5.1 Introduction

License Plate Recognition (LPR) is an image-processing technology [199] that was used to identify vehicles by their license plates. It is used in various security and traffic applications, such as the access-control system, as featured in the following figure 5 (a).

![Access control system](image)

**Figure 5 (a): Access control system**

In this example, as the vehicle approaches the gate, the LPR unit automatically "reads" the license plate registration number, compares it to a predefined list and opens the gate if there is a match.

The technology is gaining popularity in security and traffic installations. The access control system uses illumination and a camera to capture the image of the front or rear of the vehicle. Image-processing software is then used which analyzes the images and extracts the plate information. This data is utilized for enforcement, data collection, and can be used to open a gate if the car is authorized. A time record is also maintained to notify the entry or exit of the vehicle for automatic payment calculations.

The License Plate Recognition system has a significant advantage that the system can keep an image record of the vehicle which is useful to fight/crime and fraud. An additional camera is also used which focus on the driver face and save the image for security reasons.
LPR technology has a wide range of applications, which use the extracted plate number and optional images to create automated solutions for various problems. These include the following applications:

**Parking** - the plate number is used to automatically enter pre-paid members and calculate parking fee for non-members (by comparing the exit and entry times). The optional driver face image can be used to prevent car hijacking.

In this example, a car is entering a car park in a busy shopping center. The car plate is recognized and stored. The car plate is read again, when the car exits (through the gate on the right side). The driver will be charged for the duration of the parking. The gate will automatically open after payment - or if the vehicle has a monthly permit.

Figure 5 (b): Car Parking
**Access Control** - a gate automatically opens for authorized members in a secured area, thus replacing or assisting the security guard. The events are logged on a database and could be used to search the history of events. In this example, the gate has just been automatically raised for the authorized vehicle, after being recognized by the system. A large outdoor display greets the driver. The event (result, time and image) is logged in the database.

**Tolling** - the car number is used to calculate the travel fee in a toll-road, or used to double-check the ticket. In this installation, the plate is read when the vehicle enters the toll lane and presents a pass card. The information of the vehicle is retrieved from the database and compared against the pass information. The operator is notified, in case of fraud.
**Border Control** - the car number is registered in the entry or exits to the Country, and used to monitor the border crossings. It can short the border crossing turnaround time and cut short the typical long lines.

This installation covers the borders of the entire Country. Each vehicle is registered into a central database and linked to additional information such as the passport data. This is used to track all border crossings.

**Stolen cars** - a list of stolen cars or unpaid fines is used to alert on a passing 'hot' cars. The 'black list' can be updated in real time and provide immediate alarm to the police force. The LPR system is deployed on the roadside, and performs a real-time match between the passing cars and the list. When a match is found a siren or display is activated and the police officer is notified with the detected car and the reasons for stopping the car.
Enforcement - the plate number is used to produce a violation fine on speed or red-light systems. The manual process of preparing a violation fine is replaced by an automated process which reduces the overhead and turnaround time. The fines can be viewed and paid on-line. The photo is an example of a speeding car caught by the traffic camera. The rear vehicle plate is automatically extracted off the scanned film image, replacing a tedious manual operation and the need to develop and print the violation. The data block on the top-right side is additional speeding information that is automatically extracted from the developed film and used to complete the fine notice and inserted to a database. The violators can pay the fine on-line and are presented with this photo as a proof with the speeding information.

Figure 5 (f): Enforcement
**Traffic control** - the vehicles can be directed to different lanes according to their entry permits (such as in University complex projects). The system effectively reduces traffic congestions and the number of attendants.

In this installation the LPR based system classifies the cars on a congested entrance to 3 types (authorized, known visitors, and unknown cars for inquiry) and guides them to the appropriate lane. This system reduced the long waiting lines and simplified the security officers work load.

**Marketing Tool** - the car plates may be used to compile a list of frequent visitors for marketing purposes, or to build traffic profile (such as the frequency of entry verses the hour or day).
Travel - A number of LPR units are installed in different locations in city routes and the passing vehicle plate numbers are matched between the points. The average speed and travel time between these points can be calculated and presented in order to monitor municipal traffic loads. Additionally, the average speed may be used to issue a speeding ticket. In this example the car is recognized at two points, and the violation shows the photos of both locations which were taken on bridges on top of the highway. The average speed of the car is calculated from both points, and displayed if the speed passed a violation threshold, and optionally printed.

Figure 5 (h): Travel
Airport Parking - In order to reduce ticket fraud or mistakes, the LPR unit is used to capture the plate number and image of the cars. The information may be used to calculate the parking time or provide a proof of parking in case of a lost ticket - a typical problem in airport parking which have relatively long (and expensive) parking durations.

This photo shows the gate of a long term airport parking. The car is recognized on entry and the data is later used to track the real entry time in case of a lost ticket. The case of recognition of characters from vehicle number plate is a developing broad field of research. It can be well seen from the list of applications. This broad area of research was chosen, based on the importance of identifying a vehicle for various security purposes.

5.2 Vehicle Number Plate Recognition using Neuro Fuzzy Classifier.

Vehicular traffic in transportation is increasing every year due to growing population. Vehicles are playing a very important part in transportation. Traffic management and vehicle identification is becoming a complex problem to resolve. Manual identification is very cumbersome and time consuming. To resolve this problem an image processing technique called License Plate Recognition (LPR) [157] is proposed. LPR technique is used for automatic vehicle identification. Number of areas like traffic level controlling, monitoring of invalid parking, enforcing the traffic laws and auto charge collections on highways are some of the applications of license plate recognition [76, 33].

A simulated system model called Neural Network (NN) is an emulation of the cerebral system in the Human Brain. Feed forward neural network is a conventional NN and it is trained using the extensive back-propagation algorithm [135, 175]. Plate recognition and identification usually employ neural networks. Pulse Coupled Neural Network (PCNN), the Time Delay Neural Network (TDNN) and the Discrete Time Cellular Neural Network. (DTCNN) are some of the neural networks architectures
projected and realized for plate identification [61]. The provocation to use artificial neural network is because of its ability to read general character styles independent of its size and location.

Fuzzy logic is used in tackling the problem of locating license plates. Few automatic criterions were utilized, to retrieve the horizontal and vertical plate positions. A major step into LPR system is the color recognition of license plates [43]. Two license plates were different even if the characters are similar, because of their color. Not only color, but different formats are also used in license plate. As a result color recognition has a huge effect on succeeding steps, for example character separation [12,8]. The final recognition accuracy of license plates can be enhanced by an improvement in the accuracy of color recognition. The category of recognition accuracy relies not only on color, but also on various others factors [30]. Hence considering all these factors a neuro-fuzzy classifier is proposed for the recognition of characters from vehicle number plate images.

5.3 Proposed Method 3 – Pattern Recognition in Digital Images using Neuro Fuzzy Classifier

Among various application areas of pattern recognition, recognizing characters from vehicle number plate plays a very vital role in the modern era. Due to the increasing number of vehicles, the technique is absolutely necessary to identify a particular vehicle. A neuro fuzzy classifier is proposed to recognize characters from vehicle number plate images.

5.3.1 Steps in proposed method 3

The steps in the proposed method are outlined below:

i) Reading the RGB image.

ii) Grayscale conversion.

iii) Extraction of characters with bi partitioning algorithm.

iv) Clustering operation is carried out with Modified Fuzzy C Means clustering algorithm.

v) Recognition of extracted characters with neuro fuzzy classifier.

5.3.1.1 Computational procedure for the proposed method 3

Step 1: The given input image first undergoes the preprocessing stage. The input image was resized in the range [350, 400], and is then converted into a grayscale image.
Step 2: Feature Extraction - Bi partitioning Algorithm

Feature extraction process is the major step in any image recognition process. The feature from the input images are extracted and based on these feature values the classification is performed. Bi partitioning is an iterative improvement algorithm which is widely used in feature extraction process. Bi partitioning algorithm usually divides the characters as LHS and RHS.

Input: $G = (V, E) |V| = 2n$

Output: Balanced bi-partition $A$ and $B$ with “small” cut cost.

begin
Bipartition $i$ into $A$ and $B$ such that $|V_A| = |V_B|$, $V_A \cap V_B = \emptyset$, and $V_A \cup V_B = V$;
repeat
Compute $D_v$, $\emptyset v \in V$;
for $i = 1$ to $n$ do

Find a pair of unlocked vertices $v_a \in V_A$ and $v_b \in V_B$ whose exchange makes the largest decrease or smallest increase in cut cost;

Mark $v_a$ and $v_b$ as locked, store the gain, and compute the new $D_v$, for all unlocked $v \in V$;

Find $k$, such that $G_k = \sum_{i=1}^{k} \hat{g}_i$ is maximized;

if $G_k > 0$ then
Move $v_{a1} \ldots v_{ak}$ from $V_A$ to $V_B$ and $v_{b1} \ldots v_{bk}$ from $V_B$ to $V_A$;
Unlock $v$, $\emptyset v \in V$;

until $Gk \leq 0$;
end
The steps in the bipartitioning algorithm are described below:

(a) The algorithm begins with an initial partition and iteratively transforms it to get better cutsize.
(b) The number of nets connected to nodes in both partitions is called the cutsize, and it’s the value to be minimized.
(c) The algorithm moves a node in the cutsize.
(d) If the algorithm is allowed to move any arbitrary node, it could decide to move the node just moved in the previous iteration, returning to the previous state.
(e) The algorithm would then be caught in an infinite loop, making no progress.
(f) To deal with this, the node is locked after it is moved, and never moves a locked node.
(g) The algorithm continues moving the nodes until no unlocked node can be moved without violating the size constraints.
(h) It then checks whether the cutest is improved or not.
(i) It back track across all the intermediate states since the last check, finding the minimum cutsize. This allows it to climb out of local minima, finding a better later state.
(j) In the best intermediate state, all the nodes are unlocked and continue.
(k) The above steps are repeated until no more better intermediate state can be found.

Thus the feature values are extracted and the partitioned image undergoes clustering process.

Step 3: Modified Fuzzy C Means Clustering.

(a)  \[ O = \sum_{i=1}^{N} \sum_{j=1}^{C} (1-\alpha)u_{ij}^{-m} \|x_i - C_j\|^2 \]

where \( m \) is any real number greater than 1, \( u_{ij} \) is the degree of membership of \( x_i \), \( C_j \) represents the center value of the cluster, \( x_i \) is the measured data, \( \alpha \) specifies a constant that is used to minimize the objective function.

(b)  \[ f(x;a,b,c) = \max \left( \min \left( \frac{x-a}{b-a} , \frac{c-x}{c-b} \right) , 0 \right) \]
where $x$ represents the data points and $a, b, c$ are the three scalar parameters.

(c) $u_{ij} = \frac{1}{\sum_{k=1}^{c} \left( \frac{\| x_i - c_j \|^{2m-1}}{\| x_i - c_k \|^{2m-1}} \right)}$

Where, $x_i$ is the input data, $c$ is the centroid and $m$ is a positive constant.

(d) $c_j = \frac{\sum_{i=1}^{N} u_{ij}^m x_i}{\sum_{i=1}^{N} u_{ij}^m}$

(e) $\max_{ij} \left\{ |\mu_{ij}^{(k)} - \mu_{ij}^{(k+1)}| \right\} < \omega$

Where, $\omega$ is the termination condition whose value is between 0 and 1.

(f) if $(A_i < T_a)$, then discard $O_i$

(g) $c_x = \frac{p_x + q_x + r_x + s_x}{4}$; $c_y = \frac{p_y + q_y + r_y + s_y}{4}$

(h) $W = |A_R - B_L|; H = |A_R - B_B|$

FCM algorithm is used for the inclusion of fuzzy concept to have more accurate solution. The main motive here is to reduce the objective function with minimum number of iterations. It is well achieved from the equation Step 3(a). It is mainly used to minimize the objective function of FCM. The value of $\alpha$, helps to minimize the objective function.

The membership values are updated as given in equation Step 3 (c). The centroid values are updated as given in equation step 3(d). The iterative optimization procedure is continued till it satisfies the equation in step 3 (e).

A cluster may not represent an object in reality. Area constraint is then introduced to cross check. Area constraint requires the object to be of minimum allowed area so that noise is avoided. Let the minimum allowed area be represented as $T_a$, the area of the object pixel $O_i$ be represented as $A_i$, then the criteria is stated as in equation Step 3 (f)

Let the objects detected be represented as $O = \{O_1, O_2, ..., O_m\}$. For an object let the extreme corner points be represented by $p_{x,y}, q_{x,y}, r_{x,y}$ and $s_{x,y}$. Then the centre point $(c_{x,y})$ is assumed as in equation Step 3 (g).
The object height and width is calculated. Height is calculated as the difference between top and bottom edge. Similarly, width is calculated as the difference between right and left edge. Suppose the left and right edges are denoted by $A_R$ and $B_L$, the top and bottom edges are represented by $A_T$ and $B_B$. Width and height can be computed as in equation Step 3(h).

From the detected objects, it is represented by $O_i; \, \text{for} \, 0 < i \leq M$ and each object is defined by $O_i = \{c_{x,y}^{(i)}, W_i, H_i\}$. After the clustering process, classification of images is carried out through neuro fuzzy classifier.

**Step 4: Neuro Fuzzy classifier.**

The steps involved in the neuro fuzzy classifier are as follows.

\[(a) \quad f(x) = \begin{cases} 
0 & \text{if } x \leq p \\
(x - p) & \text{if } p \leq x \leq q \\
(q - p) & \text{if } q \leq x \leq r \\
(r - x) & \text{if } r \leq x \leq r \\
0 & \text{if } x \geq r
\end{cases}\]

\[(b) \quad BP_{err} = C_{tar} - C_{out}\]

The network output is determined by $C_{out} = [Y_2^{(1)} \, Y_2^{(2)} \ldots Y_2^{(N)}]$, $Y_2^{(1)}, \, Y_2^{(2)}, \ldots, Y_2^{(N)}$ are the network outputs.

\[(c) \quad Y_2^{(i)} = \sum_{r=1}^{N_r} w_{i,r} Y_1(r)\]

Where

$Y_1(r) = \frac{1}{1 + \exp(-w_{1r} \cdot C_{in})}$

\[(d) \quad \Delta w = \gamma \cdot Y_2 \cdot BP_{err}\]

where $\gamma$ is the learning rate, usually it ranges from 0.2 to 0.5.

\[(e) \quad P_{\text{max}} = \frac{P_S}{P_T}\]

\[(f) \quad \text{fitness} = \max \text{ inum popularity} = P_{\text{max}}\]

Where,

$P_S$ - signifies the selected population

$P_T$ - represents the total population
Where, L represents the constants K
Symbolizes the current generation

Sugeno FISTYPE is used in our proposed method. The inputs, outputs and their ranges are specified. The membership values are computed as in equation Step 4 (a). The numerical attributes into the fuzzy was changed, by means of the fuzzy Membership principle. The optimization procedure was used to optimize the fuzzy rules. The rules are optimized with the help of Evolutionary programming.

According to the rules, the neural network layers are constructed. Arbitrary weights are generated within the interval [0, 1] and assigned to the hidden layer neurons as well as the output layer neurons. Unity value in weight is maintained for all neurons of the input layer. The dataset was input into the classifier and the Back propagation error was determined as in equation Step 4 (b). The network outputs can be determined as in equation Step 4 (c). The weights are adjusted for all the neurons by \( w = w + \Delta w \), where \( \Delta w \) is the change in weight which can be determined as in equation step 4 (d).

In the proposed method, an innovative technique is used to optimize the weights with the help of the MCS (Modified Cuckoo Search Algorithm). The Cuckoo search algorithm represents a meta-heuristic algorithm which owes its origin to the breeding conduct of the cuckoos and it is easy for implementation. There is a multitude of nests in the cuckoo search. Each egg signifies a solution and an egg of cuckoo corresponds to a novel solution. The novel and superior solution replaces the most horrible solution in the nest. The ordinary cuckoo search algorithm is modified by including Gauss distribution in the updation phase. The gauss distribution adds better results for optimization.

The population \((m_i, \text{ where } i=1, 2, n)\) of host nest is initiated arbitrarily. With the help of the levy flights, a cuckoo is selected randomly which generates novel solutions. Subsequently, the engendered cuckoo is evaluated by employing the objective function for ascertaining the excellence of the solutions. The fitness function is evaluated in accordance with Equations as in step 4 (e) and (f), and the best one is selected.
At the outset, the solution is optimized by the levy flights by employing the cosine transform. The quality of the novel solution is evaluated and a nest is selected arbitrarily from among them. If the quality of novel solution in the selected nest is superior to the previous solution, it is replaced by the novel solution (Cuckoo). Otherwise, the previous solution is treated as the best solution. The levy flights employed for the general cuckoo search algorithm is expressed by the Equation as in step 4 (g). By suitably adapting equation step 4 (g). Levy flight equation using the gauss distribution is exhibited in equation step 4 (h). The worst nests are ignored, in accordance with their possibility values and novel ones are constructed. Subsequently, depending upon their fitness function the best solutions are ranked. Thereafter, the best solutions are detected and marked as optimal solutions.

**SUMMARY**

In this chapter, the characters from the vehicle number plate images are recognized with the help of neuro fuzzy classifier. The desired features obtained after bi-partitioning was clustered with the help of modified fuzzy C means clustering. In this case the objective function is modified suitably to obtain better results. The clustered features were then fed into the hybrid neuro fuzzy classifier for the recognition of characters. The proposed method 3 was compared with three different approaches, ANN classifier, Fuzzy classifier and KNN classifier. The result of the proposed method was discussed in chapter 6.