**AUTOMATIC NUMBER PLATE RECOGNITION**

*The technique used in this project is template matching technique to identify the segmented characters by comparing them against most accurate templates of different fonts.This process comes under the division of (OCR)optical character recognition.*

***LANGUAGES USED:***

*1)We used Python language to read the input image and performed number plate recognition followed by character segmentation.*

*2)Non-Synthesized Verilog language to perform optical character recognition and print the final number plate string in a output text file.*

***TECHNIQUE:***

*Input image is read into python program to localize number plate from the given image and it was followed by segmentation of characters. The segments which contains characters are resized to 32X 32 pixels and these were converted into matrices and saved into text files.*

*These text files which containing character matrices were matched against standard character templates and the most accurately matched character is written into output file.*

*Hence these output file contains the number plate string of given image.*

***Summary****: Project is done in two parts one by using Verilog and other by using python.*

*• In python image of car containing number plate is taken as input and is made as segments of characters.*

*• For Verilog these segments are taken as input and output is character string on the number plate.*

*• These segments are individually compared to different templates and most matched one will be taken.*

*• These templates are different characters of different font but of same pixels.*

*• Comparison of these templates and displaying the string of characters on the number plate is done using Verilog.*

***Summary for Verilog part:***

*Segments of characters sized to 32by 32 are taken as input to Verilog(pixel values will be in binary form). These are store in a file and are taken as input to Verilog by reading that file. We use file handling to read that file (i.e functions like $readmemb)and these will be compared with templates of all different alphabets and digits(36 i.e A to Z and 0 to 9 digits)of different fonts(comparison can be done by using xnor for pixel to pixel) and numbers from 0 to 9 of different fonts (template matching technique) and the most suited one will be printed and continues this process until all the segmented files are read.*

*This string will be displayed and stored in output file.(f\_o.txt).*

***MY CONTRIBUTION-****(25%)*

*1)Preprocessing: All the python part including number plate localization, character segmentation, converting image files to binary text files.*

*2)File handling after asked for modification: I modified the entire code’s file handling part by replacing all the file handling functions fopen and fscanf with the readmemb function inorder to increase the efficiency of file reading and writing, since readmemb don’t need any looping constraints to read the entire data of input file.*

*3)I wrote the Verilog code to input all the character segments automatically and also modified the code for comparing multiple characters and write the number plate string data in an RECOGNITION OF NUMBER PLATE*