# VEHICLE NUMBER PLATE RECOGNITION SYSTEM

**Arjit Avadhanam**
*S20210020257*
*Electronics and Communication Engineering*
*Indian Institute of Information Technology, Sricity*
arjit.a21@iiits.in

**Mahan Bheemireddy**
*S20210020260*
*Electronics and Communication Engineering*
*Indian Institute of Information Technology, Sricity*
maneendra.b21@iiits.in

**Gowtham N**
*S20210020277*
*Electronics and Communication Engineering*
*Indian Institute of Information Technology, Sricity*
gowtham.n21@iiits.in

***Abstract*—Vehicle Number Plate Recognition is a critical component in intelligent transportation systems, leveraging pattern recognition techniques to automatically extract and interpret alphanumeric characters from vehicle license plates. This technology plays a pivotal role in enhancing traffic management, security, and law enforcement. The system involves image acquisition, preprocessing, feature extraction, and pattern recognition. Various algorithms such as support vector machines, and deep learning models have been employed to achieve accurate and efficient license plate recognition. The utilization of pattern recognition terms ensures robust identification of license plate patterns, enabling the system to adapt to diverse environmental conditions**

***Keywords—Image Acquisition, Preprocessing, Feature Extraction, Support Vector Machines (SVM), Sequential Feature Selection.***

## I. INTRODUCTION

The Vehicle Number Plate Recognition (VNPR) System is an advanced technology designed for automatic identification and interpretation of alphanumeric characters on vehicle license plates. Employing sophisticated image processing and pattern recognition techniques, this system plays a pivotal role in various applications, including traffic management, law enforcement, and security. By utilizing cameras or sensors to capture images of license plates, the VNPR system undergoes a multi-stage process involving image preprocessing, feature extraction, and pattern recognition algorithms. The system's capability to adapt to diverse environmental conditions ensures robust performance, making it a key component in the development of smart cities, optimizing traffic flow, enhancing security measures, and facilitating efficient law enforcement.

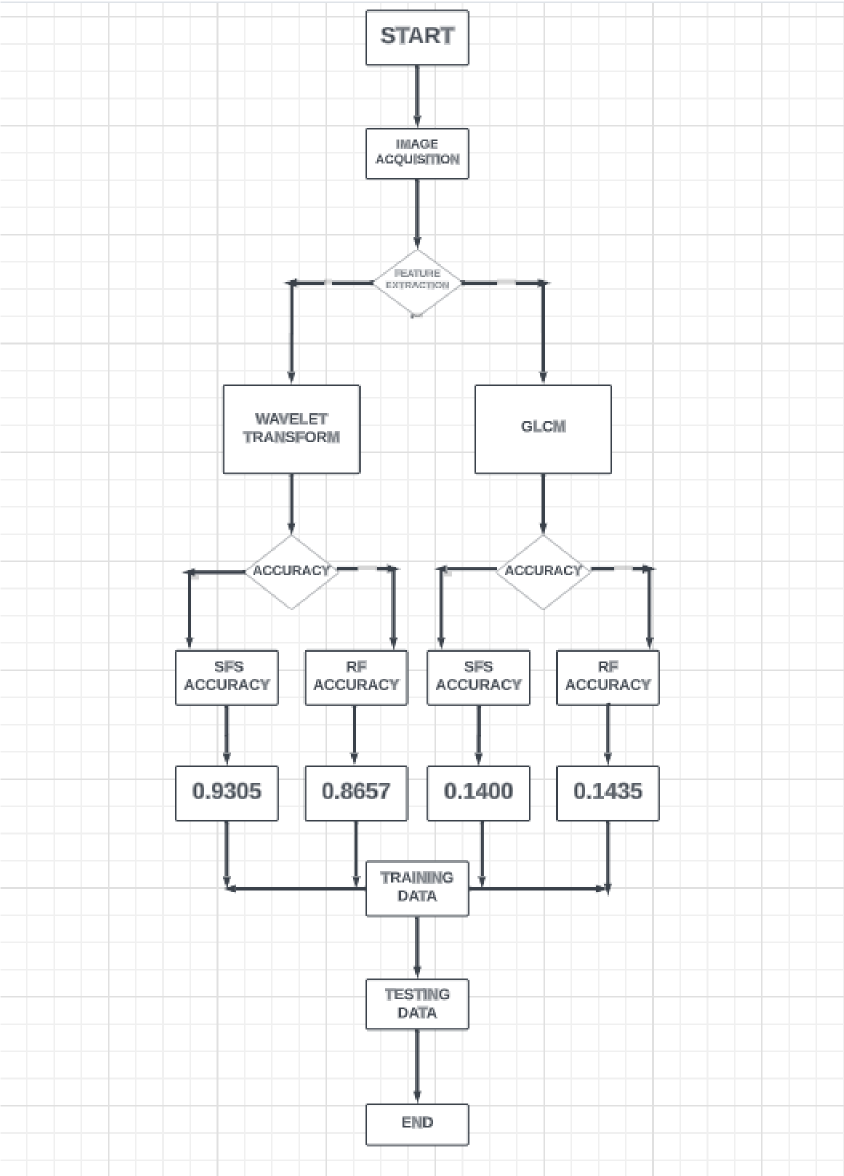
## II. EASE OF USE

The design and implementation of a robust Vehicle Number Plate Recognition (VNPR) system must prioritize *ease of use*. The *User Interface (UI)*, through which users interact with the VNPR system, whether it's a control panel or software application, should be designed to be intuitive and user-friendly. *System Integration* is crucial, with the integration process being seamless and straightforward, allowing users to easily incorporate the VNPR system into their existing systems if it is part of a larger infrastructure. *Maintenance and Updates* are key considerations; the VNPR system should be designed to facilitate easy maintenance and updates, minimizing downtime and technical challenges for administrators or operators. *Adaptability* is also a priority, with the system being capable of adjusting to different environments and conditions with minimal user intervention, demonstrating ease of use across various scenarios. Furthermore, considerations for *Training Requirements* are essential, ensuring that users can operate the system effectively with minimal training.

## III. COMPONENTS REQUIRED

- Kaggle – For Dataset
- Google Colab Notebook
- Files from google drive
- Libraries Utilized
    1. Numpy – For performing numerical operations or manipulations on arrays.
    2. Scikit-learn – For utilizing machine learning algorithms and tools from the Scikit-learn library.
    3. TensorFlow **–** For deep learning models are part of your methodology.
    4. Matplotlib or Seaborn **-** For creating visualizations or plots in your analysis.
    5. OpenCV **-** For using OpenCV for image processing tasks.
    6. Ski-image – For leveraging scikit-image for additional image processing or feature extraction tasks.

## IV. BLOCK DIAGRAM




## V. METHODOLOGY

### A. *Image Acquisition*

Image acquisition is the process of capturing visual data from the real-world using devices like cameras. It involves converting optical information into digital formats for subsequent computer-based processing and analysis. It captures images of vehicle license plates using cameras or sensors.

### B. *Image Preprocessing*

Image preprocessing involves enhancing and optimizing raw images before analysis. Common techniques include normalization, resizing, and noise reduction. This step ensures improved data quality and facilitates effective feature extraction in various computer vision applications.

### C. *Feature Extraction*

1. *Wavelet Transform*- The *apply_wavelet_transform* function, at its core, subsequently performs a 2-level wavelet decomposition using the *'haar'* wavelet, flattening the resulting coefficients. The *ImageDataGenerator* for training is configured to include pixel rescaling, width and height shifts, and the application of the wavelet transform. This generator is then used to flow images from the 'train' directory with a target size of (28, 28) and a batch size of 1, adopting a sparse class mode for integer labels. Similarly, a validation generator is set up for images from the *'val'* directory.

2. *Gray Level Co-occurrence Matrix (GLCM)*- The integration of Grey Level Co-occurrence Matrix (GLCM) feature extraction as a preprocessing step converts the given image into grey scale. The *apply_glcm_feature_extraction* function encapsulates the process of converting input images to grayscale if necessary, adjusting data type for compatibility, and computing the GLCM using specified distances and angles. The ImageDataGenerator ***(train_datagen_glcm)*** is configured to rescale pixel values, apply width and height shifts, and employ the GLCM feature extraction during preprocessing.

### *D. Feature Ranking*

Feature ranking using a Random Forest classifier for a given set of training data (*x_train* and *y_train*). The features, obtained from a wavelet transform, are converted to NumPy arrays, and the classifier is trained to evaluate feature importances.

### *E. Sequential Feature Selection (SFS)*

Sequential Feature Selector is employed for feature selection, working in tandem with Principal Component Analysis (PCA) and Support Vector Machine (SVM) classification. The script dynamically adjusts the number of selected features (**i**), utilizes PCA to reduce dimensionality, and evaluates the SVM classifier on both the validation set and the test set using the same selected features. This comprehensive approach aims in balancing accuracy and dimensionality reduction.

### *F. Support Vector Machines (SVM)*

The SVM classifier from scikit-learn to fit a model to training data (*x_train* and *y_train*). The score method is then utilized to evaluate the model's performance on a validation set (*x_val* and *y_val*). The script provides a concise approach to training an SVM classifier and assessing its accuracy on a separate validation dataset.

### *G. Training Data*

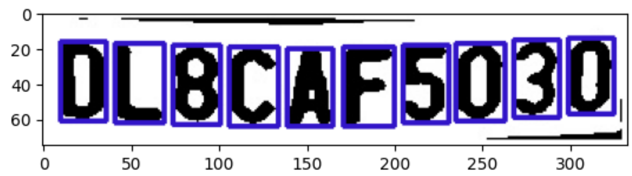
Training data is a labeled set of diverse images of license plates used to train the system. It enables the system to learn and adjust its parameters, ensuring accurate recognition of varying fonts, styles, and environmental conditions. The quality and diversity of this dataset directly impact the system's ability to generalize and improve recognition accuracy.

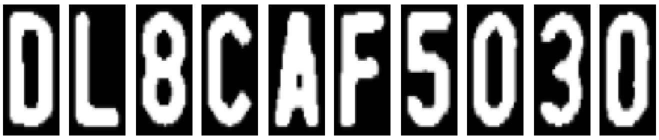
### *H. Testing and Validating Data*

Testing and validation data in Vehicle License Plate Recognition involve evaluating the system's performance using independent datasets. This process ensures accuracy, reliability, and the system's ability to generalize to diverse scenarios. The results obtained from these assessments help validate the effectiveness of the VNPR system.

## VI. RESULTS





## *1. By using Wavelet Transform:*

**Data Summary for Image Classification:**

- Training Dataset: Found 864 images belonging to 36 classes.
- Validation Dataset: Found 216 images belonging to 36 classes.

**Feature Ranking:**

Feature importances [0.00643473 0.00331517 0.00187331 0.00238787 0.00359379...] are ranked accordingly.

**Sequential Forward Selection (SFS) Performance Metrics:**

Number of Features: 90

Shape of x_train: (864, 784)

SFS Accuracy on Validation Set: 0.9351851851851852

SFS Accuracy on Test Set with Same Features: 0.9351851851851852

SFS Accuracy: 0.9351851851851852

**Random Forest Classifier Performance Metrics:**

Number of Features: 90

Random Forest Accuracy on Validation Set: 0.7824074074074074

Random Forest Accuracy on Test Set with Same Features: 0.7824074074074074

Random Forest Accuracy: 0.7824074074074074

**SVM Classifier Performance Metrics:**

SVM Accuracy: 0.8842592592592593

**Final Output:**

BL8CAF5030

## *2.By using Gray Level Co-occurrence Matrix:*

**Data Summary for Image Classification:**

- Training Dataset: Found 864 images belonging to 36 classes.
- Validation Dataset: Found 216 images belonging to 36 classes.

**Feature Ranking:**

Feature importances [0.09091893192424406, 0.028045680367904054, 0.021037982049240206...] are ranked accordingly.

**Sequential Forward Selection (SFS) and Random Forest Performance Metrics:**

Random Forest Accuracy with GLCM Features: 0.16203703703703703

SFS Accuracy with GLCM Features: 0.14004629629629628

**SVM Classifier Performance Metrics:**

SVM Accuracy: 0.14351851851851852

**Final Output:**

DL8CAF5030 - Only displayed in gray scale – due to low accuracy.

## VII. Analysis

- **Wavelet Transform:**
  - Achieved high accuracies in both SFS and Random Forest classifications.
  - SVM accuracy is notably high at 0.8843.
- **GLCM:**
  - Demonstrated significantly lower accuracy in both Random Forest and SFS classifications.
  - SVM accuracy is considerably low at 0.1435.

Thus, Wavelet Transform is a more effective feature extraction technique compared to GLCM, since it has yielded higher accuracies in both SFS and Random Forest classifications and achieved a notably higher overall accuracy compared to GLCM.

## VIII. Conclusion

Hence, we can conclude that the Wavelet Transform captures more discriminative features relevant to the classification task, making it a better choice for this specific image classification scenario. The consistently high performance of Wavelet Transform across Sequential Forward Selection (SFS) and Support Vector Machine (SVM) accuracy, underscores its robustness in extracting meaningful information from the image data. Thus, Wavelet Transform contributes to a more effective representation of image features in comparison to Gray Level Co-occurrence Matrix. Therefore, GLCM is not a suitable method for vehicle number plate recognition.

## IX. Contributions

ARJIT AVADHANAM – S20210020257
Feature Extraction – Using GLCM and Wavelet Transform

MAHAN BHEEMIREDDY– S20210020260
Training and testing of dataset and fine-tuning the model to improve performance on specific challenges in the dataset.

GOWTHAM N – S20210020277

Selecting of appropriate dataset, collecting it, and obtaining its analysis and image pre-processing.

## X. References


[1] Anish Chand Sir - for guiding us throught the semester for our pattern recogniton project.

[2] Kaggle – for datasets – downloaded the dataset named indian license plate.xml.

[3] Gonzalez, R. C., Woods, R. E., & Eddins, S. L. (2009). Pattern Recogniton Using Google Colab.

[4] Google Chrome – downloaded images from internet for various input vehicle number plate images

[5] Bradski, G., & Kaehler, A. (2008). Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media.

[6] Lohri, A., & Quack, T. (2009). License Plate Recognition using Google Colab and IEEE.