Introduction

Our thesis is about the Automatic License Plate Recognition with Template Matching Technique. Previously, we worked on English Dataset and now we are working on Bangla Dataset. Our main goal is to recognize the license plate of the Bangladeshi vehicles by different types of approaches.

Description

At the time of first defense, we worked on English dataset. After that, we started working on Bangla dataset. At first we collected the images of the cars with Bengali license plates.

After collecting the images, we made a CSV file which consists of the important criterion of the images individually. After completing the CSV file, we imported it into the python file. Then we trained the dataset and successfully tested it in a fixed ratio which gave us an accuracy score in percentage (%).

Requirements

To complete this activity, we used some hardware and software tools which are necessary.

1. Software Requirements:

- Python 3.7 with Modules
- Spyder 3 IDE
- MATLAB R2018a

2. Hardware Requirements:

- Samsung DUOS with 5 Megapixel camera

Methodology

The task is started form the dataset collection and terminated in finding the perfect accuracy score. The cycle diagram of the performed task is given below:

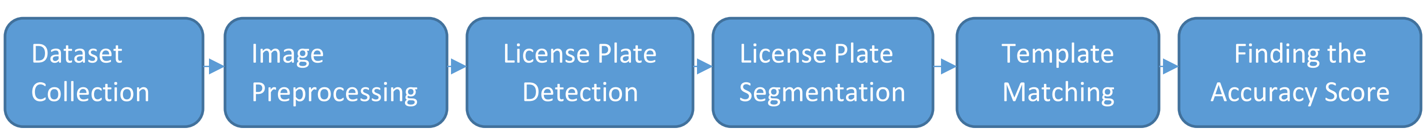


Figure: Methodology of license plate recognition from Bangladeshi vehicles

Dataset Collection

We collected the dataset by capturing images of the cars and other vehicles like motor cycles, trucks etc. There are some images of army cars in our collected dataset.

The images are captured with a SAMSUNG DUOS mobile phone with 5 Megapixel camera. We captured most of the images from the different angles of the different types of vehicles available in the Dhaka city. Some images are captured from the parking of our area. And the other images are collected from the internet.

We collected 170 images totally. But we successfully worked on 162 images and the rest of the images have some problems to work on it. All the images of different vehicles are captured from the different areas of Dhaka city by us.

Image Preprocessing

After collecting the dataset, we performed the preprocessing of the images first. To preprocess the images, we made all the images in the same resolution so that we made them in the same height and width.

License Plate Recognition

The license plate of Bangladeshi vehicles can be recognized in different ways. But we recognized the license plate of Bangladeshi vehicles by detection, segmentation and template matching techniques. All the techniques are interrelated to each other. The license plate is detected first. After detection, it is separated into different segments. After the segmentation, it is recognized by template matching technique. If the license plated cannot be detected, it can never be segmented. If the number plate is not segmented, it can never be recognized by template matching technique. So, we recognized the license plate by three consecutive approaches.

License Plate Detection

The license plate of Bangladeshi vehicles are detected in different approaches. The stepwise operations on the images of our collected dataset helps to detect the license plate from the images. The stepwise operations are noise filtering, region of interest (ROI), intensity transformation, edge detection, threshold, dilation, erosion and skewness correction. All the operations are describes briefly in the below.

i. Noise Filtering

To remove the noise, we used mean or average filtering which is very effective for the noise removal. Because mean filtering is a simple, intuitive and easy to implement method of smoothing images, i.e. reducing the amount of intensity variation between one pixel and the next. It is also known as Box filter or Average filter. It is often used to reduce noise in images.

The idea of mean filtering is simply to replace each pixel value in an image with the mean or average value of its neighbors, including itself. This has the effect of eliminating pixel values which are unrepresentative of their surroundings. Mean filtering is usually thought of as a convolution filter. Like other convolutions it is based around a kernel, which represents the shape and size of the neighborhood to be sampled when calculating the mean. Often a 3×3 square kernel is used, as shown in Figure 1, although larger kernels (e.g. 5×5 squares) can be used for more severe smoothing. (Note that a small kernel can be applied more than once in order to produce a similar but not identical effect as a single pass with a large kernel.)

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Figure 1: 3×3 averaging kernel, which is often used in mean filtering

There are some properties of mean filtering. They are given below.

- It must be odd numbered
- The sum of all the elements should be 1
- All the elements should be same
- If we follow this rule, then for a mask of 3x3. We can get the following result.

Since it is a 3x3 mask that means it has 9 cells. The condition that all the elements sum should be equal to 1 can be achieved by dividing each value by 9. So,

$$1/9 + 1/9 + 1/9 + 1/9 + 1/9 + 1/9 + 1/9 + 1/9 + 1/9 = 9/9 = 1$$

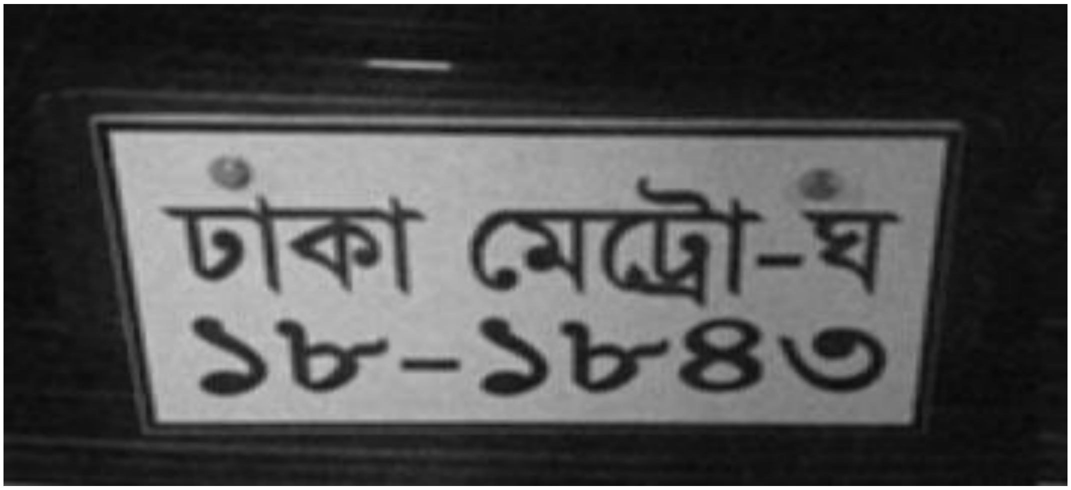
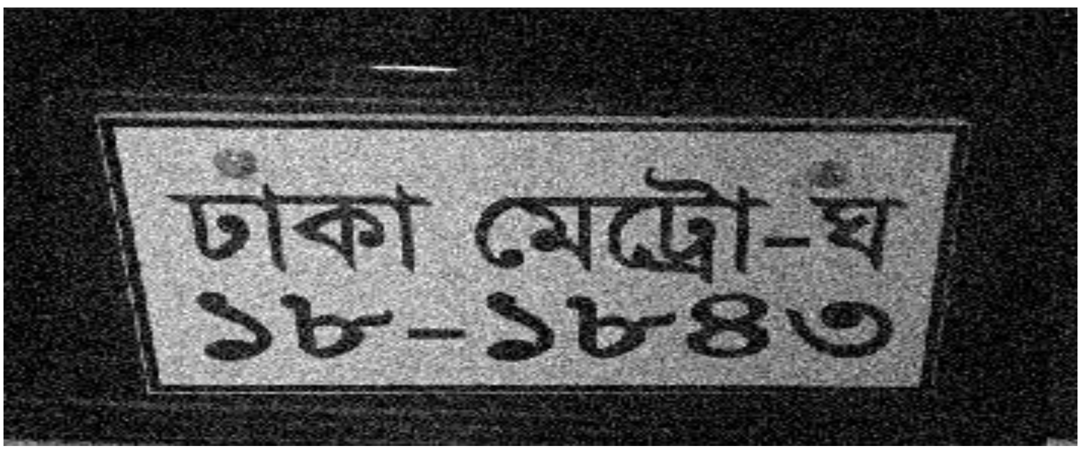


Figure: Original Image



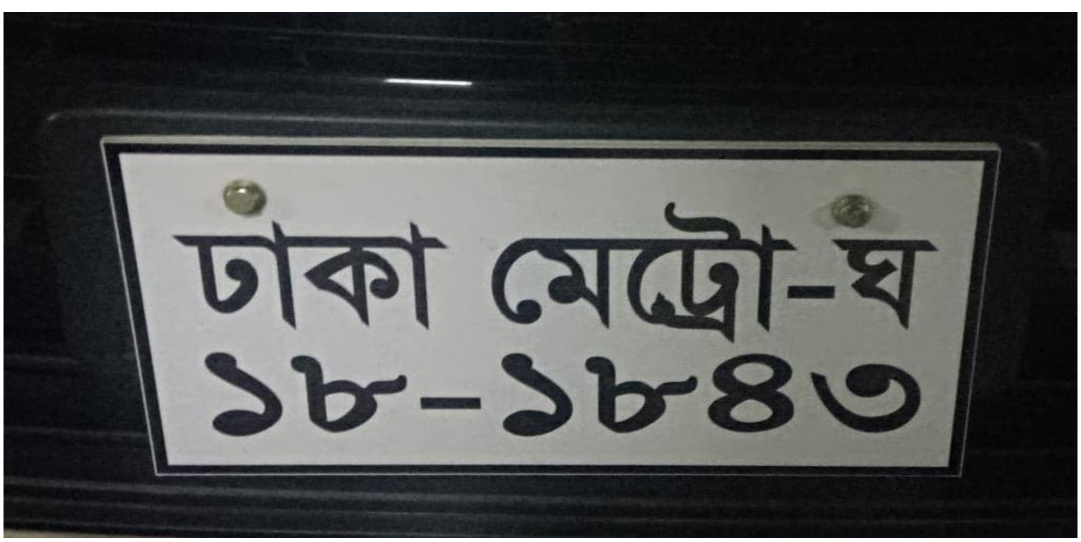


Figure: Filtered Image (Average Filtering)

The significance of Average Filtering in the license plate detection from Bangladeshi vehicles is given below.

- To remove the noise very smoothly and sharpen the image. This type of filtering is more effective than the others.
- Where median filtering needs sorting to find the median value for the filtering, but mean or average filtering needs no sorting technique. It only needs the addition and division approach to perform the arithmetic average for the filtering.

So, mean filtering is simpler and easier to implement for noise filtering.

ii. Region of Interest (ROI)

It is sometimes of interest to process a single sub-region of an image, leaving other regions unchanged. This is commonly referred to as region-of-interest (ROI) processing. A region of interest (ROI) is a portion of an image that you want to filter or perform some other operation on.

For the license plate detection from Bangladeshi vehicles, we captured the image of the license plates with the cars and other materials of the environment. But there are some unnecessary objects in our captured image, which can be problematic for our license plate detection. So, we need to select the region which we want to detect properly i.e. the license plate of the vehicles. So, the license plates of different vehicles are the region-of-interest (ROI). So, we need to extract it because of detecting the license plate properly from the original image.

Now, an example of applying the region-of-interest in an original image of our dataset is given below.

Figure: Original Image

Figure: The original image after ROI extraction

The significances of applying the Region-of-interest extraction in our collected images are given below.

- The defined region is the license plate of the main image. So, the unnecessary objects of the original image are removed
- Possible to detect the license plate perfectly from the captured image by eliminating the external objects in this approach.
- Easy to recognize the license plate very accurately from the captured image by eliminating the external objects in this approach

So, it is necessary to apply the region-of-interest (ROI) extraction to our collected images to detect and recognize the license plate properly.

iii. Intensity Transformation

Intensity transformation is a very basic image processing task that defines us to have a better subjective adjustment over the images. Enhanced images provide better contrast of the details that images contain. It is applied in every field where images ought to be understood and analyzed.
It simply means, transforming an image I into image G using T, where T is the transformation. The values of pixels in image I and G are denoted by r and s respectively. As said, the pixel values r and s are related by the expression,

$$s=T(r)$$

Where T is a transformation that maps a pixel value r into a pixel value s. The results of this transformation are mapped into the grayscale range. So, the results are mapped back into the range [0, L-1], where L=2k, k being the number of bits in the image considered. So, for instance, for an 8 bit image the range of pixel values will be [0,255].

There are three basic types of transformations that are used frequently in image enhancement. They are Linear, Log and Power-Law transformation.
For license plate detection of the Bangladeshi vehicles, we need to use Power-Law transformation or Gamma transformation to transform the intensity of the image in order to brighten the selected image. Because, Gamma transformation is more effective than the other transformation to detect the license plate from the Bangladeshi vehicles properly. The mechanism of power-law or Gamma transformation is described below.

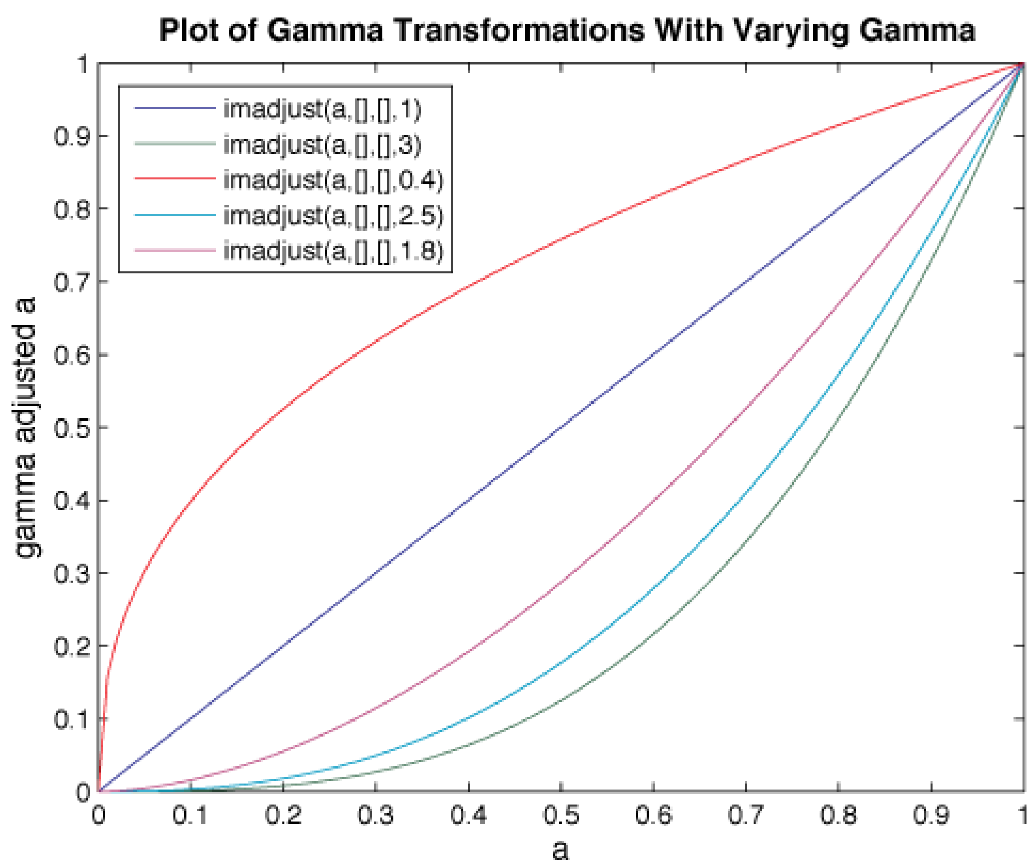


Figure: Power-Law transformation or Gamma transformation

The nth power and nth root curves in the figure can be given by the expression,

$$s=cr^{\gamma}$$

This transformation is also called as gamma correction. For various values of γ, different levels of enhancements can be obtained. This technique is quite commonly used as Gamma Correction. Different display monitors display images at different intensities and clarities. That means, every monitor has built-in gamma correction in

it with certain gamma ranges and so a good monitor automatically corrects all the images displayed on it for the best contrast to give user the best experience.
The main difference between log transformation and power-law transformation is that using the power-law transformation curves can be obtained by varying the ɣ.
Now, an example of applying the power-law transformation in an original image of our dataset is given below.

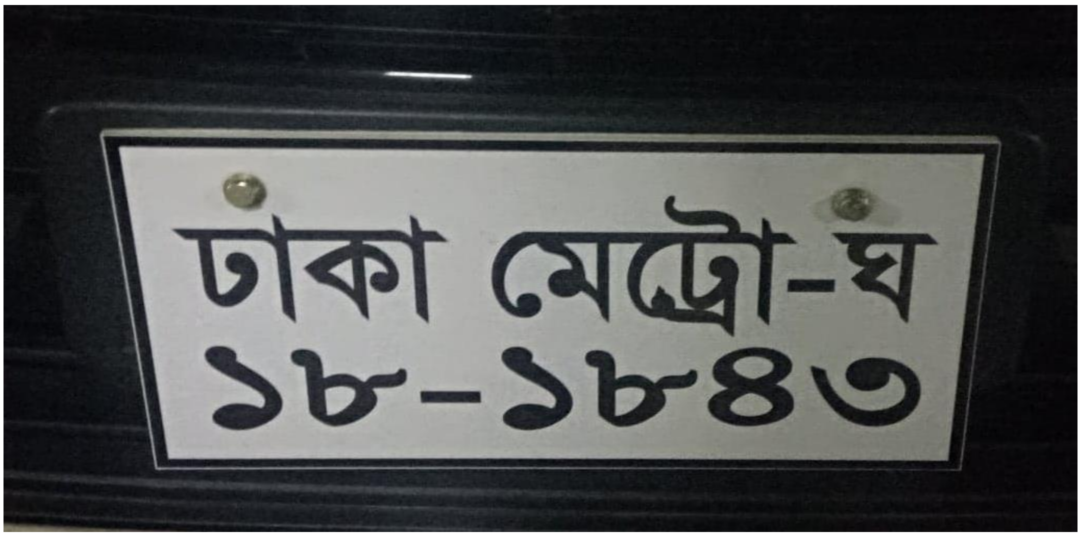


Figure: Original Image

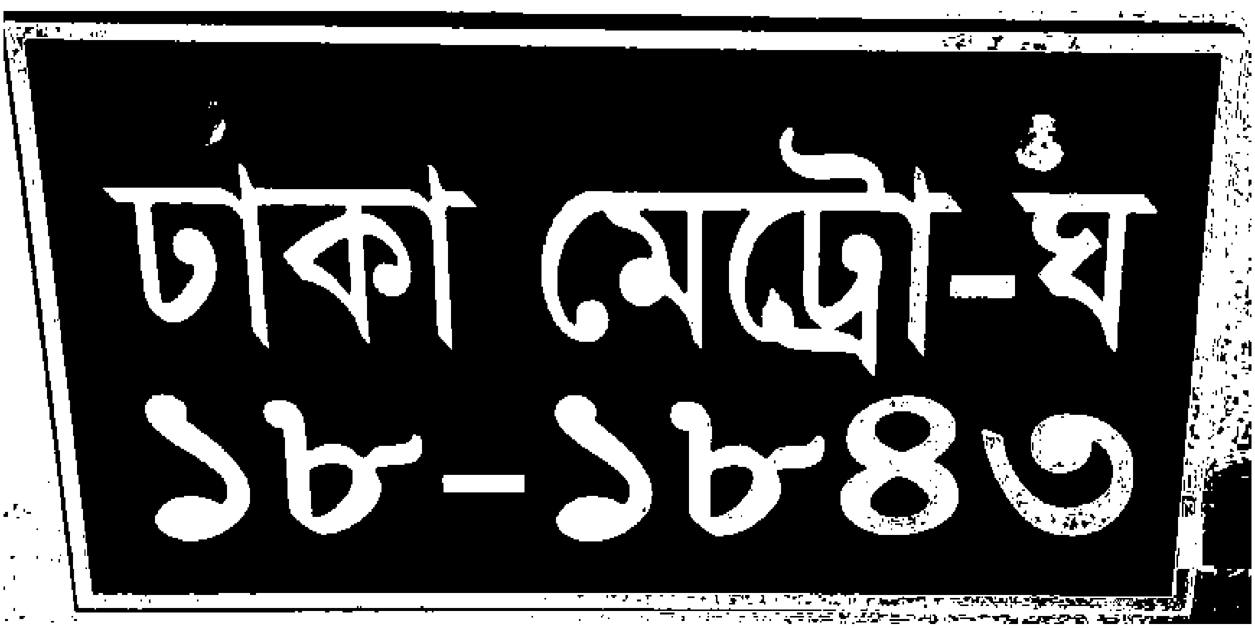


Figure: The original image after applying Power-Law transformation

The significance of applying power-law transformation on the images of the license plates is given below.

- If the image of the license plate is too much bright, the values greater than one create a darker image of the license plates.
- If the image of the license plate is too much bright, the values between zero and one create a brighter image of the license plates with more contrast in dark areas

iv. Thresholding

Thresholding is a simple process to separate the interested object from the background. The formula for achieving thresholding is as follows,

s=0; if r<=t

s=L-1; if r>t

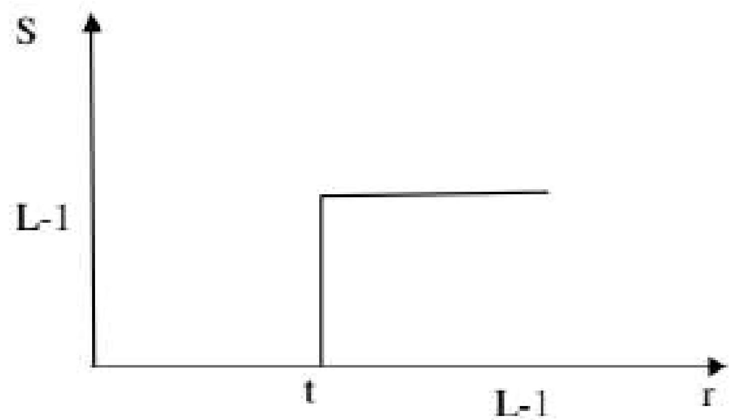


Figure: Thresholding

In details, an image processing method that creates a binary image based on setting a threshold value on the pixel intensity of the original image. While most commonly applied to grayscale images, it can also be applied to color images. Image thresholding is a simple, yet effective, way of partitioning an image into a foreground and background. This image analysis technique is a type of image segmentation that isolates objects by converting grayscale images into binary images.

It is one of the simplest method to separate regions which are higher than the set threshold. Simple or Global thresholding, where one provides the threshold value as an input constant. This threshold is applied for all pixels of the image. The mechanism of thresholding is described below.

Automatic thresholding is a great way to extract useful information encoded into pixels while minimizing background noise. First it selects initial threshold value, typically the mean 8-bit value of the original image and divides the original image into two portions. If the pixel values that are less than or equal to the threshold, this portion is called background. If the pixel values are greater than the threshold, this portion is called foreground. Then it finds the average mean values of the two new images. And it calculates the new threshold by averaging the two means. If the difference between the previous threshold value and the new threshold value are below a specified limit, our task is finished. Otherwise we need to apply the new threshold to the original image in order to keep trying.

The input to a thresholding operation is typically a grayscale or color image. In the simplest implementation, the output is a binary image representing the segmentation. In a single pass, each pixel in the image is compared with this threshold.

Now, an example of applying threshold in an original image of our dataset is given below.

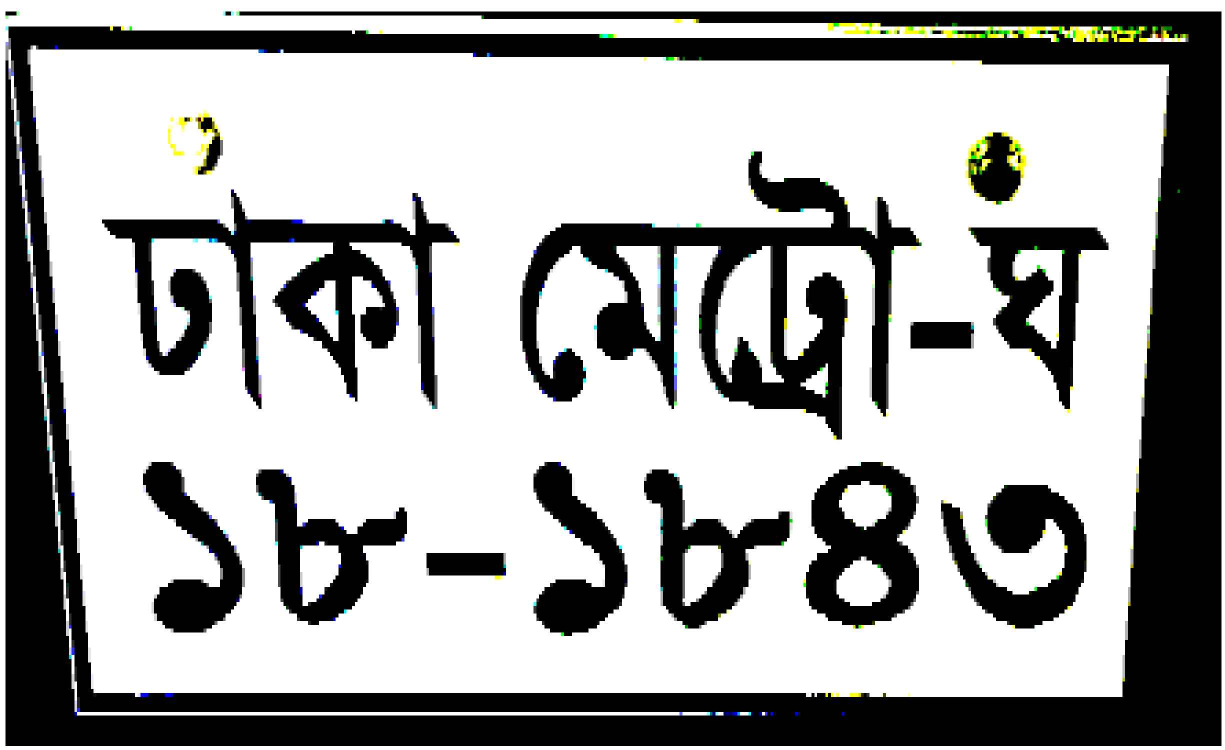


Figure: Thresholded Image

The significance of applying the threshold on the images of the license plates of Bangladeshi vehicles is given below.

- It is the simplest process to separate the license plate from the background
- It gives the binary image of the license plate so that we can recognize it very perfectly.

So, it is necessary to apply thresholding on the images of the license plates of Bangladeshi vehicles.

v. Edge detection

Most edge detection methods work on the assumption that the edge occurs where there is a discontinuity in the intensity function or a very steep intensity gradient in the image.
There are many methods of detecting edges. The majority of different methods may be grouped into these two categories. They are gradient method and laplacian method.

The gradient method detects the edges by looking for the maximum and minimum in the first derivative of the image. For example Roberts, Prewitt, Sobel where detected features have very sharp edges.

The Laplacian method searches for zero crossings in the second derivative of the image to find edges e.g. Marr-Hildreth, Laplacian of Gaussian etc. An edge has one dimensional shape of a ramp and calculating the derivative of the image can highlight its location.

For the license plate detection of Bangladeshi vehicles, we need to use the sobel operator, which is a gradient method used to detect the edges of the license plate very smoothly so that it can be detected perfectly. The Sobel operator is a discrete differentiation operator, computing an approximation of the gradient of the image intensity function. The mechanism of sobel operator is described below.

Using this assumption, if one take the derivative of the intensity value across the image and find points where the derivative is maximum, then the edge could be located. The gradient is a vector, whose components measure how rapid pixel value are changing with distance in the x and y direction.
Thus, the components of the gradient may be found using the following approximation:

$$Gx= \frac{\delta f(x,y)}{\delta x} = \frac{f(x+dx,y)-f(x,y)}{dx}$$

$$Gy= \frac{\delta f(x,y)}{\delta y} = \frac{f(x,y+dy)-f(x,y)}{dy}$$

Where, dx and dy measure distance along the x and y directions respectively. In discrete images, one can consider dx and dy in terms of numbers of pixel between two points. dx = dy = 1 (pixel spacing) is the point at which pixel coordinates are(i, j) thus,

$$Gx = f(i + 1, j) - f(i, j)$$
$$Gy = f(i, j + 1) - f(i, j)$$

In order to detect the presence of a gradient discontinuity, one could calculate the change in the gradient at (i, j) .This can be done by finding the following magnitude measure.

$$G=\sqrt{Gx^2 + Gy^2}$$

And the gradient direction θ is given by $\theta=\tan^{-1}(\frac{Gy}{Gx})$

A 3x3 region of an image the sobel edge detection filter can be used

-1	-2	-1
0	0	0
1	2	1

-1	0	1
-2	0	2
-1	0	1

Figure: Sobel Filter

Now, an example of applying sobel operator in an original image of our dataset is given below.

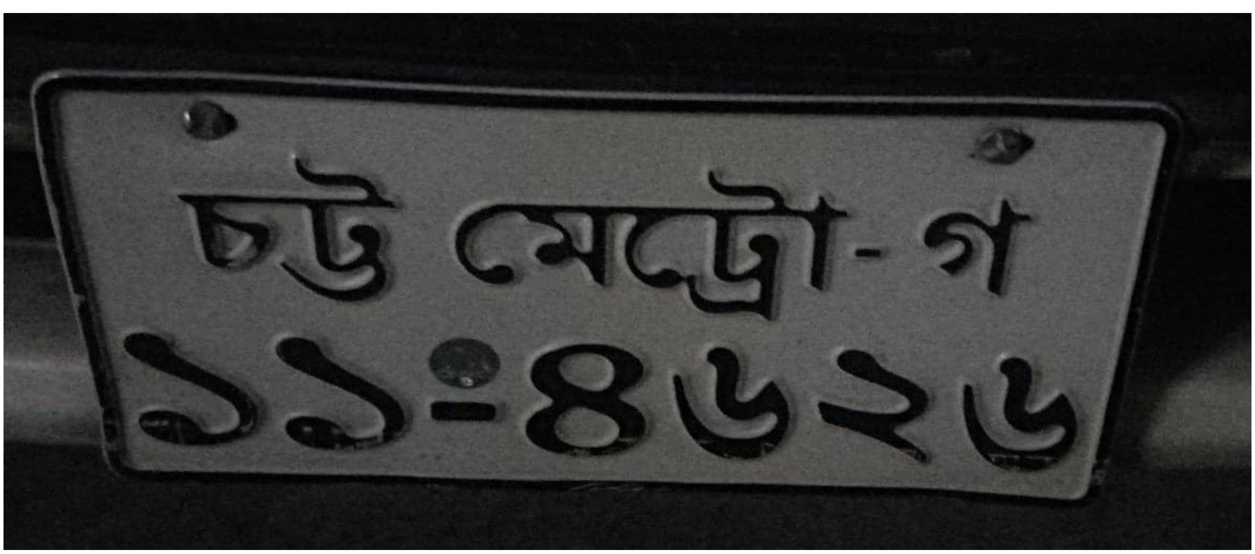


Figure: Original Image

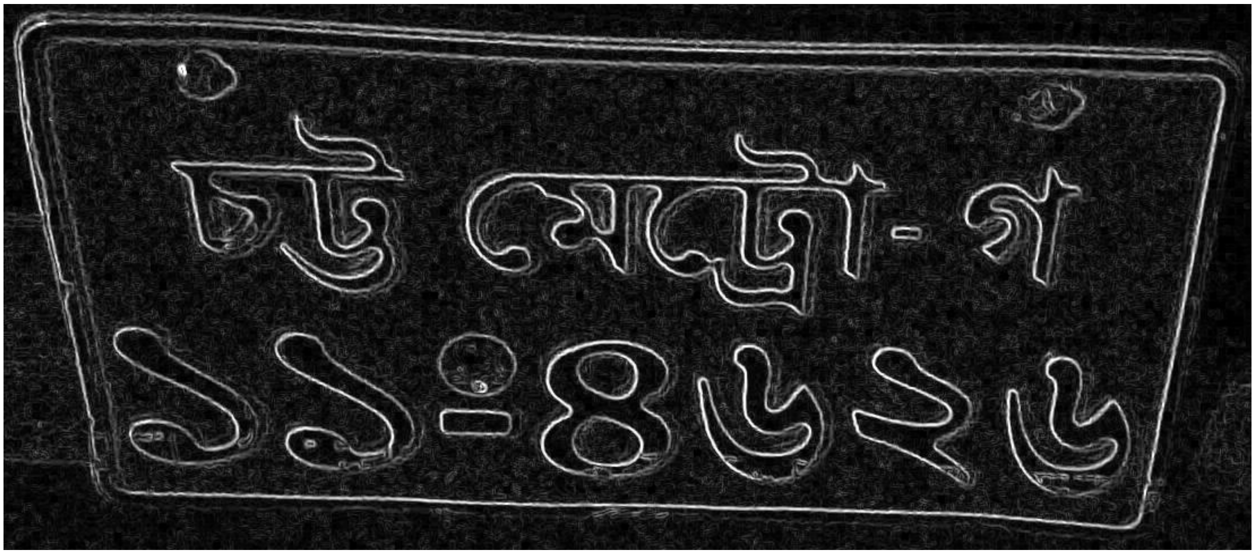


Figure: Original Image after Sobel Edge Detection

The significance of applying sobel edge detection approach to detect the license plate of Bangladeshi vehicles is given below

- To detect the features which have the sharp edges
- Features having sharp edges are detected very accurately and perfectly

So, it is necessary to apply sobel operator on the images to detect the edges of the license plates of Bangladeshi vehicles.

vi. Skewness Correction

Skewness is a measure of symmetry, or more precisely, the lack of symmetry. A distribution, or data set, is symmetric if it looks the same to the left and right of the center point. The skewness for a normal distribution is zero, and any symmetric data should have a skewness near zero.

Negative values for the skewness indicate data that are skewed left and positive values for the skewness indicate data that are skewed right. By skewed left, we mean that the left tail is long relative to the right tail. And if the skewness is zero, it indicates that there is no skewness in the data.

At the time of dataset collection, we captured different types of images from different positions, different distances and different angles. Since of the images are captured from different angles, so most of the images are skewed. So, we need to correct the skewness of the images of the license plates contained in the dataset.

First of all, we need to detect the skewness of the image of the license plates of Bangladeshi vehicles. If the skewness is zero, it means that there is no skewness in the

image. If the skewness is anything but zero, it means that there is skewness remaining in the image. So, no skewness is detected in the image.
If the skewness is positive, it means that the image is skewed in the right position. If the skewness is negative, it means that the image is skewed in the left position. It is the skewness detection of an image.
After detecting the skewness of an image, we need to correct the skewness. We have to correct the position of the image by the skewness angle. So, we can detect and correct the skewness of the images of license plates of Bangladeshi vehicles.

Now, an example of applying the skewness correction in an original image is given below.

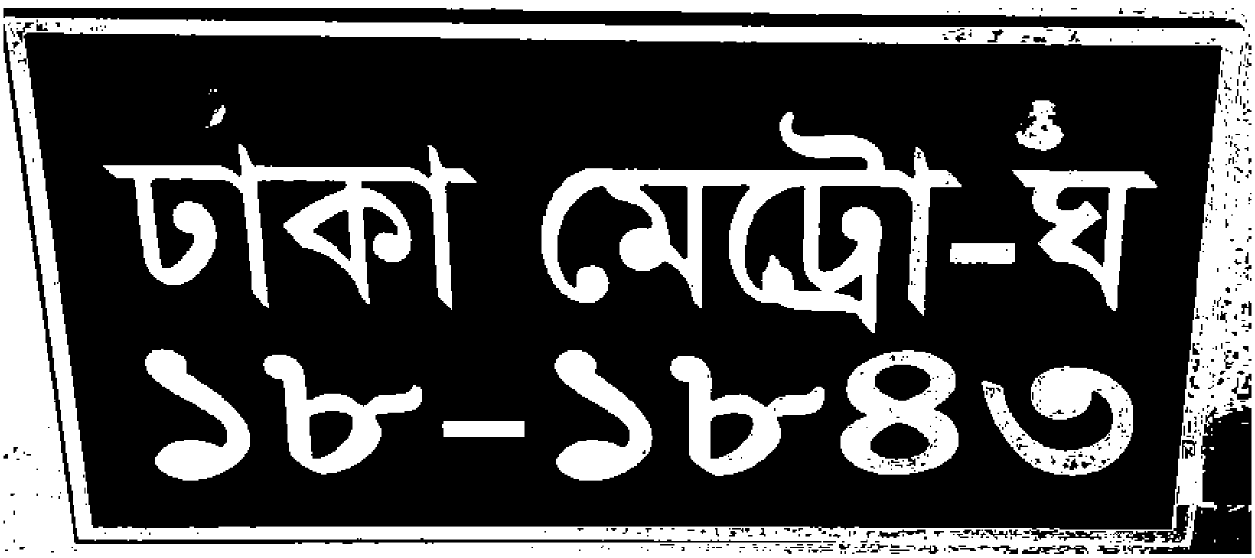


Figure: Original Image before skewness correction

Figure: The image after correcting the skewness

The significance of skewness correction of an image is given below.

- Skewed images are very difficult to be detected. So, we need to correct the skewness of the images to detect it properly.
- We need to recognize the license plate from the vehicle images. But it is very problematic to recognize the license plate from an image of a Bangladeshi vehicle which is skewed. So, we need to correct the skewness of the license plate images.

So, we have to correct the skewness for the perfect detection and recognition of the license plate images.

vii. Morphological Operations

Morphology is a pixel shape based analysis. It is a tool for extracting image components that are useful in the representation and description of an image. It deals with the shape of objects in an image. The morphological operation is related to the basic set theory.
For the license plate detection of Bangladeshi vehicles, we need to use the morphological operations because of growing and shrinking the image regions. Because of this, we need to apply the dilation and erosion technique in order to detect the license plate properly.

There is no necessity of opening and closing operations in our images to detect the vehicle license plates.
The mechanism of dilation and erosion are described below.

Dilation

Dilation is one of the two basic operators in the area of mathematical morphology, the other being erosion. It is typically applied to binary images, but there are versions that work on grayscale images.

The basic effect of the operator on a binary image is to gradually enlarge the boundaries of regions of foreground pixels (*i.e.* white pixels, typically). Thus areas of foreground pixels grow in size while holes within those regions become smaller. The mechanism of dilation is described below.

The dilation operator takes two pieces of data as inputs. The first is the image which is to be dilated. The second is a usually small set of coordinate points known as a structuring element, which is also known as a kernel.

It is this structuring element that determines the precise effect of the dilation on the input image. The mathematical definition of dilation for binary images is as follows:

Suppose that *X* is the set of Euclidean coordinates corresponding to the input binary image, and that *K* is the set of coordinates for the structuring element.

Let *Kx* denote the translation of *K* so that its origin is at *x*.

Then the dilation of X by K is simply the set of all points x such that the intersection of Kx with X is non-empty.

The mathematical definition of grayscale dilation is identical except for the way in which the set of coordinates associated with the input image is derived. In addition, these coordinates are 3-D rather than 2-D.

As an example of binary dilation, suppose that the structuring element is a 3×3 square, with the origin at its center, as shown in the table. Note that in this and subsequent diagrams, foreground pixels are represented by 1's and background pixels by 0's.

The basic rule for dilation is, if any of the pixel (in the neighborhood defined by structural element) is 1, then the output is 1.

To compute the dilation of a binary input image by this structuring element, we consider each of the background pixels in the input image in turn. For each background pixel (which we will call the input pixel) we superimpose the structuring element on top of the input image so that the origin of the structuring element coincides with the input pixel position.

If at least one pixel in the structuring element coincides with a foreground pixel in the image underneath, then the input pixel is set to the foreground value. If all the corresponding pixels in the image are background, however, the input pixel is left at the background value.

For our example 3×3 structuring element, the effect of this operation is to set to the foreground color any background pixels that have a neighboring foreground pixel (assuming 8-connectedness). Such pixels must lie at the edges of white regions, and so the practical upshot is that foreground regions grow (and holes inside a region shrink).

Dilation is the dual of erosion. Dilating foreground pixels is equivalent to eroding the background pixels.

1	1	1
1	1	1
1	1	1

Figure: A 3x3 square structuring element

Most implementations of this operator expect the input image to be binary, usually with foreground pixels at pixel value 255, and background pixels at pixel value 0.

Such an image can often be produced from a grayscale image using thresholding. It is important to check that the polarity of the input image is set up correctly for the dilation implementation being used.

The structuring element may have to be supplied as a small binary image, or in a special matrix format, or it may simply be hardwired into the implementation, and not require specifying at all.

In this latter case, a 3×3 square structuring element is normally assumed which gives the expansion effect described above.

The 3×3 square is probably the most common structuring element used in dilation operations, but others can be used. A larger structuring element produces a more extreme dilation effect, although usually very similar effects can be achieved by repeated dilations using a smaller but similarly shaped structuring element.

With larger structuring elements, it is quite common to use an approximately disk shaped structuring element, as opposed to a square one.

For the license plate detection of Bangladeshi vehicles, we need to use dilation in order to grow the regions of the images of the vehicle license plates so that it can enlarges the foreground and shrinks the background of the images of vehicle license plate. It also helps to expand the size of the foreground objects. It smooths the boundaries of the objects situated in the vehicle license plate image.

Again, at the time of license plate detection, we found some gaps and holes situated in the license plate, which is very problematic to detect the license plate accurately. But after dilating the image of the vehicle license plate, these holes and gaps are closed completely. So, we can detect it very accurately.

Now, an example of applying dilation in an original image is given below.

Figure: Image before dilation

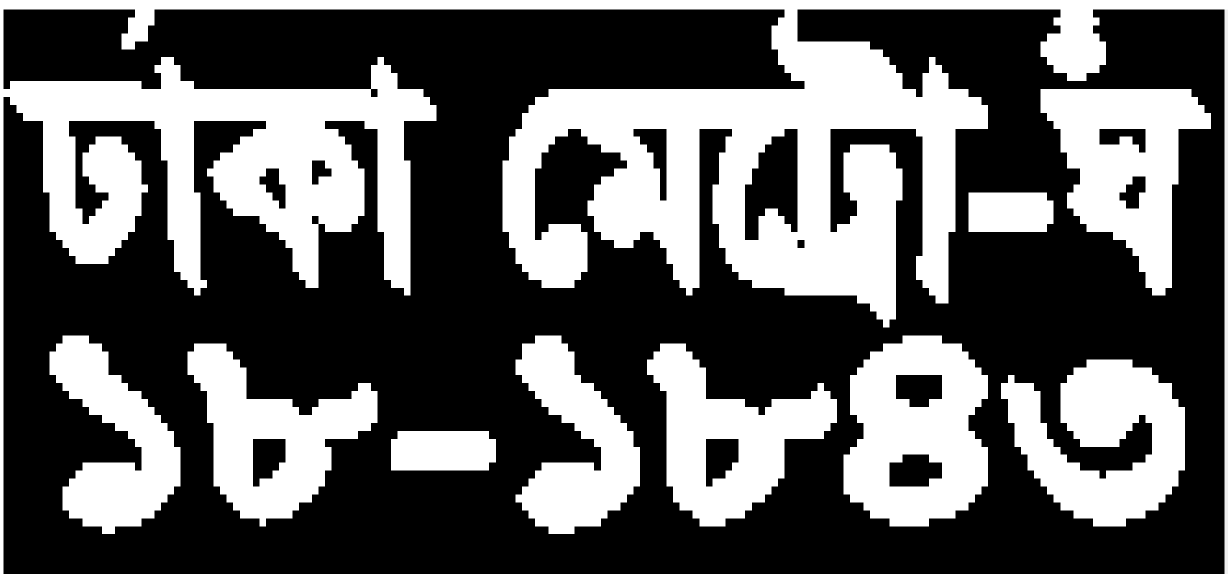


Figure: Image after dilation

The significance of applying dilation of an image is given below.

- We have some images in the dataset, which indicates that some small holes and gaps are situated in the features of the license plate. In order to close the holes and gaps, we need to dilate the image of vehicle license plate.

- In order to smooth the boundaries of the features of the license plates of Bangladeshi vehicles, we need to dilate it.

So, we can detect the image of vehicle license plates properly by dilating the image.

Erosion

Erosion is one of the two basic operators in the area of mathematical morphology. It is typically applied to binary images, but there are versions that work on grayscale images.

The basic effect of the operator on a binary image is to erode away the boundaries of regions of foreground pixels (i.e. white pixels, typically). Thus areas of foreground pixels shrink in size, and holes within those areas become larger. The mechanism of erosion is described below.

The erosion operator takes two pieces of data as inputs. The first is the image which is to be eroded. The second is a usually small set of coordinate points known as a structuring element which is also known as a kernel.

It is this structuring element that determines the precise effect of the erosion on the input image.

The mathematical definition of erosion for binary images is as follows:

Suppose that X is the set of Euclidean coordinates corresponding to the input binary image, and that K is the set of coordinates for the structuring element.

Let, Kx denote the translation of K so that its origin is at *x*.

Then the erosion of X by K is simply the set of all points *x* such that K*x* is a subset of X.

The mathematical definition for grayscale erosion is identical except in the way in which the set of coordinates associated with the input image is derived. In addition, these coordinates are 3-D rather than 2-D.

As an example of binary erosion, we supposed that the structuring element is a 3×3 square, with the origin at its center. Note that in this and subsequent diagrams, foreground pixels are represented by 1's and background pixels by 0's.

The basic rule for dilation is, if any of the pixel (in the neighborhood defined by structural element) is 0, then the output is 0.

To compute the erosion of a binary input image by this structuring element, we consider each of the foreground pixels in the input image in turn. For each foreground pixel (which we will call the input pixel) we superimpose the structuring element on top of the input image so that the origin of the structuring element coincides with the input pixel coordinates.

If for every pixel in the structuring element, the corresponding pixel in the image underneath is a foreground pixel, then the input pixel is left as it is. If any of the corresponding pixels in the image are background, however, the input pixel is also set to background value.

For our example 3×3 structuring element, the effect of this operation is to remove any foreground pixel that is not completely surrounded by other white pixels (assuming 8-connectedness). Such pixels must lie at the edges of white regions, and so the practical upshot is that foreground regions shrink (and holes inside a region grow).

Erosion is the dual of dilation, i.e. eroding foreground pixels is equivalent to dilating the background pixels.

1	1	1
1	1	1
1	1	1

Figure: A 3x3 square structuring element

Most implementations of this operator will expect the input image to be binary, usually with foreground pixels at intensity value 255, and background pixels at intensity value 0. Such an image can often be produced from a grayscale image using thresholding. It is important to check that the polarity of the input image is set up correctly for the erosion implementation being used.

The structuring element may have to be supplied as a small binary image, or in a special matrix format, or it may simply be hardwired into the implementation, and not require specifying at all. In this latter case, a 3×3 square structuring element is normally assumed which gives the shrinking effect described above.

The 3×3 square is probably the most common structuring element used in erosion operations, but others can be used. A larger structuring element produces a more extreme erosion effect, although usually very similar effects can be achieved by repeated erosions using a smaller similarly shaped structuring element.

With larger structuring elements, it is quite common to use an approximately disk shaped structuring element, as opposed to a square one.

For the license plate detection of Bangladeshi vehicles, we need to use erosion in order to grow the regions of the images of the vehicle license plates. It also helps to shrink the size of the foreground objects. It smooths the boundaries of the objects situated in the vehicle license plate image. It also removes small objects situated in the vehicle license plate image.

Now, an example of applying erosion in an original image is given below.

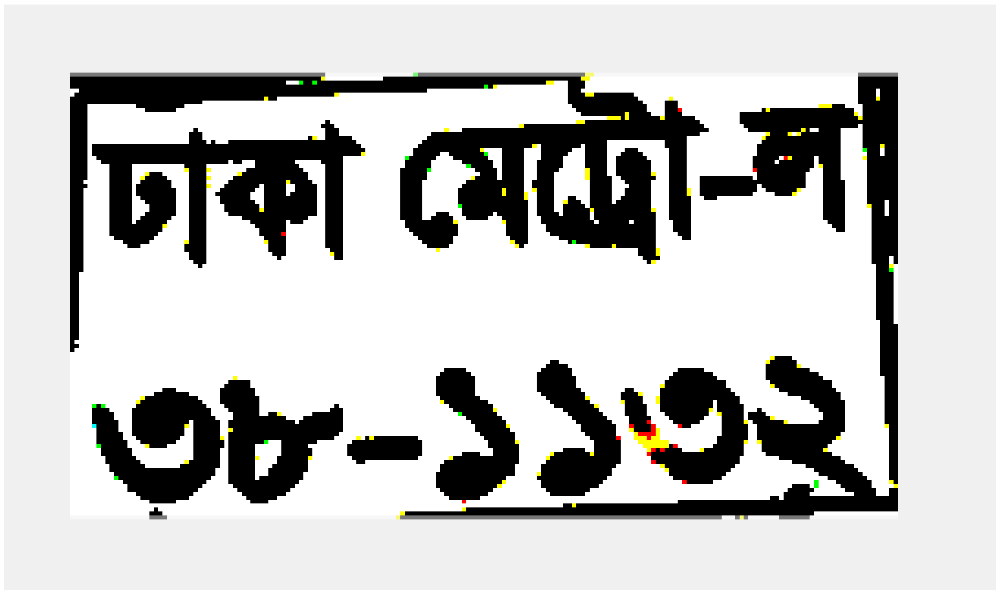


Figure: Image before erosion

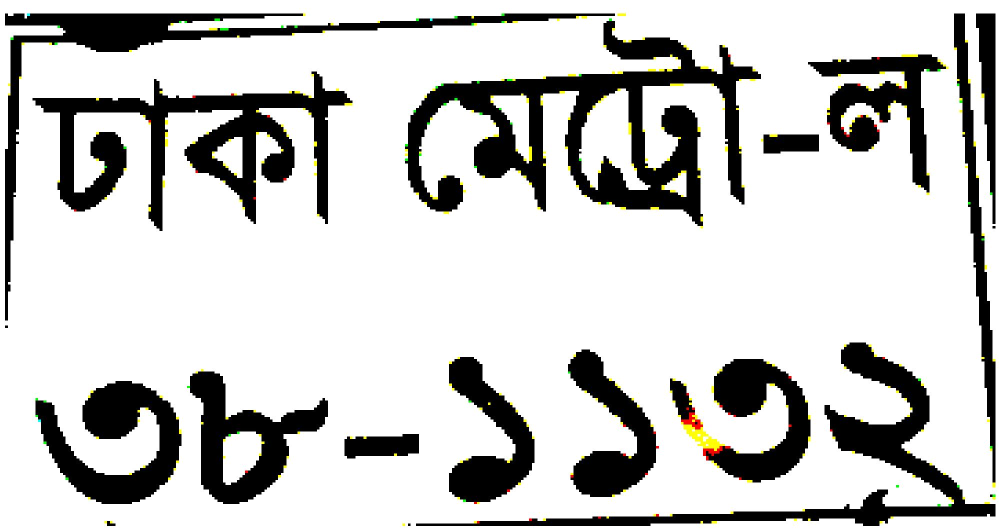


Figure: Image after erosion

The significance of applying dilation of an image is given below.

- To smooth the boundaries of the features of the license plates of Bangladeshi vehicles, we need to erode it.
- In order to close the holes and gaps situated in the features, we need to erode the image of vehicle license plate.

Since we can detect the image by dilating and eroding the images of vehicle license plates, so both dilation and erosion are necessary as morphological operation to detect and recognize the license plate of the Bangladeshi vehicles very accurately.

Image Binarization

At the time of license plate detection of the Bangladeshi vehicle from the whole image, we need to binarize that image. Image binarization means to convert the whole image into black and white only. In other words, it is the process of converting a pixel image to a binary image.

Images also need to be converted into binary in order for a computer to process them so that they can be seen on our screen. Digital images are made up of pixels. Each pixel in an image is made up of binary numbers. This data is called metadata and computers need metadata to know the size of an image.

We used Otsu binarization method to convert the image into a binary image which is used to automatically perform clustering-based image thresholding or the reduction of a graylevel image to a binary image. This algorithm assumes that the image contains two classes of pixels following bi-modal histogram (foreground pixels and background pixels), it then calculates the optimum threshold separating the two classes so that their combined spread (intra-class variance) is minimal, or equivalently because the sum of pairwise squared distances is constant, so that their inter-class variance is maximal.

Otsu's thresholding method involves iterating through all the possible threshold values and calculating a measure of spread for the pixel levels each side of the threshold, i.e. the pixels that either fall in foreground or background.

MATLAB has built-in function graythresh() in Image Processing Toolbox which are implemented with Otsu's method.

Converting an image into a binary image is necessary to detect an object from an image. And so, it is necessary for us to detect the image of the license plate form a vehicle image. Moreover, it is also necessary to divide the image into different segments and matching the template image with the main image. Because, we need to convert the image into binary image before image segmentation and template matching technique.

So, we have detected the license plates of Bangladeshi vehicles by performing the above tasks successfully. After detecting the license plate, we need to perform segmentation of the license plates of Bangladeshi vehicles.

License Plate Segmentation

After detecting the license plates of Bangladeshi vehicles successfully, we have to segment the license plates as possible as we can.

The segmentation of the license plates of Bangladeshi vehicles helps to divide the image of the license plates into different regions. It also helps to separate objects from the background and gives them individual id numbers (labels).

There are eleven features in a license plate of Bangladeshi vehicle such as private cars, motor cycles etc. They are city name, metro, letter, two hyphens and six digits. These eleven features are divided into two lines. Anyways, we need to segment the license plates into these eleven features.

First of all, we need to perform the line segmentation to divide the license plate into two lines. After line segmentation, we have to perform word segmentation to divide the lines into different words. After word segmentation, we need to perform character segmentation to divide the words into different characters. That's all about the process of license plate segmentation.

There are some images of army car in our dataset. During the segmentation of the license plate image of the army cars, we cannot perform line segmentation. Because there is a single line remaining in the image of the license plate of an army car, which consists of an arrow symbol and some digits. So, we need to perform word segmentation directly in case of the segmentation of an image of the license plate of an army car. No character segmentation is required during the segmentation of an image of the license plate of an army car.

Line Segmentation

The features of the license plates of Bangladeshi vehicles are divided two lines. City name, metro, a hyphen and letter are situated in the first line of the license plates of Bangladeshi vehicles. Again, six digits and a hyphen are situated in the second line of the license plates of Bangladeshi vehicles. It is not a great approach to segment all of the eleven features first. So, we need to perform a line segmentation to divide all the features into two segments.

We need to scan the input image horizontally to execute the license plate. Then, we can count the frequency of black pixels in each row to construct the row histogram. The position between two consecutive lines where a number of pixels in a row zero indicates a boundary between two lines.

For the license plate segmentation of the Bangladeshi vehicles, we need to convert the original image into a binary image first. After converting the original image into a binary image, we need to perform the line segmentation approach.

Figure: Original Image indicating 2 lines

The mechanism of line segmentation in the images of license plates of Bangladeshi vehicles is described below.

We used LSO algorithm to segment the images of license plates of Bangladeshi vehicles. It first divide the binary image of the license plate into two separate lines, which contains separate features.

Firstly, we calculated the average of top and bottom row of a random connected component from the binary image and save the connected component in an array.

Secondly, we checked and saved rest connected components one by one in the array along with the chosen random connected component if their number of top row is less and number of bottom row is greater than the value of average row of the chosen random connected component.

Thirdly, we reorganized the connected components of both arrays according to the ascending order of their top row number in the binary image.

Then, according to the number of top row in the binary image of first connected component of both arrays first and second line has been chosen.

Now, an example of applying line segmentation in a binary image is given below.

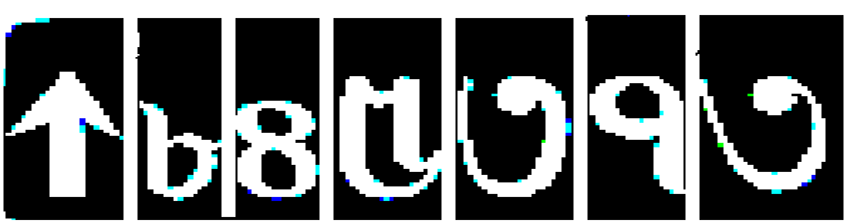
Figure: The binary image after applying line segmentation

Word Segmentation

After performing the line segmentation, we need to divide each line into different segments. This is *Word Segmentation*. In order to perform the word segmentation, we need to scan each line vertically.

The number of black pixels in each column is calculated to construct the column histogram. We used region based segmentation algorithm to perform word segmentation. The portion of each line, which remains with continuous black pixels is considered to be a word in each line. If there is no black pixel in a vertical scanning, it will be considered as the spacing between two words. And, different words in different lines are separated. So, the binary image can be considered as a collection of words.

Again, we need to perform word segmentation directly in case of the segmentation of an image of the license plate of the army cars. Only word segmentation is applied on it.



Now, an example of applying word segmentation in a binary image is given below.

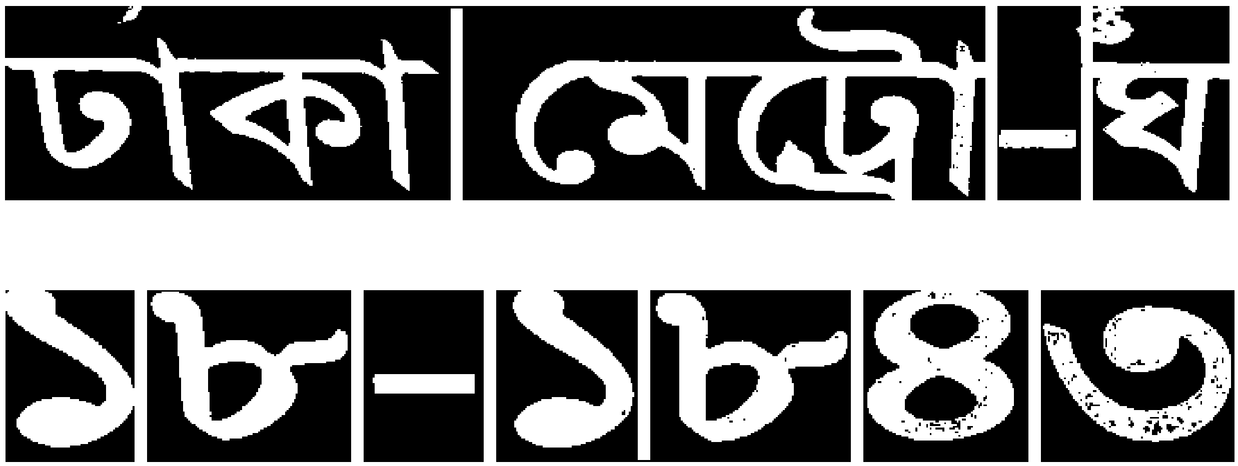


Figure: Word Segmentation in each line

Figure: Combined image of Word Segmentation in both lines

Character Segmentation

We have segmented the license plates of Bangladeshi vehicles by performing line segmentation and word segmentation that divides the license plate into different segments according to their features. Now, we need to perform character segmentation to divide the words into different characters.

We can perform the character segmentation only in city name and metro as they are associated with different characters. We have to perform some operations to divide the words into different characters. The operations are described below.

Firstly, we need to detect the zones. The texts of Bengali alphabets are partitioned into three zones.

The upper zone denotes the partition above the headline. The middle zone denotes the portion of basic characters below the headline. And the lower zone denotes the portion where some of the modifiers can reside. The imaginary line which separates the middle and lower zone is called the base line.

Now, we need to find out the headline of the word which is called 'Matra' in order to segment the character separately from the segmented word.

A row histogram is constructed by counting the frequency of each row in the word. The headline is determined by the row with the highest frequency value.

Now, after removing the headline, the characters in a word are isolated and it can easily be separated by detection of character between baseline and headline. Then, we need to eliminate the 'Matra' in order to find a differentiation line between characters successfully.

Now, we need to detect the connected components below the base line which contains the lowest point called 'Base Point'. The depth first search algorithm helps to detect the characters below the baseline successfully.

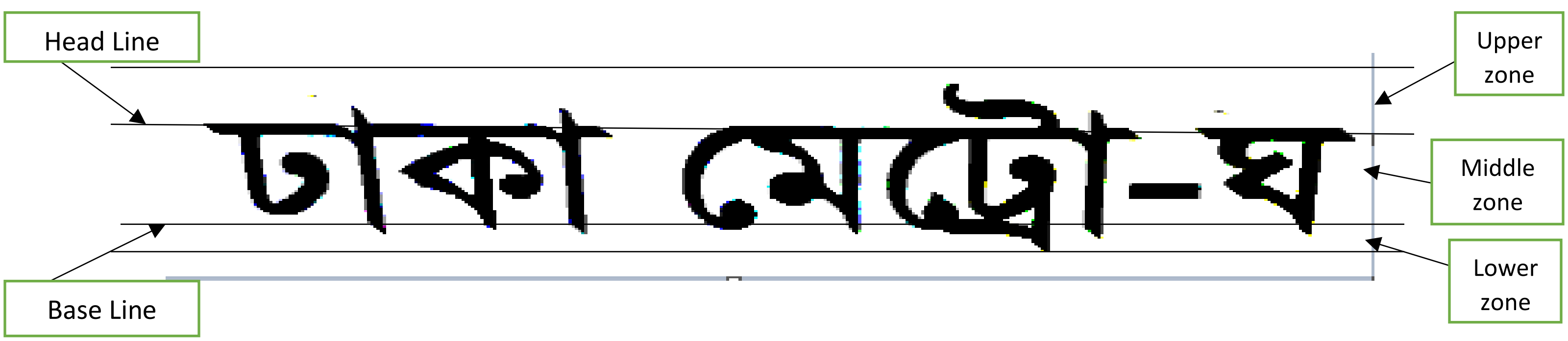


Figure: Different zones of the texts of Bengali alphabets

But we cannot perform the character segmentation perfectly because Matra Elimination is a very problematic task for Bengali alphabets. Though Matra Elimination is not an easy process, it is a very tough work to perform character segmentation without Matra Elimination. So we'll try to improve it in future.

After all, we have performed the line segmentation and word segmentation in the images of the license plates of Bangladeshi vechicles. And in the images of the license plates of the army cars, we have performed only the word segmentation. No line segmentation is performed here.

That's all about the segmentation of the license plates of Bangladeshi vehicles.

Template Matching

After performing the detection and segmentation of the license plates of Bangladeshi vehicles successfully, we need to recognize the license plates of Bangladeshi vehicles by *Template Matching* technique.

Template matching is a technique in digital image processing for finding small parts of an image which match a template image. It can be used in manufacturing as a part of quality control, a way to navigate a mobile robot, or as a way to detect edges in images.

As the templates of the license plates of Bangladeshi vehicles are not available in the internet, so we made the template images of the license plates of Bangladeshi vehicles by slicing the images. Moreover, some template images of the license plates are rare such as the license plate of army car. Since there are some images of army car in our dataset, so we used the template images which is made by slicing the images of the license plates remaining in our dataset.

Before matching the templates with an original image, we need to normalize or standardize all the images in our dataset in order to keep all of the images under same brightness. It is too problematic to match the template image with the images of different brightness. It is a process that changes the range of pixel intensity values and change the brightness or darkness of an image.

If we want to match a darker template image with a brighter image, it can never be matched. Again, if we want to match a brighter template image with a darker image, it can never be matched. The darker template image will match with a darker image and the brighter template image will match with a brighter image. That's why we need to normalize or standardize all the images to convert them in the same pixel intensity value and same brightness by changing the range of pixel intensity values and brightness.

Moreover, before applying the template matching technique, we need to convert the original image to a binary image. And we also convert the template images into binary images. As both original images and template images are converted into binary images, so now we can perform template matching technique.

We used *cosine similarity* algorithm in applying the template matching technique because this algorithm shows the perfect way to match the template image with the original image successfully.

The mechanism of template matching technique with cosine similarity algorithm is described below.

Cosine similarity is the cosine of the angle between two n-dimensional vectors in an n-dimensional space. It is the dot product of the two vectors divided by the product of the two vectors' lengths or magnitudes.

The angle between two images is inversely proportional to the similarity of that two images. If the cosine of the angle between two images decreases, the similarity of that two images will be increased. If the cosine of the angle between two images decreases, the similarity of that two images will be increased.

Again, the cosine of the angle between two images is proportional to the similarity of that two images. If the cosine of the angle between two images increases, the similarity of that two images will be increased. If the cosine of the angle between two images decreases, the similarity of that two images will be decreased.

We know that, A.B = ||A|| ||B|| cos θ

So, cos θ = $\frac{A.B}{||A||||B||}$

As similarity=cos θ, so similarity = $\frac{A.B}{||A||||B||}$.. (1)

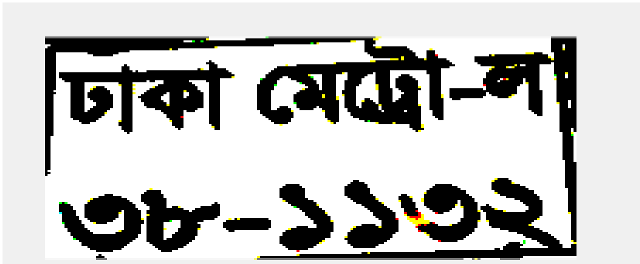
Values range between -1 and 1, where -1 is perfectly dissimilar and 1 is perfectly similar. The function is best used when calculating the similarity between small numbers of sets. The procedures parallelize the computation and are therefore more appropriate for computing similarities on bigger datasets.

In the above equation (1), we can let A as an original image and B as a template image. According to the cosine similarity, B must match with the original image A, where the similarity between B and A is the most. So, the template image matches with the original image where the similarity between the template image and the original image is the most.

At the time of license plate recognition with template matching technique, we matched the template image with the features of the license plates of Bangladeshi vehicles. We know that there are eleven features in a license plates of Bangladeshi vehicles. They are city name, metro, letter, six digits and two hyphens. We need to match all the features with the template images individually which shows a similarity or correlation result in percentage (%).

Again, at the time of license plate recognition with template matching technique we found a single template matched multiple times with an original image. In this case, the template matched multiple times with the original image shows the multiple times template matching with rectangles. But it shows the similarity or correlation result in the average of the similarity of multiple times template matching in percentage (%). And it shows this average similarity multiple times according to the number of same features.

Now, an example of applying template matching technique in a binary image is given below.



Original Image

Template Image

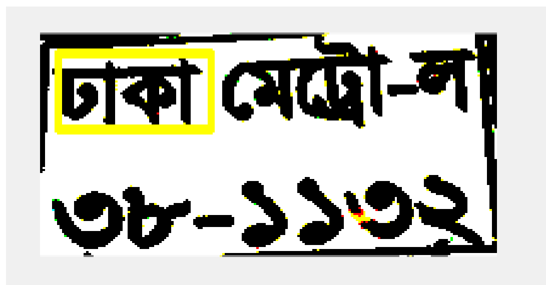
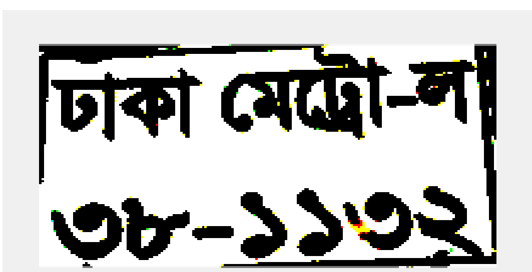
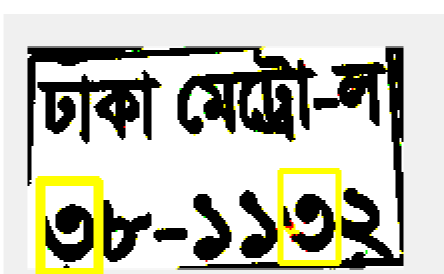


Figure: Original Image after template matching (single time matched)



Original Image

Template Images



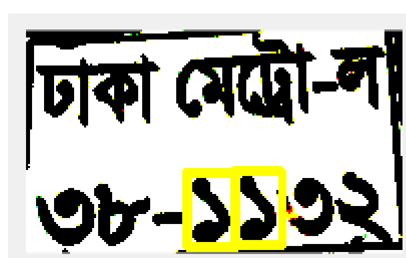


Figure: Original Image after template matching (multiple times matched)

That’s all about the template matching technique to recognize the license plates of Bangladeshi vehicles.

So, after performing the detection, segmentation and template matching technique, we can recognize the images of the license plates of Bangladeshi vehicles successfully. Now, we need to find the accuracy score to identify how accurate it is.

Finding the Accuracy Score

After recognizing the license plate of Bangladeshi vehicles successfully, we have to find the accuracy score by training and testing the data. We have tried to train and test the data by implementing some algorithms. But we have used only one algorithm implemented in order to get the perfect accuracy score in our model. The steps of finding the accuracy score are described below in details.

CSV File

After matching the templates of the license plates with the modules of the license plate images, we made an Excel file which was converted into a CSV file later. The CSV file consists of such columns which defines the information of the city name, metro, letter, two hyphens and six types of digits which may be same or different.

The visibility of the images are Boolean values which indicates either it is true or false which indicates that if the image of the vehicle license plate is successfully recognized or not. And the other information are in numerical values.

The CSV file contains 13 columns initially which consists the unique image number, city name, metro, letter, two hyphens and six types of same or different number letters and the visibility of the images.

Training and Testing Data

A machine learning model can be trained with a specific dataset in order to transform to a specific prediction model by training. It is necessary to retain the model frequently. We have to train 70% data of the dataset and the other 30% data kept for the testing issue.

In order to find the proper accuracy of the dataset, we need to train and test the dataset. Now we have to describe how we completed the training and testing issue of the dataset.

Previously, we imported the dataset in the python file with pandas module. We used spyder IDE for the data splitting. We need to use some machine learning algorithms such as Random Forest

Classifier and Logistic Regression Algorithm of the python scikit learn module, which are perfect for supervised learning. We imported the 'train_test_split' function provided by scikit learn for the training. Then we defined the fixed test size which is 0.3. We used Binary Classification method for training the dataset in order to find the proper accuracy score.

We fixed the first seven columns of the dataset as the featured column and the last column (visible) as a predicted column which says either the image of the vehicle license plate is successfully recognized or not.

Last of all we checked the training and testing ratio whether it is 70%-30% or not. We found that 70% data was successfully trained and 30% data is tested.

Why only 70% data is trained?

We haven't train more than or less 70% data, trained only 70% data because 70% is the ideal ratio for training the dataset. We can verify by the testing data whether our model is predictive or not. We have some data which is not provided in the training. If the model predicts something around the training score, it will prove that we have trained the data successfully.

Result

We trained and tested data in the 70%-30% ratio. Then we fitted the instance using the Logistic

Regression algorithm. The instance was fitted with the featured columns and a predicted column (visible). After fitting the instance, the result gave us an accuracy score which is less than 1. Then we multiplied the score with 100 in order to convert the accuracy into percentage.

The result shows 58.3334% accuracy score after 70% training and 30% testing the data by applying the Logistic Regression Algorithm.

We know that 58.3334% accuracy is not a good result. It is not enough for us to complete the task successfully. So, we need to increase the accuracy of this dataset. To improve the accuracy, we have to implement some machine learning algorithms. This process is called 'Improving Performance with Ensembles'.

To improve the performance of the model, we need to implement Bagged Decision Trees Algorithm, Random Forest Algorithm, Extra Trees Algorithm, AdaBoost Algorithm, Stochastic Gradient Boosting Algorithm and Voting Ensemble Algorithm. All of these algorithms can classify the model in different ways and they can show different results to improve the accuracy score.

But we think it is necessary to implement the Random Forest Algorithm with extra importance in order the find the accuracy score. The result shows 73.334% accuracy score after 70% training and 30% testing the data by applying the Random Forest Algorithm. It is the best accuracy found by applying all the models.

The comparisons of the Random Forest model with the other models are described below.

- Random forests are among the most popular machine learning methods thanks to their relatively good accuracy, robustness and ease of use. It provides two straightforward methods for feature selection: mean decrease impurity and mean decrease accuracy.
- The Random Forest model is better than Decision Trees model. Because this model combines hundreds or thousands of decision trees, trains each one on a slightly different set of the observations, splitting nodes in each tree considering a limited number of the features. The final predictions of the random forest are made by averaging the predictions of each individual tree. When we don't bother much about interpreting the model but want better accuracy. Random forest will reduce variance part of error rather than bias part, so on a given training data set decision tree may be more accurate than a random forest.
- When our independent features are categorical, random forest tends to perform better than logistic regression. With continuous features, logistic regression is usually better. It all depends on the specifics off the problem being solved.
- Random forest models overfit a sample of the training data and then reduces the overfitting by simple averaging the predictors, which is totally useless for new data. Again, gradient boosting uses regression trees for prediction purpose where a random forest use decision tree. The random forest is easy to parallelize but boosted trees are hard to do. In this case, we can apply Gradient Boosting model, which performs better than Random Forest model.
- When all the variables are relevant, both Random Forest and Extra Trees methods seem to achieve the same performance. Since Extra Trees model seem to keep a higher performance in presence of noisy features, Extra Trees are better than Random Forest model.
- Voting Ensemble model performed better than our individual k-NN, random forest and logistic regression models in order to improve the accuracy score.

So, we can improve the accuracy score by using different models according to their advantages of classification method.

Result Discussion

After finding the accuracy score, we have to discuss the result whether it can be improved or not. That’s why, we applied some of the techniques which analyzes the result well. We tried to get the best feature selection of our dataset as well as we tried to reduce the features of our model to improve the accuracy score.

We applied the sequential feature selection such as forward selection and backward elimination as well as Principle Component Analysis in order to improve the accuracy score. The analysis of the result is described below in detail.

Sequential Feature Selection

First of all, we applied forward selection and backward elimination technique which shows the feature combinations sequentially according to the best accuracy score of the combinations of the features. In order to get the best case of the feature combination, we need to apply the forward selection and backward elimination technique.

Feature Selection is a very critical component in our model. Because our model presents data with high dimensionality. And when it presents data with very high dimensionality, models usually choke. The reasons of choking are given below.

- Training time increases exponentially with number of features.
- Models have increasing risk of overfitting with increasing number of features.

In order to apply forward selection and backward elimination technique, we have to implement sequential feature selection algorithm which makes a proper way to use forward selection and backward elimination technique.

It helps with these problems by reducing the dimensions without much loss of the total information. It also helps to make sense of the features and its importance.

Sequential feature selection algorithms are a family of greedy search algorithms that are used to reduce an initial d-dimensional feature space to a k-dimensional feature subspace where $k < d$.

The motivation behind sequential feature selection algorithms is to automatically select a subset of features that is most relevant to the problem. The goal of feature selection is two-fold which are,

- Improving the computational efficiency
- Reducing the generalization error of the model by removing irrelevant features or noise.

Above all, Sequential Feature Selection Algorithms remove or add one feature at the time based on the classifier performance until a feature subset of the desired size k is reached.

There are four types of Sequential Feature Selection Algorithms. They are,

- Forward Selection
- Backward Elimination
- Forward Floating Selection
- Backward Floating Elimination

But we need only forward selection and backward elimination for our model. Because, our main target is to get the best feature combination according to the accuracy.

In the sequential selection algorithm, Random Forest Algorithm is also applied in order to identify the accuracy score. Because, identifying the accuracy score is also necessary to get the best feature combination.

Forward Selection

Forward selection is a type of stepwise regression which begins with an empty model and adds in features one by one. In each forward step, we can add the one feature that gives the single best improvement of accuracy to our model.

In other words, no independent feature is entered into the equation first and each one is added one at a time if they contribute to the accuracy improvement.

After applying forward selection to our model, we got the combinations of features according to the accuracy improvement which is represented in the following graph.

Figure: Forward Selection Technique

Backward Elimination

Backward elimination is the reverse process of forward selection technique. It begins with a full set of features and deletes in features one by one. In each backward step, we can delete the one feature that gives the single best improvement of accuracy to our model.

In other words, all the independent features are entered into the equation first and each one is deleted one at a time if they do not contribute to the accuracy improvement.

After applying backward elimination to our model, we got the combinations of features according to the accuracy improvement which is represented in the following graph.

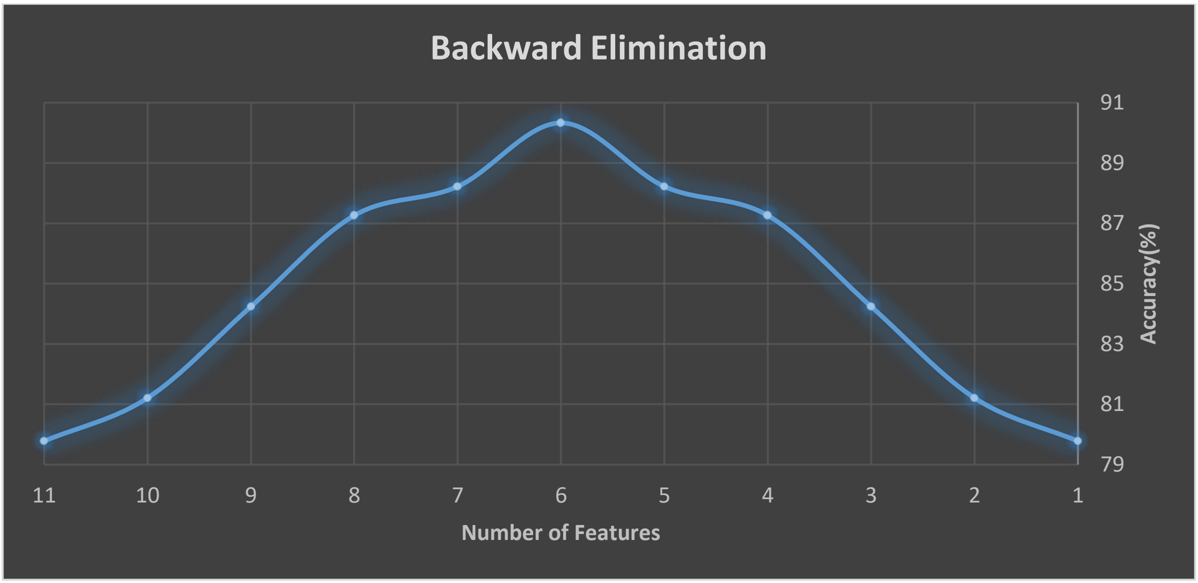


Figure: Backward Elimination Technique

Analysis of Forward Selection and Backward Elimination

There are some criterion found in forward selection and backward elimination techniques. They are given below.

- In both forward selection and backward elimination techniques, it shows that when the number of features is 6, it provides highest accuracy score, that's why the best feature combination is 6 for both forward selection and backward elimination techniques.
- The accuracy is same when the number of feature is one (first feature) for the forward selection and when the number of features is nine (last feature) for the backward elimination.
- The accuracy is same when the number of feature is nine (last feature) for the forward selection and when the number of features is one (first feature) for the backward elimination,
- The forward selection technique is the reverse work of the backward elimination technique.

That's all about the forward selection and backward elimination techniques.

Principle Component Analysis (PCA)

Principal Component Analysis (PCA) is a dimension-reduction tool that can be used to reduce a large set of features to a small set that still contains most of the information in the large set. In other words, it is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of features into a smaller one that still contains most of the information in the large set.

It is a statistical procedure and linear dimensionality reduction algorithm to find a more meaningful basis or coordinate system for our data and works based on covariance matrix to find the strongest features.

It is used when we need to handle the curse of dimensionality among data with linear relationships, i.e. where having too many dimensions or features in your data causes noise and difficulties as well as increases features space. So, this approach is used to reduce the dimensionality of the model in order to avoid the curse of dimensionality.

We also applied principle component analysis to reduce the dimensionality of the model in order to avoid the curse of dimensionality. Because there are eleven features in our model. The increased number of features increases feature space, which causes the curse of dimensionality. That's why, we need to reduce the number of features by eliminating the less important features. And so, we used it for dimensionality reduction.

The eigenvectors and eigenvalues of a covariance or correlation matrix represent the "core" of a Principle Component Analysis. The eigenvectors which are known as principal components, determine the directions of the new feature space, and the eigenvalues determine their magnitude.

In case of Principle Component Analysis, variance means summative variance or multivariate variability or overall variability or total variability. Below is the covariance matrix of some three variables. Their variances are on the diagonal and the sum of the three values is the overall variability.

It is necessary to normalize data before performing Principle Component Analysis. The Principle Component Analysis calculates a new projection of our model. If we normalize our data, all the features have the same standard deviation, thus all features have the same weight and our Principle Component Analysis calculates relevant axis.

Since the sample per class is less in our model, Principle Component Analysis performs well in order to reduce the dimensionality.

Principle Component Analysis finds a vector that "best represents" our data set in a much lower dimension. To get better accuracy, we need to find a vector that "best discriminates" between your classes. In this case, Linear Discriminant Analysis (LDA) works better than Principle Component Analysis.

After applying Principle Component Analysis to our model, we got the important features according to their "Explained Variance Ratio" which is represented in the following graph.

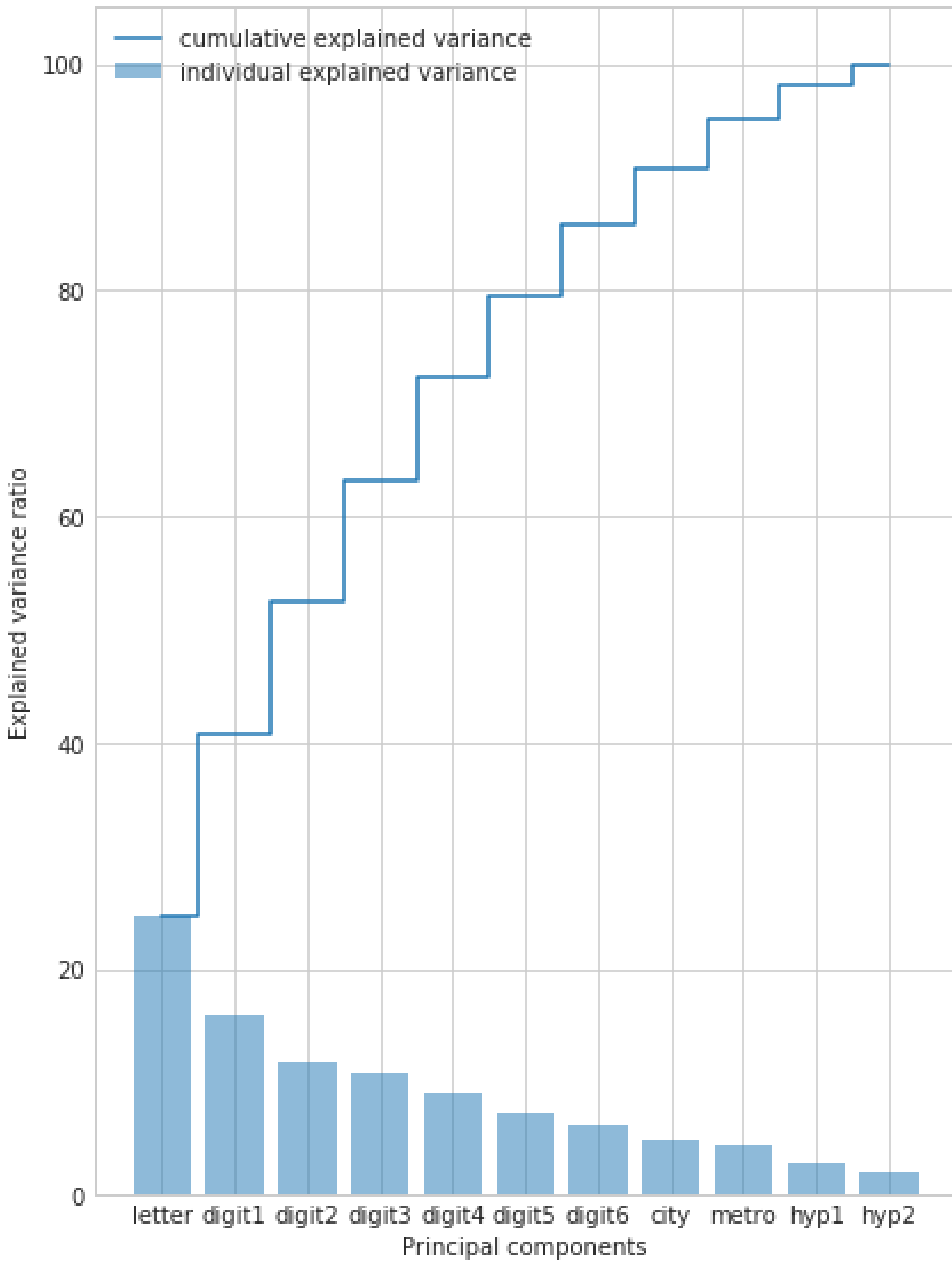


Figure: Principle Component Analysis

Here, we have eleven features in which hyphen1 and hyphen2 are less important according to their explained variance ratio. Without these two features, the explained variance ratio is above 95%. So, we can eliminate these two hyphens in order to reduce dimensionality.

Limitations

Though we have completed most of the tasks, we have faced some problems to perform some activities at the time of completing the tasks. So, there are some limitations in our activities. The limitations are given below.

- Character Segmentation is a very important process to segment the image of a license plate of a Bangladeshi vehicle. But we couldn’t perform the character segmentation perfectly because of “Matra Elimination” process.
- Though we have performed Principal Component Analysis (PCA) to reduce the dimensionality to the features, we perform least discriminant analysis to reduce the dimensionality of the features.
- Some vehicles are too old and the license plates of these vehicles are blurry. So, it is too tough to recognize the license plates of these vehicles from the captured images. We eliminated the images from our dataset later.
- The images are captured in the different modes of daylight. So, some images cannot be recognized which were captured without the daylight.

We couldn’t get the proper solutions of these problems. So, we need to make a proper solution to overcome these problems.

Future Plan

We have recognized the license plates by completing license plate detection, segmentation and template matching. Again, we identified the accuracy score and discussed the result briefly by Feature Selection (forward selection and backward elimination) and Principal Component Analysis (PCA). Our future plan is to implement limitations of our activities which are described below.

- We couldn’t perform the character segmentation perfectly because of Matra Elimination process. So, we will improve it in future.
- We couldn’t perform least discriminant analysis to reduce the dimensionality of the features. So, we’ll try to apply it in future.

So, we have to complete these works in our further update.

Conclusion

We have done the tasks of our thesis activities so far, Automatic License Plate Recognition using

Template Matching Technique as possible as we can. We have completed successfully the tasks of license plate detection, license plate segmentation and template matching as well as finding the accuracy score. Moreover, we have discussed in detail improvement of the accuracy score as far as possible so that it will help us. Though there are some limitations in our activities, we have tried our best to complete the tasks successfully. We will improve our work by making a proper solution of the limitations and completing more tasks in our further update.

References

1. Bangladeshi Vehicle Digital License Plate Recognition for Metropolitan Cities Using Support Vector Machine
2. Bangla Digital Number Plate Recognition using Template Matching for Higher Accuracy and Less Time Complexity
3. Forward Selection Component Analysis: Algorithms and Applications
4. Real Time Bangla Vehicle Plate Recognition towards the Need of Efficient Model – A Comprehensive Study
5. Bangla Automatic Number Plate Recognition System using Artificial Neural Network
6. Automatic License Plate Recognition (ALPR) for Bangladeshi Vehicles
7. A Tutorial on Principal Component Analysis

Papers we have studied

The summarizations of the above papers are given in the “summary.pdf”.