMATIC NUMBER PLATE RECOGNITION

Introduction:

Automatic Number plate recognition is the process of recognising the number plate of the vehicles from the images of the vehicles.This problem contains 3 sub-problems:

1. Number plate recognition: This is the problem of recognising the number plates of the vehicles from an image.
2. Character segmentation from the Number plates: This corresponds to the extraction of characters from the number plates.
3. Recognition of the characters: This is the problem of recognising the characters from which were extracted from the number plate in the previous part.

Number plate recognition:

The approach here is based on the fact that the number of horizontal and vertical lines are higher across the breadth of the number plate than that of any other region of the car. This contains 2 sub problems:

1.Band clipping :

- To detect the edges in the number plate, Sobel operator is used. After detecting the edges in the image, vertical projection of the edges of the image is used for detection of the regions with more edges in the image across the y-axis.
- The following algorithm is used for detecting the regions with more number of edges across the y-axis from vertical projection of the edges in the image.
- Algorithm:

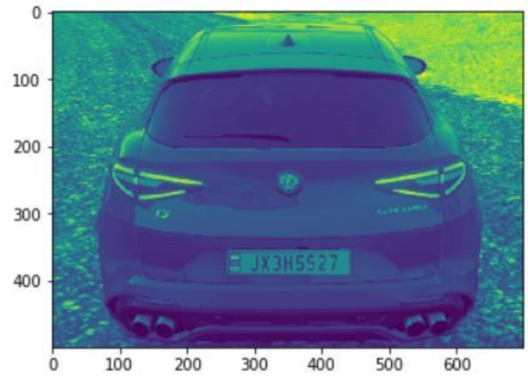
```
1.Get the maximum vertical projection value(be m).
2.If the current maximum vertical projection value(be at ym) is greater than a multiplied by m continue to step 3 , else go to step 5.
3.Get the maximum index of the vertical projection value from 0 to ym which is less than b*current maximum vertical projection(be y1).
4.Get the minimum index of the vertical projection value from ym which is less than b*current maximum vertical projection(be y2).
5.Store the values of y1 and y2.
6.Nullify the values of the vertical projection across y1 and y2.Go to step 2.
7.End of the algorithm.
Note: a and b are given constants.
```

- By using the above algorithm we get the candidate upper and lower boundaries of the number plate called as horizontal bands.

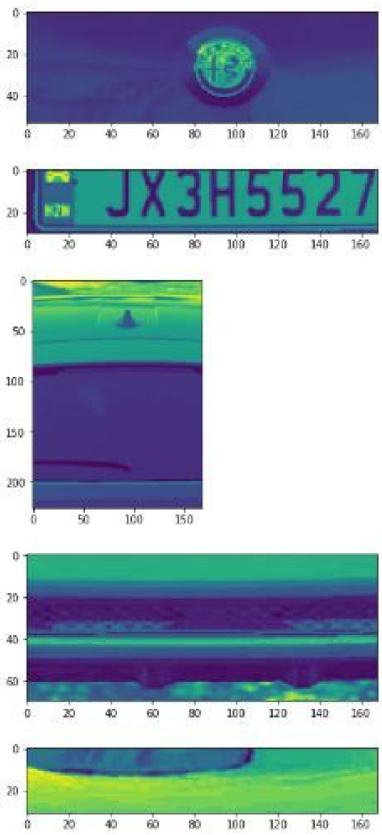
2. Plate clipping :

- For each such obtained band, we take horizontal projection of the band and use a similar algorithm on vertical coordinates with different constants to get the possible left and right boundaries of the number plate called as vertical bands.
- If we combine the horizontal boundaries along with their corresponding vertical boundaries, we get the possible number plate candidates.
- After this, we do segmentation on each of these candidate plates.

Image of the car:



Candidate Plates after band clipping and plate clipping :



Character Segmentation of Number Plates:

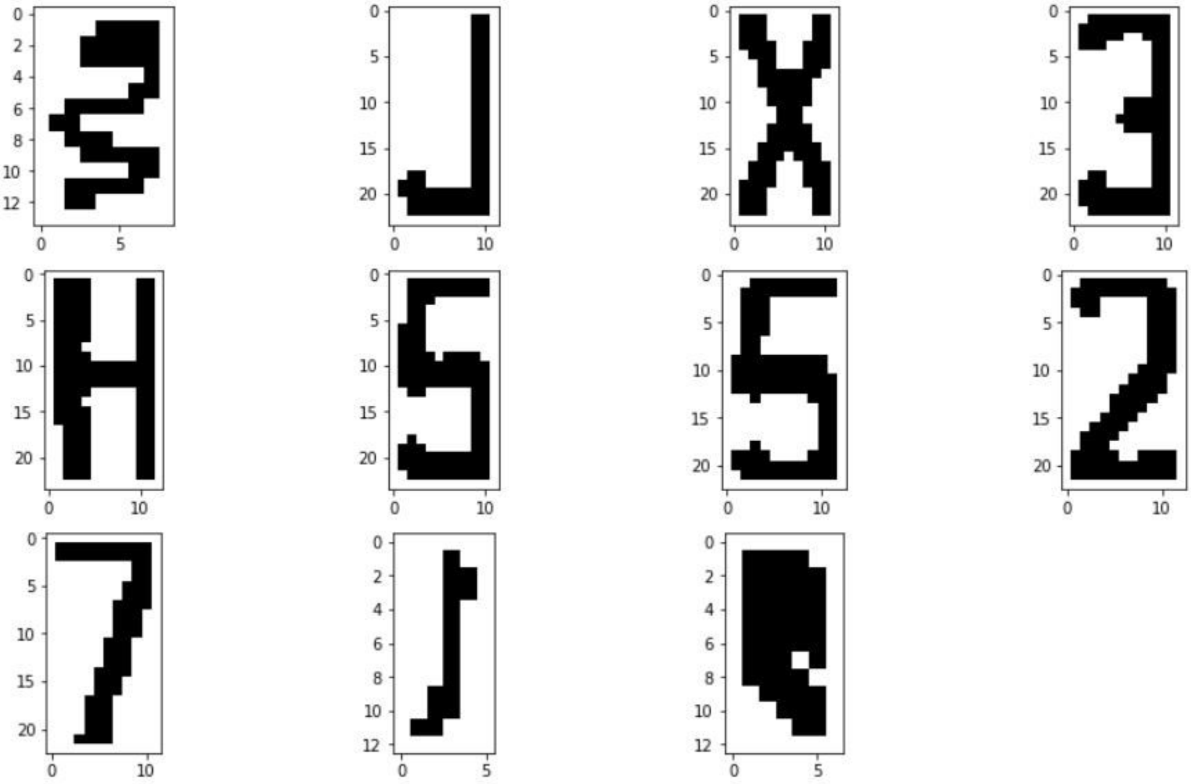
- After the above method is employed,we are left with several candidate plates. Now each of these candidate plates need to be segmented.
- The candidate plate contains spaces between the characters and this observation is exploited for segmentation. So find the horizontal projection on the candidate plate. Adaptive thresholding is used to enhance the area of the plate before segmentation. This results in plate having only black and white pixels. So when the horizontal projection is obtained, the peaks in the graph correspond to the spaces between the characters as there is a sharp transition in color between the character and the space after. So it should be sufficient to iterate over the peaks in the horizontal projection graph to obtain the left and right boundaries of the spaces. With this information, the coordinates of the possible segments can be computed for that respective candidate plate.
- The segments obtained from the above process need to be pruned as it can contain undesirable elements along with the necessary character. So this segment has to split into multiple pieces. This can be achieved using the seed-fill algorithm. This is described as follows:

  Flood Fill:

  1. Consider the set S which contains the coordinates of all black pixels.Let A be an auxiliary set for computation and P be the final set which contains the coordinates of a particular element in the given segment.Let PieceList be a list of isolated elements from the segment
  2. Now pick a cell from S.Add this to the auxiliary set A.
  3. Now pop this from the set.As long as this set remains non empty ,keep adding neighbours of this cell provided the following two conditions is satisfied:
     - ➢ The neighbouring cells must belong to S,i.e they must be black cells.
     - ➢ These cells must not belong to P,i.e they must be visited for the first time.
  4. If the above conditions are satisfied,then the neighbouring cell will be added to the set A and will also be removed from the set S.
  5. The original cell is added to P and the algorithm returns to step 3.If the set A becomes empty then move to the next step.
  6. P is appended to PieceList and the algorithm returns to step 2.
  7. If the set S becomes empty, then the algorithm terminates.
- After we obtain all the components from the candidate plates by doing flood fill on each of the plates, we eliminate several components as non-characters and non-numbers based on certain heuristics given in the research paper. When we tried using these heuristics for eliminating non-character components, we found that many of the components which contain a character or a number are getting eliminated. This could possibly be due to the strictness of the heuristics.So we tweaked the heuristics and concluded to use the below heuristics:

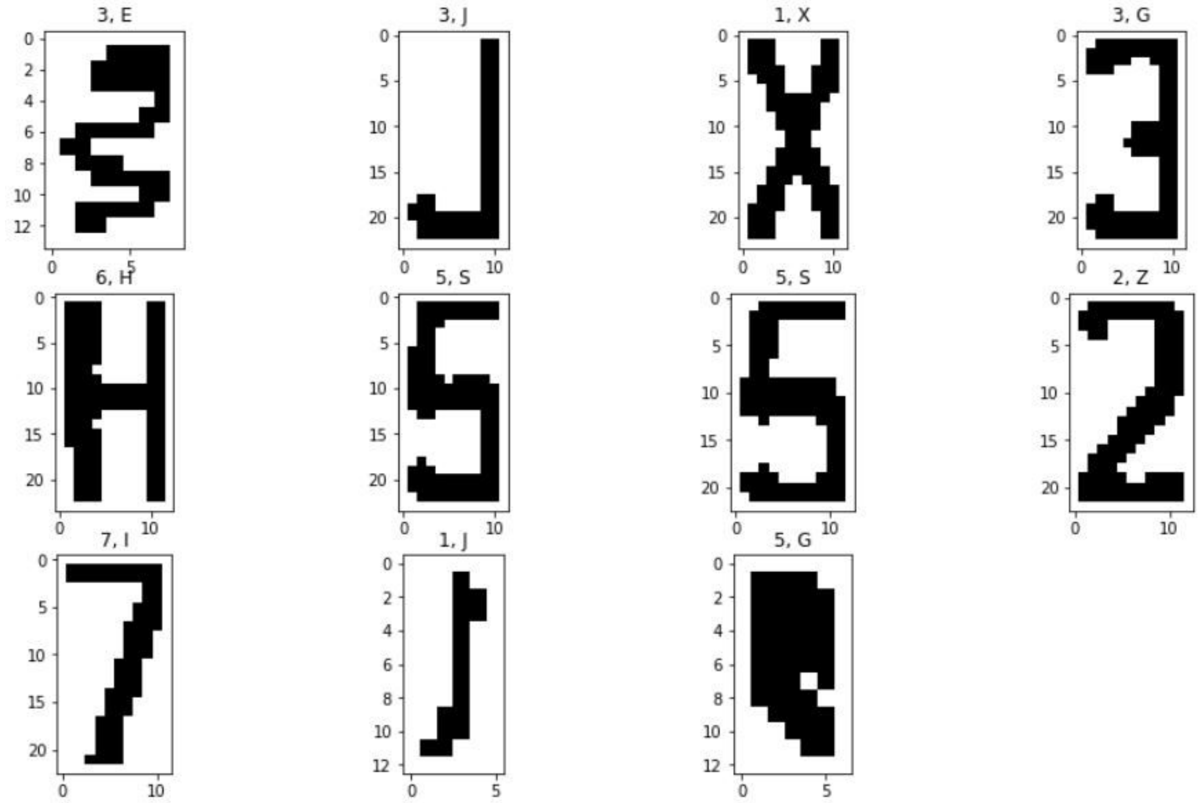
- First we considered the ratio of black to total no of cells in an image.If this ratio lies between 0.2 and 0.8, then this heuristic accepts the image.
- Next the height of the image should be at least 60% of the height of the number plate to which it belongs.
- After the above heuristics are applied, a list of candidate characters will be obtained.The count of these characters is computed and the number plate which gives the maximum number of characters is selected.

Components recognized as possible characters:

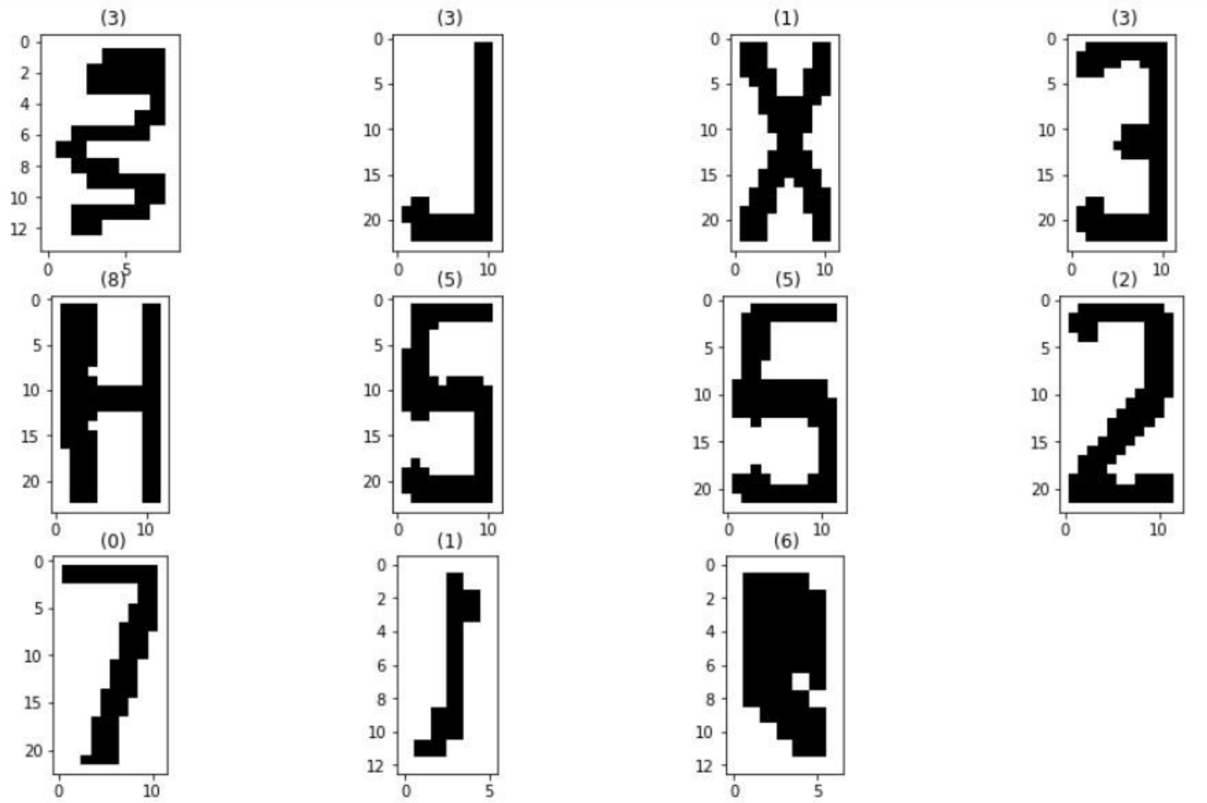


Recognition of Characters:

- We trained two different models for recognizing alphabets and numbers using char74K dataset.
- Each character is predicted using both the models and the ground truth values are compared to figure out if the character is more likely to be a number or an alphabet.



- Using a single model for both alphabets and numbers resulted in the following output which is not as good as the predictions made by separate models.


(3)
(3)
(1)
(3)
(8)
(5)
(5)
(2)
(0)
(1)
(6)