Abstract—This paper examines a method of vehicle logo detection through edge detection or adaptive thresholding to determine the license plate location which is then used to segment an area above the plate to look for the vehicle logo. The logo location is then correlated with the model logo binary images to determine what the make of the car is. All of the computing was done in Matlab.

Keywords— Vehicle logo detection; Vehicle logo recognition; Edge detection; Adaptive Threshold; Cross-correlation; Normalized cross-correlation

I. INTRODUCTION

Vehicle logo recognition has been a topic researched in computer vision for quite some time because of the benefit to the transportation monitoring system set up along the road. The recognition allows the classification of cars on the road which can be used for surveying purposes or vehicle tracking. This has a benefit for police when looking for a suspect in a vehicle the cameras on the roads can detect possible cars that can be investigated.

The recognition system proposed in this paper is only for the make of the car and must be one of the given six brands: Aston Martin, Audi, Chevrolet, Honda, Mercedes, and Toyota. Both front and back images were looked at in cars. The method proposed uses license plate recognition to segment the original image into a smaller region which contains the logo of the car. After that, the smaller region uses edge detection or adaptive thresholding to create a binary mask of what the logo looks like on the car to compare to the ideal logos. Normalized cross-correlation was used to determine the car make from the two binary images of the logo and see which of the six models the image correlated highest to.

The paper is organized as follows. Section II discusses previous works and how they influenced this project. Section III then lays out the method used in this program to determine the make of the car as well as methods that were changed throughout the development of this project to improve accuracy. Results and Analysis are discussed in section IV and the paper is concluded in section V.

II. RELATED RESEARCH

There were three main papers that were used in the development of this piece that will be discussed.

A. A Vehicle-Logo Location Approach Based on Edge Detection and Projection by Yang Liu and Shutao Li of Hunan University, 2011. [1]

This paper was used to help determine the method of segmentation to find the logo in the original image. If the original image were to be edge detected and binarized then correlated, the exact size of the logo would have to be known to get a good reading. This would be impossible to tell without prior knowledge of the logo so a smaller region is needed to find the logo. This paper discussed many methods but the one used was license plate detection from which an area above the plate is segmented and correlated against.

B. An Efficient Method For Vehicle Model Identification Via Logo Recognition by Huihua Yang, Lei Zhai, Zhenbing Liu, Lingqiao Li, Yichen Luo, Yong Wang, Haiyan Lai, Ming Guan of Guilin University, 2011. [4]

This paper was used to help determine the method of segmentation to find the logo in the original image. If the original image were to be edge detected and binarized then correlated, the exact size of the logo would have to be known to get a good reading. This would be impossible to tell without prior knowledge of the logo so a smaller region is needed to find the logo. This paper discussed many methods but the one used was license plate detection from which an area above the plate is segmented and correlated against.

C. An Efficient Features-Based License Plate Localization Method by Hamid Mahini, Shohreh Kasaei of Sharif University of Technology and Faezeh Corri and Fatemeh Dorri of Amirkabir University of Technology, 2006. [2]

Plate localization is very important in this project because it is the basis of logo location which is then correlated for
recognition. This paper discussed the operations that could be used for license plate detection such as extracting the edges and blurring the image to find gray areas that could correspond to license plate locations that also have light backgrounds. This method is very effective if done correctly, on the premise that the license plate is light colored and contrasts well with the black writing on top. The method used in this project was vertical and horizontal line detection to find areas that could represent the license plate, but this method could be used to further increase results for those that are not perfectly straight on with the license plate. The part used from this experiment was determining which region was most likely the license plate because of the ratio to height and width and the area that is contained by the region detected.

III. METHOD
Some morphological operations used in this procedure are:

- **Opening**: Erosion then Dilation.
- **Closing**: Dilation then Erosion.
- **Sobel Edge Detection**: Convolution of the image with 2 edge operators to find gradients in the x and y direction, then finding the magnitude of the two operations to give the overall edge features.

A. Logo Location

As discussed earlier, the first step for logo location in this project was to determine the location of the license plate. This was done by first applying edge detection with Sobel operators to determine the most significant edges in the piece as that will most likely correspond to the license plate edges. The resulting binary image will then have horizontal and vertical opening and dilation done simultaneously then added back together to find square or rectangular shapes in the piece. These regions are then found by doing connected component extraction, or a boundary detection algorithm, and determining which region is most likely the license plate of the piece through width to height ratio and area as compared to the entire image. The desired width to height ratio was around 1.5 to 6 to include European license plates and an area that was less than half and greater than a couple thousandths of the original image.

Once this area is detected, the region above the license plate is segmented out and said to have the logo inside, as that is the location for most car logos. The region segmented was:

$$[X_1, X_2] U [Y_1, Y_2]$$

$$X_1 = \text{First column of plate location}$$
$$X_2 = \text{Last column of plate location}$$
$$Y_1 = \text{First row of plate location} - 1.6*(\text{Height of plate})$$
$$Y_2 = \text{First row of plate location} - 0.2*(\text{Height of plate})$$

If the Sobel operators were not successful in finding the license plate, a secondary operation was done to give higher detection due to shadows and low lighting that could affect the prominence of edges. This was done with adaptive thresholding with a window size of 15 and a local threshold of the mean of the local – 0.05 to find the edges and features of the piece. This was then put through the same process above to find the license plate region and logo location.

B. Logo Recognition

The first step to logo recognition was to create ideal car make model logos that were binarized. These will be correlated against to find the make of the original image. This was done by edge detection on an image of only the car maker’s logo. A threshold value was then imposed to generate a binary image. An opening operation is then done to erase noise pixels and the logo was filled in through closing operations. This was then scaled to fit the region that would most likely be segmented out when a license plate is found.

For the region found in part A, the edges are found using the method to get the license plate region, i.e. Sobel operators or adaptive thresholding. This region is then opened to erase small pixel dots around the entire image and then the larger portions are filled with closing operations which should fill and leave behind the large logo areas.

One of the common problems with logo detection is the glare of the car paint or chrome features / lights in the same region as the detected logo. This is countered by opening with a large horizontal line structural element and subtracting that from the segmented image. This will create a binary image without long horizontal lines as those are most likely not part of the logo needing to be correlated.

The last part of logo recognition used normalized cross-correlation. At first cross-correlation was used and yielded good results for 2 or 3 car logos as there were some distinct differences. This algorithm is defined by the element-wise multiplication of a mask matrix and the original image as seen in Figure 1 when taking the gray area of the lower matrix and multiplying it with the masking 12 matrix:

This was done by forcing the model logo to be the same size as the logo region found and zero padding the found region to correlate the pictures at every pixel they are both defined.
Although this worked for a smaller sample size of makes, this needed to change to normalized cross correlation to find how much of the images actually correspond not just the number of pixels that are both white.

The equation for normalized cross-correlation was the value found by cross-correlating divided by the total number of white pixels in both the mask and the same size region in the zero padded logo region. The equation is:

\[
\frac{\text{Cross Correlation value}}{\sqrt{\text{dot}(\text{segmented image}) \times \text{dot}(\text{mask})}}
\]   

(2)

Where dot computes the element-by-element dot product of the desired image with itself to find the value of the pixels in that image. The denominator gives the entire value of pixels in both the mask and logo region added together.

The other method attempted but in the end not used was SIFT operations for logo classification. [2] The reason this was not used is because there were many false alarms with chrome features of cars for corners and the fact that some logos, like Audi, have very few corners so detection is difficult.

IV. RESULTS AND ANALYSIS

To evaluate the proposed method, 80 images were tested against the premade car logo models to see which of the six makes they were. The experiments were done on an Intel i7 processor 8 core CPU at 3.5 GHz, 16 GB RAM, and Matlab 8.1 with parallel computing on 4 cores to decrease the computation time through self-created functions.

The output of the function gives an image with the license plate in red and the logo region in green; an image of the segmented image to be correlated; a pop-up saying the make; and the correlation output in the command window of the six car makes.

The boundary identifying code used in Matlab replaced the connected component extraction because of the time it saves – 0.5 seconds versus 5 minutes – and that it closed some of the boundaries if they were close to touching and formed some regions which increased accuracy of detection.

The other operation used was a self-created cross-correlation and normalized cross-correlation function. This was much slower than the built in functions: 20 seconds versus 0.1 seconds for cross-correlation and 50 seconds versus 1 second for normalized cross-correlation. This time constraint was cut to 3 seconds and 20 seconds respectively by operating on four cores simultaneously with Matlab.

The results from the lab were shown to be relatively reliable. A sample result is seen in Figure 2 where a is the image with found regions, b is the binary logo region, and c is the message that tells the user which make the car is. Figure 2 had normalized cross-correlation values of 0.4251 for Audi, 0.2792 for Aston Martin, 0.2976 for Mercedes, 0.3215 for Honda, 0.3666 for Toyota, and 0.2657 for Chevrolet hence the message saying “This car is an Audi”.

Table 1 shows the effectiveness of this method for determining the different steps such as license plate detection and normalized cross-correlation versus cross-correlation.

![Figure 1. Cross-Correlation](image1.png)

![Figure 2. Sample Output for Logo Recognition](image2.png)

**TABLE 1:**

<table>
<thead>
<tr>
<th></th>
<th>Normalized Cross-Correlation</th>
<th>Cross-Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct Recognition</td>
<td>47</td>
<td>35</td>
</tr>
<tr>
<td>License Plate Detection</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>Recognition after correct detection</td>
<td>47/64</td>
<td>35/64</td>
</tr>
</tbody>
</table>

The boundary identifying code used in Matlab replaced the connected component extraction because of the time it saves – 0.5 seconds versus 5 minutes – and that it closed some of the boundaries if they were close to touching and formed some regions which increased accuracy of detection.

The other operation used was a self-created cross-correlation and normalized cross-correlation function. This was much slower than the built in functions: 20 seconds versus 0.1 seconds for cross-correlation and 50 seconds versus 1 second for normalized cross-correlation. This time constraint was cut to 3 seconds and 20 seconds respectively by operating on four cores simultaneously with Matlab.

The results from the lab were shown to be relatively reliable. A sample result is seen in Figure 2 where a is the image with found regions, b is the binary logo region, and c is the message that tells the user which make the car is. Figure 2 had normalized cross-correlation values of 0.4251 for Audi, 0.2792 for Aston Martin, 0.2976 for Mercedes, 0.3215 for Honda, 0.3666 for Toyota, and 0.2657 for Chevrolet hence the message saying “This car is an Audi”.

Table 1 shows the effectiveness of this method for determining the different steps such as license plate detection and normalized cross-correlation versus cross-correlation.
TABLE 2:

<table>
<thead>
<tr>
<th>Results After Correct License Plate Detection</th>
<th>Normalized Cross-Correlation</th>
<th>Cross-Correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Audi</td>
<td>95%</td>
<td>95%</td>
</tr>
<tr>
<td>Honda</td>
<td>75%</td>
<td>55%</td>
</tr>
<tr>
<td>Toyota</td>
<td>50%</td>
<td>25%</td>
</tr>
<tr>
<td>Chevrolet</td>
<td>70%</td>
<td>50%</td>
</tr>
<tr>
<td>Mercedes</td>
<td>90%</td>
<td>40%</td>
</tr>
<tr>
<td>Aston Martin</td>
<td>50%</td>
<td>25%</td>
</tr>
</tbody>
</table>

The effectiveness of each brand of car is seen in Table 2 for both normalized cross-correlation and cross-correlation. The sample size for some brands like Aston Martin were smaller than the others because the others were common in everyday and pictures were taken in parking lots to get more data to run the program on. The most recognized were Audi and Mercedes. This is because Audi is so distinct from the others but Mercedes is surprising because the Toyota symbol is also a circular shape.

The false alarm rate for Aston Martin was higher in the normalized cross-correlation because it had the least number of pixels in the model logo. This forced the results to say Aston Martin if very few pixels correlated. For cross-correlation, Audi had the largest false alarm rate because its logo had the greatest number of pixels so it was most likely to correlate with more if there were extra features detected. Chevrolet and Aston Martin looked most similar to one another and those triggered for one another most often. This shows that the larger the sample size of car logos the less effective this correlation method will become.

Appendix A shows the Logo models that were correlated against. Appendix B has more results of the car images and Appendix C contains some failures that gave the wrong car.

V. CONCLUSION AND FUTURE WORK

The performance of the code was not ideal but worked over 50% of the time to get the right make of car. If the license plate is detected the performance increased greatly as 80% of the images detected the license plate, and 75% using normalized cross-correlation from the detected images were correctly recognized. The best possible conditions are a close up image with the license plate flat and little glare/shadows on the car. This should be able to detect accurately the make in these conditions.

The faults of the program are low light situations, shadows across the license plate, logo not in segmented region, too low resolution image so logo has very few pixels, and car is some distance away in the picture. Another fault is that if any image of a car not in the set it run through it will pick the most correlated logo and say it is that car which is false. These could be fixed with extensions to the project but this was all that could be done in time of the project.

Further extensions could be adding feature operators like SIFT and getting them to increase performance, or adding Model recognition to cars so the Make and Model are both found.

VI. ACKNOWLEDGMENTS

This would not be possible without the help of Professor Mohan Trivedi, Naresh Kumar Modhipalli, and Alfredo Ramirez.

REFERENCES

[1] S. Li, Y. Liu, “A vehicle-logo location approach based on edge detection and projection,” Hunan University; Chansha, China; 2011; web
Appendices:

A. Logo Models

B. More Results
C. Detection of Errors

Low light
The Logo is very hard to detect even with segmenting

Logo out of segmented image range

Multiple regions detected