SENG 435 – DIGITAL IMAGE PROCESSING TERM PROJECT REPORT

ROJDA SÜSLÜ

21050911017

AUTOMATIC LICENSE PLATE RECOGNITION SYSTEM

1.ABSTRACT

Digital image processing is a field which goal is improves image quality, obtain meaningful information and perform automated analysis by applying various operations to images, also recent days it is using in security systems, automatic recognition systems, health industry etc.

Automatic License Plate Recognition (ALPR) systems have become an important application of digital image processing, especially in areas such as parking management, access control, and security systems. In this project I examine the problem of" how can we recognize the plates accurately even in different conditions?" This study is a computer based license plate recognition system is developed using classical image processing techniques and Optical Character Recognition (OCR), without any mechanical or hardware based gate control.

The proposed system processes both static images and live video frames obtained from a computer's webcam. There is a sample house garage, some vehicles which are already

belongs the this house have an acces to the house garage based on a predefined list. The system is accepting only authorized plates otherwise garage door is not opening. Entries and exits are recorded in real time after the recognition completes,so we can always ensure that data is securely updating and protecting. The pipeline includes also some image processing techniques like ,grayscale conversion, noise reduction, contrast enhancement, adaptive histogram equalization .Plate region detection using a Haar Cascade classifier. After extracting the license plate region, OCR is applied to recognize the plate characters. To improve robustness under varying lighting conditions, histogram based analysis and enhancement techniques such as contrast stretching and adaptive histogram equalization are integrated before the OCR stage.

Instead of relying completely on a rule-based decision mechanism, I integrated a lightweight machine learning approach into the system in this study. First, the K-Nearest Neighbor (KNN) classifier was used in the OCR to distinguish an image region is actually a license plate. This allowed for a comparison between traditional rule-based processing and a simple machine learning-based approach. The combined use of classical image processing techniques and a basic machine learning model aims to contribute to a more stable and reliable operation of the system.

2.INTRODUCTION

License Plate Recognition (LPR) is a known problem in the field of digital image processing and computer vision. It detecting a vehicle's license plate from video frame as i used in this project and accurately recognizing the characters on the plate. LPR systems are widely used in applications such as parking automation,garage systems ,traffic monitoring and access control systems.

Today, several models that are widely used and frequently discussed in academic studies, such as the Haar Cascade model, have performed license plate recognition. However, in some cases, these models alone may not be sufficient to accurately recognize license plates. However if some challenges occur like illumination, image contrast, camera angle, motion blur, and environmental noise. Plates captured under poor lighting conditions, such as at night or in shadows such kind of things effects OCR perfomance.

In this project, a computer-based license plate recognition system is designed to give the vehicles permissons as authorized or unauthorized , at the same time I tried some digital image processing techniques to increase the performance of the OCR and recorded them in experiments to see the best way to enhance plate recognition. These techniques includes like histogram analysis, contrast stretching, histogram equalization, adaptive thresholding, and noise removal methods.

The system operates entirely in software, using a computer's webcam for live video input and does not include any physical gate or mechanical control. To introduce an intelligent component into the system, a machine learning-based K-Nearest Neighbors (KNN) classifier is integrated. The classifier is trained using plate and non-plate image samples and utilizes extracted features such as grayscale intensity histograms, mean intensity, contrast like that; mean_intensity = np.mean(gray)

contrast = np.std(gray)

Additionally, edge orientation and local shape information are extracted using Histogram of Oriented Gradients (HOG) features.This classifier is used to filter candidate regions and reduce false detections before OCR is performed, to improve the overall robustness of the system.

3. LITERATURE REVIEW (RELATED WORK)

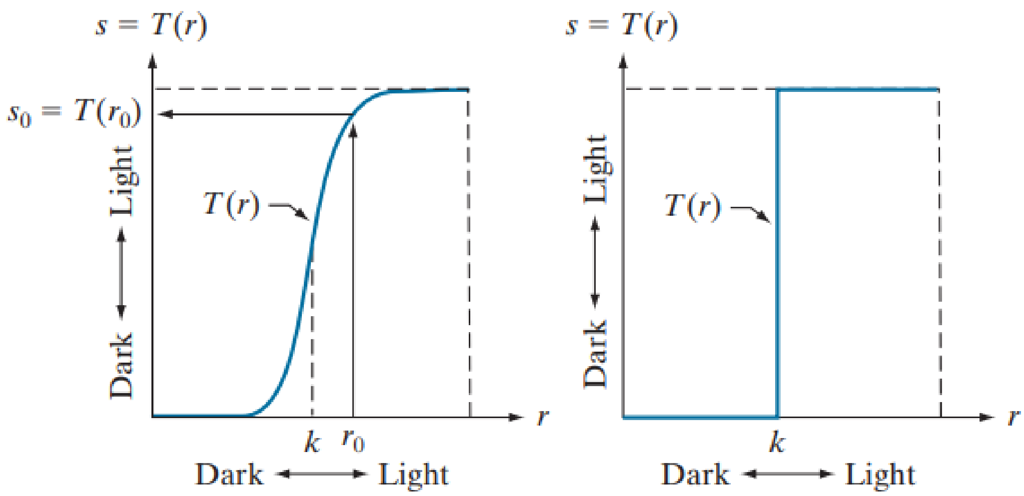
Automatic license plate recognition (ALPR) systems are widely used in fields such as transportation systems, security applications, and access control. ALPR systems generally consist of two main stages: license plate detection and character recognition. Early studies in this field addressed the problem of license plate recognition through still images and video streams.In this project images showed to the webcam to understanding plates.

Anagnostopoulos et al. [1] investigated the license plate recognition process using still images and video sequences and showed that classical image processing based methods yielded successful results under certain conditions, especially in the license plate detection phase. However, this study emphasized that factors such as lighting variations, low contrast, and noise seriously affect the recognition performance as I examine at my study, also this study was helpful for the understand which factors affect how recognition performance and of course fathers of digital image processing.

Fig. 9.
Binarization results from left to right: noisy LP image, global threshold, and adaptive threshold in [59].

In this Anagnostopoulos et al. [1] the results of global thresholding and adaptive thresholding methods on noisy license plate images were visually compared. Inspired by this study, a similar approach was applied in this project as by myself.

In the pioneering work of Rafael C. Gonzalez and Richard E. Woods, leading figures in the field of digital image processing [2], fundamental image enhancement techniques are introduced to improve contrast and image clarity ,it was stated that histogram-based methods are effective in increasing image contrast. such as histogram equalization, contrast stretching, and adaptive histogram equalization are said to make details more visible in images with low light or irregular light conditions. These methods directly affect OCR performance give an clearer separation of characters on the plate.



This figure is from Gonzalez and Woods' Digital Image Processing book. As seen in Figure a, dark pixels are made darker and light pixels are made lighter,The contrast in midtones is increased smoothly and parabolically. In the other graph the thresholding process converts pixel values into black and white form according to a threshold value. Transitions are sharper.Thanks to these processes, the license plate characters are separated more clearly from the background and better results are obtained in the OCR stage [2].

shows an 8-bit image with low contrast. Figure 3.10(c) shows the result of contrast stretching, obtained by setting $(r_1, s_1) = (r_{min}, 0)$ and $(r_2, s_2) = (r_{max}, L-1)$, where r_{min} and r_{max} denote the minimum and maximum intensity levels in the input image,

shows an 8-bit image with low contrast. Figure 3.10(c) shows the result of contrast stretching, obtained by setting $(r_1, s_1) = (r_{min}, 0)$ and $(r_2, s_2) = (r_{max}, L-1)$, where r_{min} and r_{max} denote the minimum and maximum intensity levels in the input image,

Contrast stretching was applied based on the minimum and maximum intensity values of the image, as described by Gonzalez and Woods [2], to expand the intensity range and enhance low-contrast license plate images and this is how i used this formula in my study :

```
min_val = gray.min()

   max_val = gray.max()
```

```
    if max_val - min_val > 0:
        gray = ((gray - min_val) * (255 / (max_val - min_val))).astype("uint8")
```

These studies, in line with the literature, emerged from the improvement of classical license plate detection methods based on image processing. These studies, from the past to the present, inspired me in terms of the approach I should take in my work. In this project, instead of complex models based on deep learning, techniques such as histogram analysis, contrast enhancement, and adaptive histogram equalization, which were covered in the course, were used to target the ALPR system as the central element. These studies also supported the project in terms of ideas and visuals.

4. METHODOLOGY

In this study, classical image processing techniques and optical character recognition (OCR) methods were used together to improve the accuracy and robustness of a real-time license plate recognition system. All methods were designed in theoretical approaches covered in the digital image processing course and implemented on a Python based application.

4.1 System Overview

The system developed within the scope of this study is based on a digital image processing approach that performs real-time license plate recognition and access control. The system generally includes of four main stages: image acquisition, image enhancement, license plate detection and character recognition (OCR) with decision making, and record management. These stages are designed for increase the performance.

4.2 Image Acquisition

Image acquisition constitutes the first and fundamental step of the system. In this study images were obtained from two different sources: static image datasets and video frames captured in real time from computer camera (webcam) with this code block:

```
cap = cv2.VideoCapture(0)
cap.set(3, 640)
cap.set(4, 480)
```

And with this sizes.VideoCapture(1) is for normal cameras but in this project I only used webcam so I adjusted to zero .also static images were used for comparison of image enhancement methods, histogram analysis, and performance evaluations, while live camera images were used to test the real-time operational performance of the system.

4.2.1 Dataset

In this project dataset is mostly includes Turkish vehicle license plates , for training knn classifier images are taken from open access license plate datasets in kaggle , however majority of of data sets available on Kaggle[3] consist only of images taken under daytime conditions and in well environments. This situation may limit the system's resilience to real-world conditions such as low light, shadows, brightness variations, and environmental noise.

In this project dataset includes over the 2500 vehicle plate photos itself to train model more efficiently with its unique features.

So,I manually expanded the dataset , additional license plate images collected from surrounding vehicles under different time periods (daytime, evening, and low-light conditions), viewpoints, brightness levels, and noise factors. This manual images were collected with kaggle dataset and created a new mix dataset for the system works more reliable.

4.2.2 Real-Time Image Acquisition

In live camera mode, the system continuously captures video frames using a computer's built-in webcam which is provided by using **OpenCV** library. The resulting video stream is processed frame by frame, with each frame examine as an independent input image. These frames then go to the preprocessing, license plate detection, image enhancement, and character recognition stages with an order.

The process of capturing images from the camera is performed continuously in the process_camera() function: success, img = cap.read()

4.3 Image Preprocessing

Image preprocessing has critical role in plate recognition systems. The images which are come from the camera sometimes not suitable for processing because of the shadow ,low contrast, lightning conditions Etc. Accordingly, I provided some digital image processing techniques that we see in the course to enhance quality of input image for effective results.

4.3.1 Grayscale Conversion

In the first step, the identified plate region was converted from a color (RGB) image to a grayscale image in here, gray = cv2.cvtColor(img_roi, cv2.COLOR_BGR2GRAY)

It is because of:

- reduction the cost,
- understanding edge and shape information well
- Make system works more effective.
  This conversion allows for a single intensity value to be obtained for each pixel, facilitating the application of histogram based analyses.

4.3.2 Noise Reduction

In license plate images, noise from the camera sensor, motion blur, and environmental effects can negatively impact OCR performance. Therefore, a **bilateral filter** applied to reduce noise with this code block: gray = cv2.bilateralFilter(gray, 11, 17, 17)

Compare than **classical Gaussian approach** ,bilateral filter protecting edges while reducing noise. This preserves the edges of the characters on the plate while reducing background distortions. This feature contributes to obtaining clearer characters, especially on muddy or dirty plate surfaces.

4.3.3 Contrast Stretching

In low-contrast images, the difference between the characters and the background may not be sufficiently clear. What's written on the license plate may not be visible. For example, on a very sunny day, when the license plate surface reflects sunlight intensely:

- The white license plate background becomes excessively bright.
- The difference in intensity between the black characters and the background decreases.

Figure 1: Comparison of image processing techiques on a license plate

4.3.4 Histogram Equalization

Global histogram equalization was initially applied to enhance image contrast; however, in small areas like license plates or in mixed day/night lighting and also if shadow and bright areas are combined, it can sometimes create overbrighten and noise.

Figure 2: contrast stretched image after histooram

Therefore, I used a more advanced histogram equalization method that can adapt easily to environmental conditions.

4.3.4 Adaptive Histogram Equalization (CLAHE)

The classic histogram equalization method, while performing a global contrast enhancement across the entire image, can increase noise, especially in plate images, and cause higher contrast in some areas. This can lead to character lost during the OCR stage.

The **CLAHE** method, on the other hand, divides the image into small regions (tiles) and applies a separate histogram equalization process to each region. This allows for a more balanced enhancement of both the bright and dark areas of the plate. CLAHE method applied the image after the grayscale and bilateral filtering process ,

clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))

gray = clahe.apply(gray) # We limited the contrast to 2.0 to prevent aggressive contrast enhancement and then divided the image into 8×8 tiles.

What is more, noise generation is controlled thanks to the clip limit parameter, which restricts contrast enhancement. This feature ensures more stable results, especially in challenging conditions such as night shots, shadowy environments, day or night lights or dirty plate surfaces.

It has been observed that CLAHE improves the transitions by making them smoother and adapting the photograph to the environment without requiring excessive lighting so in this project, CLAHE was applied after the contrast stretching step for enhance before OCR. This resulted in sharper edges for letters and numbers on the plate, enhanced character separation, and improved OCR accuracy.

4.3.5 Adaptive Thresholding

Before the OCR process, an adaptive thresholding method was applied to allow the characters on the plate to be more clearly separated from the background. In this study, a **Gaussian-based adaptive thresholding** method was used. In this method, instead of a single threshold value, a separate threshold value is calculated for each pixel depending on its neighboring pixels.

This approach shows more successful results, especially in cases where there are shadows, glare, or uneven lighting on the plate. In global thresholding methods, since a single threshold value is used for the entire image, the characters may not be sufficiently clear in scenes with variable lighting conditions.

Adaptive thresholding has resulted in more clearly binarized license plate characters and enhanced character-background separation. This has contributed to more accurate character recognition and increased the OCR process.

4.3.6 Image Resizing for OCR Enhancement

Finally, the resulting binary image was resized to improve OCR performance. Increasing the image scale made character details more distinct and contributed to the Tesseract OCR engine producing more accurate results.

h, w = thresh.shape thresh = cv2.resize(thresh, (w * 2, h * 2), interpolation=cv2.INTER_CUBIC) # The thresholded image resized by a factor of 2 using cubic interpolation to improve character readability.

4.4. OCR Comparison (Before and After Enhancement)

In this section, license plate images with excessive brightness, low contrast, and uneven lighting conditions were given to the OCR engine, then i gave images of the license plate after enhancement, because of measure its performance. The results showed that the OCR engine could not accurately recognize the characters before enhancement.

	Plate 1	Plate 2	Plate3
Before enhancement OCR performance	38A735	860DY0977	F03H408
After enhancement OCR performance	34ADA725	60DT097	03HP408

Figure 4: example plate images

4.5 License Plate Detection (Haar Cascade + KNN Integration)

In this study, two different approaches were used together in the license plate detection phase: Haar Cascade-based object detection and **K-Nearest Neighbors (KNN)** based machine learning classification. This integration allows for the utilization of both traditional image processing methods and a simple machine learning approach ,also applied in code like that:

```
def is_plate_ml(img_roi):
    try:
        features = extract_features(img_roi)
        prediction = knn_model.predict([features])[0]
        return prediction == 1  # 1= plate, 0= non-plate
    except:
        return False
```

With this “is_plate_ml()” function candidate area is verifying with KNN whether it is actually a license plate.

In the first stage, potential license plate regions are detected using the Haar Cascade method. **Haar Cascade** harcascade = "haarcascade_plate_number.xml" which is examined here [4] is a fast and widely used method that detects objects using features such as edges, corners, and rectangles. This method identifies regions in the image that could potentially contain license plates as candidates. However, in some cases, the Haar Cascade method can also identify regions that are not license plates as license plates.

As seen in this example, sometimes the model can also frame areas that are not license plates, such as car headlights, hood, etc.

Therefore, for the model to learn which region is a plate and which is not, and to make the system smarter, a **KNN-based classifier** was used in the second stage. Each candidate region detected by Haar Cascade was given as input to the KNN model. The KNN model classifies this region as either a plate (1) or a non-plate(0), determining whether it is truly a plate.

By tesseract’s wrong windowing I scripted car pictures non plate parts as non_plate dataset and plates to the plate dataset to train model very well , by its own ROI’s.

In this code block we cropped images plate and non-plate dataset:

```
cv2.imshow("Plate Candidate | y=plate  n=non-plate", debug_small)
    key = cv2.waitKey(0)

    if key == ord("y"):
        cv2.imwrite(
            os.path.join(PLATE_DIR, f"plate_{plate_count}.jpg"),
            crop
        )
        plate_count += 1
    else:
        cv2.imwrite(
            os.path.join(NON_PLATE_DIR, f"non_plate_{non_plate_count}.jpg"),
            crop
        )
        non_plate_count += 1
```

A dataset consisting of Turkish vehicle license plates was used to train the KNN model. Images without license plates were also photographed by me to enhance the model's discrimination capabilities. Basic features such as grayscale histogram, mean intensity, and contrast information were extracted from each image. These features were used in the KNN classifier's decision-making process.

Thanks to this integration:

- False license plate detections were reduced.

- Only regions with a high probability of being license plates were sent to the OCR process.
- The overall accuracy and reliability of the system were increased.

In conclusion, while the Haar Cascade method provided a rapid preliminary detection, the KNN classifier acted as an intelligent filter to validate this detection. This approach introduced a machine learning-based intelligent technique to the project and offered a more robust structure compared to classical rule-based systems.

4.6 Character Recognition (OCR) and Decision Making

At this stage, character recognition **(OCR)**[5] was performed on the identified license plate region. The goal of OCR process is to obtain the vehicle license plate number as text from the license plate image that inside of the image processing steps.

After the license plate region was determined, the image was converted to grayscale, and preprocessing steps such as noise reduction, contrast enhancement, adaptive histogram equalization (CLAHE), and adaptive thresholding were applied. After these steps, OCR recognizes the characters in the image and converts them into text.

The **Tesseract OCR library "**import pytesseract"was used for character recognition. Tesseract's settings were limited to recognizing only uppercase letters and numbers. This reduced the possibility of incorrect character identification. The text obtained as a result of OCR was cleaned using regular expressions for suitable withTurkish vehicle license plate format.

This garage system belongs to a house, and the house's license plates are pre-defined. If the arriving vehicle is on the authorized list and has not entered before, access is granted and the system records this as "Authorized Access". For unauthorized license plates, an "Unauthorized Access" warning is generated, and the entry process is blocked.

52DRO17	2025-12-28 02:07:27	Unauthorized Access
35ADA725	2025-12-28 02:08:18	Authorized Access
16ED4168	2025-12-28 02:10:50	Unauthorized Access

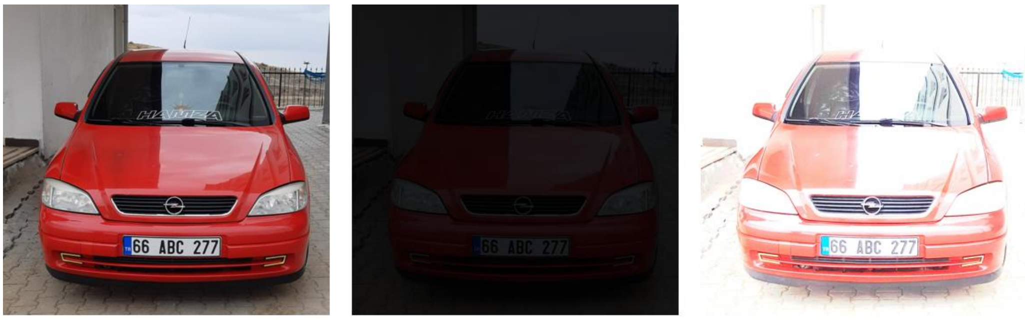
System does not allow the open garage door for unauthorized acces.

5.EXPERIMENTAL RESULTS

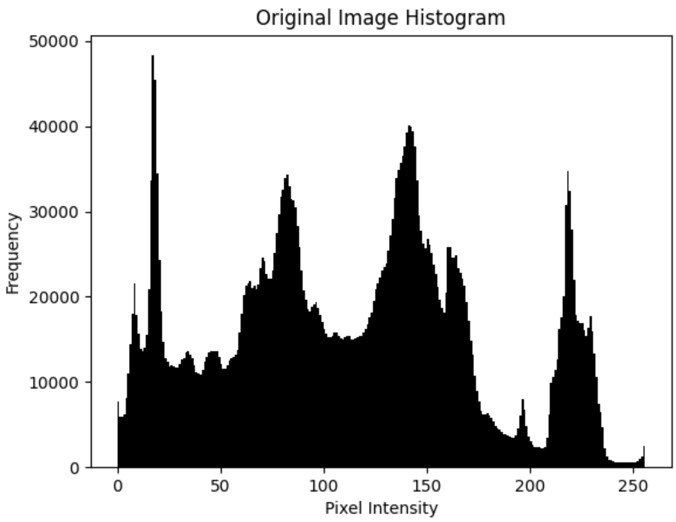
In this section, the performance of the developed license plate recognition system is evaluated through experimental results. Experiments were conducted using both static and live camera images. The aim is to examine the effect of image enhancement steps and the added machine learning-based approach on system performance.

5.1 Histogram Analysis Results

In experimental studies, dark, bright, and normally lit images of the same plate were used. When the histograms of these images were examined, it was observed that pixel densities were concentrated on the **left side** of the histogram in low-lit images and on the **right side** in bright images. This results in a loss of contrast and a decrease in OCR performance.

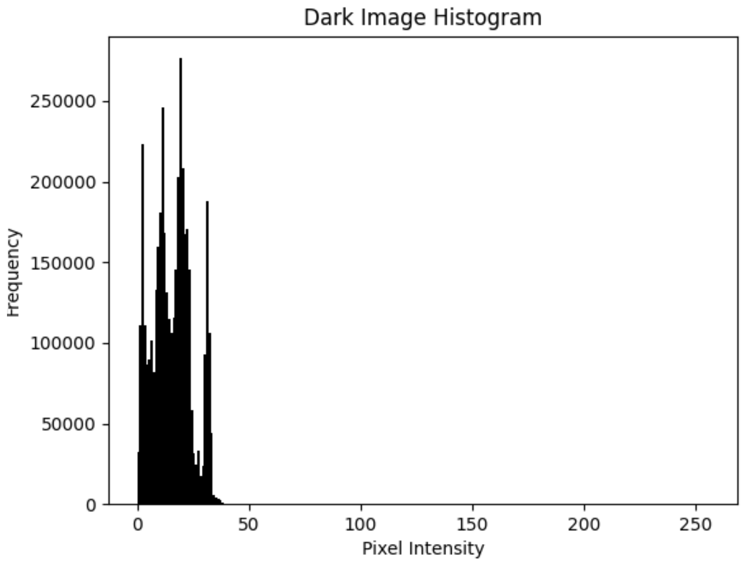


Original Plate | Dark Plate | Bright Plate

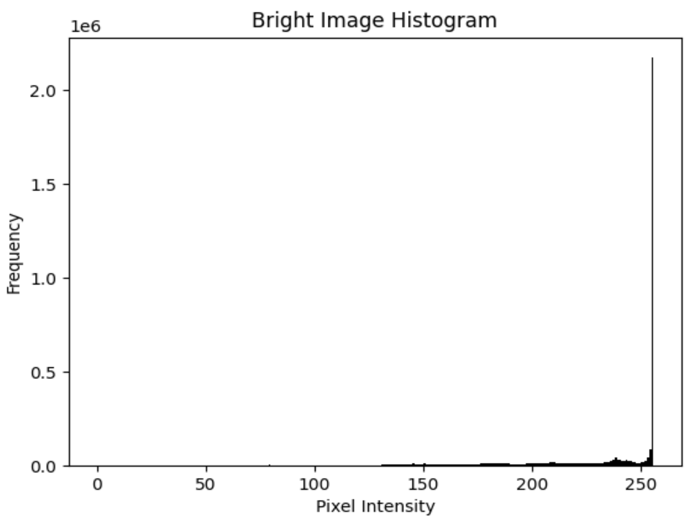



```
def show_histogram(img, title):
plt.figure()
plt.hist(img.ravel(), bins=256,
range=[0,256],color="black")
plt.title(title)
plt.xlabel("Pixel Intensity")
plt.ylabel("Frequency")
plt.show()
```

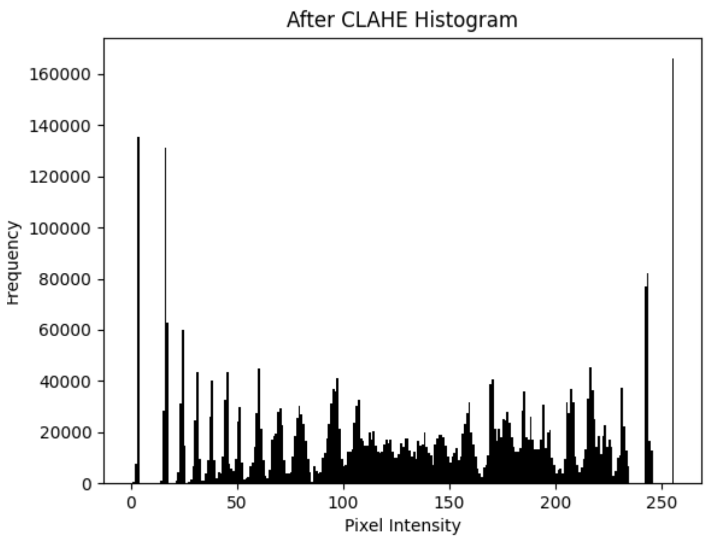
it was observed that pixel densities were concentrated on the left side of the histogram in low-lit images



Overbright plate observed that pixel densities were concentrated now on the right side of the histogram in low-lit images



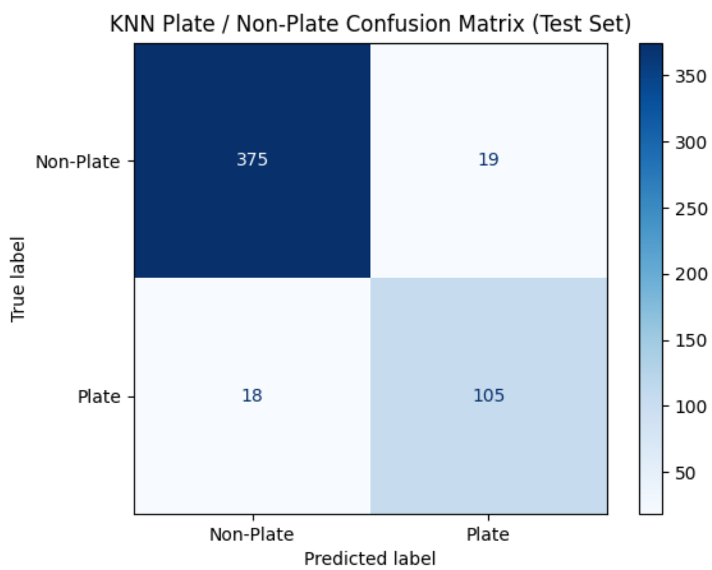
After applying histogram equalization, and specifically Adaptive Histogram Equalization (CLAHE), a more even distribution of pixel densities was observed. This resulted in more distinct character edges and improved readability.



This histogram shows dark light images after enhancement techniques used like CLAHE then its last histogram graph. Pixel intensities are distribute more balance.

5.2 KNN-Based License Plate Classification Results

KNN model integrate to the system for understanding which place is plate and which is not.Thanks to the KNN some regions detected by Haar Cascade but not plate-related were eliminated before the OCR stage. This both reduced **false positives** and increased the overall accuracy of the system. Let's see confusion matrix and train results



Since the dataset is imbalanced, accuracy alone is not sufficient; therefore precision, recall, and the confusion matrix are used for evaluation. The dataset was split into 80% for training and 20% for testing using stratified sampling to preserve class distribution.

test_size=0.2,   # %20 test %80 train

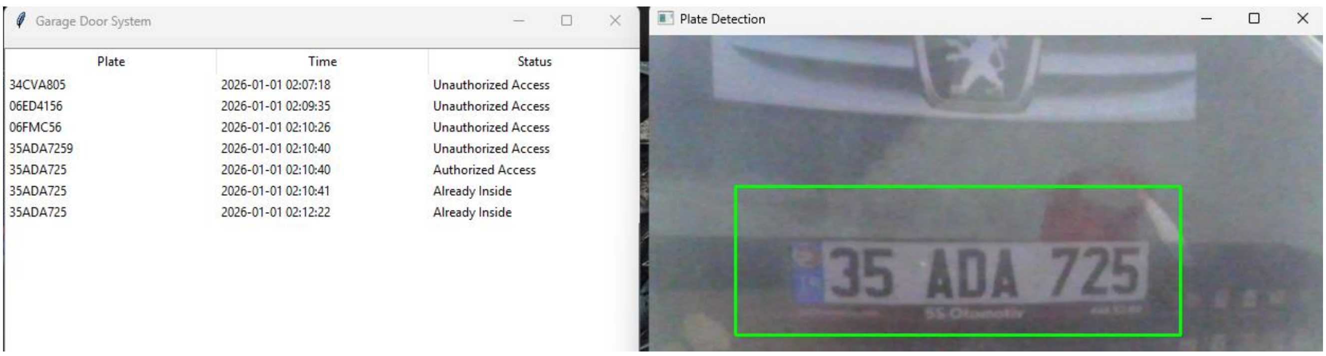
Method	Accuracy	Precision	Recall
Baseline	82%	63%	63%
KNN+HOG	91%	78%	90%
HOG + Tuned k	93%	85%	85%

Histogram based intensity features were insufficient to capture the structural characteristics of license plates. Therefore, **Histogram of Oriented Gradients (HOG)** was employed to encode edge and shape information,hog_features = hog.compute(gray).flatten()

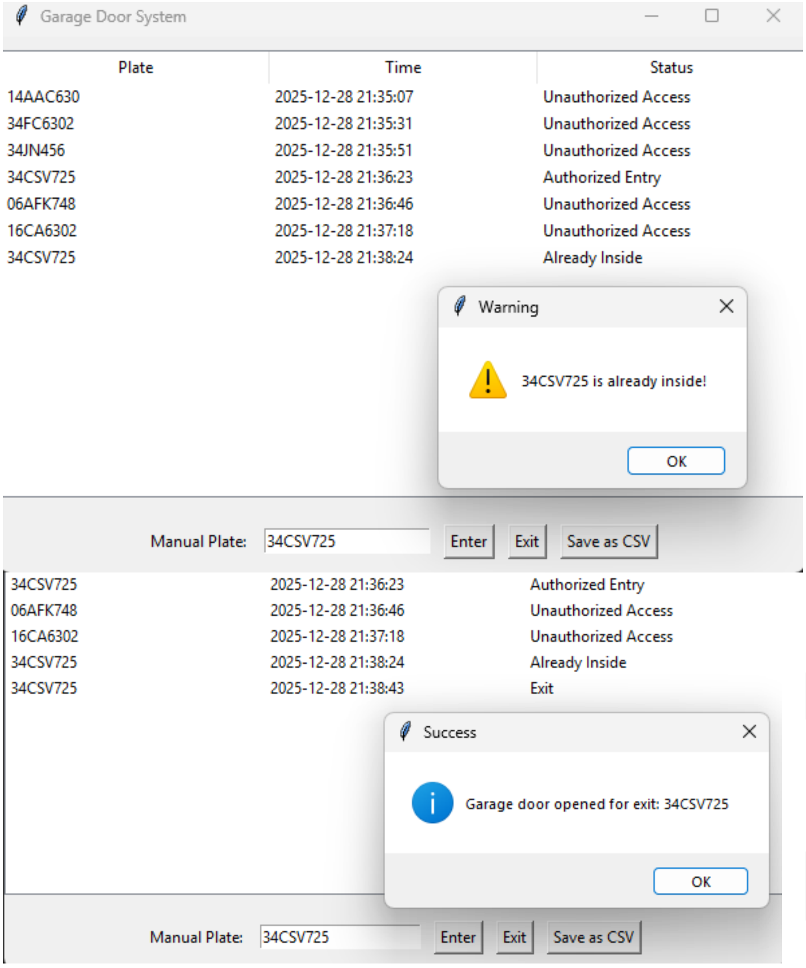
leading to a significant improvement in classification performance. It increased the performance . For increase the precision . To analyze the model's behavior, the number of neighbors (k value) was tested for different values.In this study, by choosing k = 2, for the model make more selective decisions. Lower k values caused the model to make decisions only based on stronger similarities, which contributed to a decrease in false positive predictions the precision value increased.However, the model becoming more selective led to the failure to detect some true positive examples, and a decrease in the recall value was observed with the increase in the number of false negatives. This precision-recall balance was clearly analyzed through the confusion matrix.

5.3 Real-Time System Performance

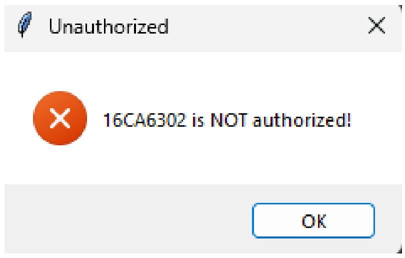
In live camera tests, the system operated in real-time and successfully detected and recognized license plates. "Authorized Access" warnings were generated for authorized plates, and "Unauthorized Access" warnings for unauthorized plates.



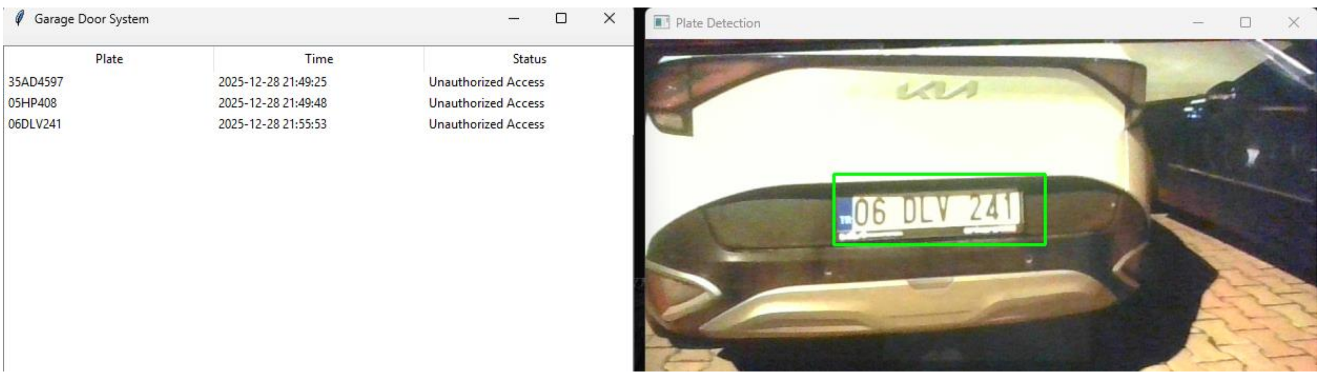
Entries can possible with manually handwriting or with webcam video frame. When a car is inside, then system gives a warning.



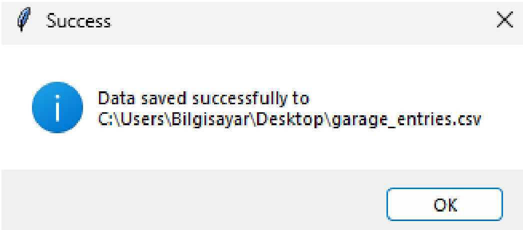
Also for exit, garage door is opening for only authorized acces.otherwise it gives a warning.



Despite the improvements and enhancements made, many experiments have shown that during live camera tests, the system sometimes makes errors in character recognition or fails to read text on the license plate area, even though it correctly detects the license plate region. The main reasons for this include insufficient or excessive ambient light, reflections and glare effects on images displayed on the phone screen, camera angle, and reduced image clarity.



Lastly, all recognized license plates along with their access status (authorized, unauthorized, already inside) are stored in a CSV file in real time, enabling systematic logging and future evaluation of the system performance.



6. CONCLUSION

In this study, a real time Automatic License Plate Recognition (ALPR) system was designed and implemented by combining classical digital image processing techniques optical character recognition (OCR) and a machine learning approach. The main objective was to improve license plate recognition performance under challenging conditions such as low contrast, dirty plates, reflections, and noise, which are commonly captured from real world environments.

Experimental results showed that image enhancement techniques are milestones in improving OCR accuracy. Methods such as contrast stretching, histogram analysis, adaptive histogram equalization (CLAHE) significantly enhanced the visibility of license plate characters, especially in poorly illuminated or overly bright images. Histogram based analysis clearly demonstrated how brightness and contrast variations affect pixel intensity distributions and how these variations can be balanced using enhancement techniques.

In addition to classical image processing, a K-Nearest Neighbors classifier was integrated into the system to seperate license plate regions from non-plate regions. This machine learning-based component added an intelligent decision layer to the system and reduced false detections before OCR processing. .

The final system successfully performed live license plate recognition using a webcam and classified detected plates as authorized or unauthorized based on predefined access rules. All detected entries were recorded in real time dynamically, making the system suitable for access control scenarios such as garage management.

One problem I encounter when capturing live camera images is the effects of reflection, glare, and shadows caused by ambient light. Especially when displaying license plates on a mobile phone screen, differences in screen brightness and angle can make it difficult to read the characters.to mitigate these issues, the following improvements could be made in the future:

- Exposure adjustment could be applied at the camera input.
- Motion blur reduction techniques could be used.
- A larger dataset could be created using license plate images taken from different angles to improve the machine learning model.

With these improvements, the system could produce more stable and reliable results in real-world conditions.

7.REFERENCES

1. [1] Anagnostopoulos, C. N. E., et al., "License Plate Recognition from Still Images and Video Sequences," IEEE Transactions on Intelligent Transportation Systems. https://doi.org/10.1109/TITS.2008.922938
2. [2] Gonzalez, R. C., Woods, R. E., Digital Image Processing, Pearson, 2018.
3. S. Durcan, "Turkish License Plate Dataset," Kaggle, 2023. https://www.kaggle.com/datasets/smaildurcan/turkish-license-plate-dataset
4. [4] OpenCV Documentation,"Haar Cascade Classifier," https://docs.opencv.org/
5. [5] Smith, R.,"An Overview of the Tesseract OCR Engine," Proceedings of the International Conference on Document Analysis and Recognition (ICDAR), 2007.