UNDERGRADUATE THESIS PROPOSAL

License Plate Recognition Using CNN for Outdoor Area

Muhammad Bagas Dewantoro

15/380918/PA/16726

Computer Science

Department of Computer Science and Electronic

Faculty of Mathematics and Natural Sciences

Universitas Gadjah Mada

Yogyakarta

2018

APPROVAL PAGE

UNDERGRADUATE THESIS PROPOSAL

License Plate Recognition Using CNN for Outdoor Area

Proposed by

MUHAMMAD BAGAS DEWANTORO

15/380918/PA/16726

Has been defended in front of the examiner team

On December …, 2018

The composition of Examiner Team

Drs. Agus Harjoko, M.Sc., Ph.D
Supervisor

Moh. Edi Wibowo, S.Kom., M.Kom., Ph.D
Head of Examiners

Azhari S.N, Drs., MT., Dr
Examiner

TABLE OF CONTENT

TABLE OF CONTENT i

TABLE OF TABLES iii

TABLE OF FIGURES iv

CHAPTER I INTRODUCTION 1

1.1 Research Background 1

1.2 Research Problem 2

1.3 Research Scope 2

1.4 Research Objective 2

1.5 Research Advantage 2

CHAPTER II LITERATURE REVIEW 3

CHAPTER III BASIC THEORY 7

3.1 Histogram Equalization 7

3.2 Gaussian Filter 7

3.3 Otsu's Method 8

3.4 Canny Edge Detection 10

3.5 Artificial Neural Network 11

3.6 Activation Function 12

3.6.1 Rectified Linear Unit 12

3.7. Stochastic Gradient Decent 13

3.8 Convolution Neural Network 14

3.8.1 Convolution Layer 14

3.8.2 Pooling 15

CHAPTER IV RESEARCH METHODOLOGY 16

4.1 Research Description 16

4.2 Propose Method 17

4.2.1 Data Acquisition 17

4.2.2 License Plate Detection 17

4.3.3 License Plate Recognition 18

4.3 Evaluation 20

4.3.1 Average Precision 21

4.3.2 Mean Average Precision (MAP) 22

4.4 Implementation 22

CHAPTER V RESEARCH SCHEDULE 23

CHAPTER VI REFERENCE 24

TABLE OF TABLES

Table 2.1 Comparison with Previous Works 5

Table 5.1 Research Timeline 23

TABLE OF FIGURES

Figure 3.1 A simple 3 input units, 2 hidden layers with 4 hidden units, and 1 output units (I Wayan Suartika E. P et al., 2016) 11

Figure 3.2 A Graph representation of the ReLu Activation Function (Agarap, 2018). 13

Figure 3.3 Full Model of The CNN Architecture (A. et al., 2013) 14

Figure 3.4 The explanation of how convolutional works (Albawi et al., 2017) 15

Figure 3.5 Result of computing the output values of a max pooling layer (I Wayan Suartika E. P et al., 2016) 15

Figure 4.1 Samples of the Image from the dataset 16

Figure 4.2 Samples of the cropped Image use for further process 16

Figure 4.3 The data example that separate based region to be recognize by CNN 18

Figure 4.4 Diagram of the Experiment 20

CHAPTER I INTRODUCTION

1.1 Research Background

Development of a smart city can be said to have come from the improvements of technology that contributes in the quality of citizen welfare. Such technology that can be said to create a more efficient smart city is the License Plate Recognition (LPR). Xiang et al., (2018) mention that LPR can be implemented in many practical applications such as automatic toll collection, traffic law enforcement, and private spaces access control, and road traffic monitoring, and with the development of Intelligence Transportation Systems (ITS), the detection of license plate have become an important issue.

With the development of Intelligence Transportation Systems (ITS), an LPR will greatly improve the security system of the transportation in the parking area, being able to detect the license plate can be used for security and safety purposes. An LPR can also be used in parking systems for obtaining the license plate for the tickets. The main goal of this research is to create a system that can reads an image of a vehicle and recognize the license plate of the vehicle in outdoor parking area. This system utilizes image processing and machine vision to obtain the license plate of the vehicle.

However there are numerous problems in creating an LPR that researchers have been trying to solve. The major problem is to create a detector that can be used in outdoor area with high accuracy value. Creating a system that can solve this problem may prove to be difficult, there are already multiple system that can be used in real time and high accuracy, but when it implement in outdoor area the accuracy decreased. Another problems would be the lighting within the image, this may cause the detector to get false positive results, since lighting can be affected by the environment easily such as bad weather and night time, or by the system itself such as bad camera in low light. All these problems may cause interruptions in the LPR accuracy (Xiaohong Long and Jing Zhou, 2013).

With the newly development of machine learning, it have greatly improve traditional LPR. With machine learning have also pushed the development of deep learning and as well as Convolution Neural Network (CNN). CNN is a class of deep learning algorithm that can be used in machine vision, and in this research it will be used to create a system that takes the image of the vehicle and outputs its license plate. CNN have also shown to be robust against bad lighting, given sufficient training data for the CNN, it can detect objects in a noisy image to a small degree, for as long as the license plate is visible, unlike traditional methods where the image is required to be very clear and in a certain position for it to get an accurate reading.

1.2 Research Problem

This research discuss about a better way to recognize a license plate for outdoor area where brightness and noises become the main problem for the researcher to recognize the license plate. Because of these problems, a system with high accuracy and high rate of mean average precision in any condition is needed.

1.3 Research Scope

This implementation will used the FMIPA UGM CCTV camera located in the FMIPA UGM security post entrance to obtain the video footage, which therefore will be train using the data from the specified location of the FMIPA UGM CCTV, this is so that to maintain consistencies. Due to the location of the implementation the camera used will be in a fixed position.

1.4 Research Objective

The objective of this implementation is to design an effective method for license plate recognition in outdoor area using the Canny edge detection and CNN that would have more than 60% accuracy.

1.5 Research Advantage

To expand the capabilities of the Canny edge detection and CNN to be able to recognize characters in outdoor area and be used for new researches.

CHAPTER II LITERATURE REVIEW

In most case regarding the ALPR, most researchers will use an object detection algorithm and then, uses a character recognition algorithm to recognize the license plate. Massoud et al., (2013) proposed combination of methods to increase the accuracy of the license plate recognition. The edge detection uses Sobel edge detector, Morphological algorithm, dilation and erosion method and the Filling Holes algorithm was used to fill rectangles that result from dilation process. Then 2-D Median Filter with 5x5 mask used for filtering and smoothing. For the last step, the Statistical Correlation method was used for the character recognition. The final results which tested by 100 patterns under several condition, shows the performance and accuracy were 91% success.

Keong and Iranmanesh, (2016) proposed Pearson Correlation for automatic number plate recognition system. Some methods like Canny method, Otsu's method and Median Filter were used for pre-processing stage. The spatial resolution of the input image was reduced from 3264 x 2448 to 1000 x 750 which decrease the computing time from 6.07 seconds to 1.24 seconds. Bounding Box technique used in character segmentation stage. Bounding Box technique is occurred based on measuring connected components that is relied on the neighbor's pixel intensity. The character segmentation achieved 99.6% accuracy and character recognition achieved 91.5% accuracy. The overall accuracy achieved 91.1%

Tabrizi and Cavus, (2016) proposed the combination of K-NN and SVM to decrease the training phase time and increase the efficiency of LPR system. K-NN was used as the initial step to classify all data set without training and then multiple class SVMs was performed on only smaller data set with similar characters. While in the segmentation stage, Prewitt method and Dilate image function were used for edge detection. The experiment take 257 images of car license plate taken randomly by normal digital camera with 1024 x 768 pixel frame size in RGB and JPG format in the range of 5 to 10 m. The final result shows that the KNN-SVM model improves the character recognition rate significantly from 94% to 97% for all cases tested in the experiment.

Wang et al., (2015) proposed SIFT feature for the Chinese character license plate recognition which consist 4 major stages: scale-space extrema detection, feature point localization, orientation assignment and feature point descriptor. From 800 license plate candidate images, 700 images are real license plates and 100 images are noise areas. The Chinese character region extracted range from 20 x 39 to 112 x 200 and each template is fixed to 55 x 100. Experimental result show the recognition success rate is 96.0% with average executive time 258 ms.

Ktata et al., (2015) focus on time-frequency analysis based on auto-correlation feature and neural network recognition. In plate extraction stage, low-pass filters are used to smooth the image and remove the high frequency components related to the noise and dynamic threshold to filter out horizontal and vertical histograms values. The method is evaluated on a PC 2x2.4 GHz CPU, 8 GB RAM and Windows 8 operating system and implemented using Matlab. The experiment uses 2 databases consist of 740 images for the detection and 740 images for the recognition. The result shows 88.67% accuracy for the detection and 90.78% accuracy for the recognition with 1 hidden layer and 10 neurons.

Surekha et al., (2018) proposed Feed-forward backpropagation neural network for plate recognition and Otsu's method for image thresholding. The implementation of the system was achieved using Matlab and Raspberry PI sensor and camera to take the image. There are 2 types of processing takes place for license plate localization efficiency, morphological processing with 105 total samples acquire 83% efficiency at the rate of 0.7 seconds and horizontal and vertical edge projection with 105 total samples acquire 100% efficiency at the rate of 0.15 seconds. The result for neural network prediction efficiency acquires 97% efficiency from 105 total samples at the rate of 1.3 seconds.

The previous works show many different ways to recognize the license plate. Some research use tools such as infrared for sensor detector for the image acquisition. The previous works also show that the license plate recognition will be more efficient using image processing and neural network. The summary of the comparison from the previous work can be seen in Table II.1

Table 2.1 Comparison with Previous Works

No	Research Topic	Method	Result
1	Automated new license plate recognition in Egypt (Massoud et al., 2013)	-Uses Sobel edge detector for edge detection -Uses Dilation operation for separates the characters. -Uses Filling Holes algorithm -Uses 2-D Median Filter for filtering and smoothing -Uses Statistical Correlation for matching technique	From 100 patterns, 91% success
2	Malaysian Automatic Number Plate Recognition System using Pearson Correlation (Keong and Iranmanesh, 2016)	-Uses Otsu's method to convert grayscale image to binary image. -Uses Median filter to smooth the image -Uses Canny method for edge detection	Overall accuracy 91.1%
3	Automatic License Plate Recognition Using Image Processing and Neural Network (Surekha et al., 2018)	-Uses Feed-Forward Backpropagation Neural Network -Uses Otsu's method for thresholding	The character recognition rate 94% to 97%

Table 2.1 Comparison with Previous Works

No	Research Topic	Method	Result
4	Tunisian License Plate Number Recognition (Ktata et al., 2015)	-Uses Dynamic Threshold for plate extraction and Low-pass filter to smooth the images -Uses Neural Network for recognition	The recognition success rate is 96.0% with average executive time 258 ms.
5	License Plate Recognition Based on SIFT Feature (Wang et al., 2015)	- Uses SIFT feature matching	From 740 images for the detection and 740 images for the recognition, 88.67% accuracy for the detection and 90.78% accuracy for the recognition
6	A Hybrid KKN-SVM Model for Iranian License Plate Recognition (Tabrizi and Cavus, 2016)	-Uses K-Nearest Neighbors and Support Vector -Uses Prewitt for edge detection	Acquires 97% efficiency from 105 total samples at the rate of 1.3 seconds

CHAPTER III BASIC THEORY

3.1 Histogram Equalization

Histogram Equalization (HE) is a gray level transformation method that forces the transformed gray level to spread over the entire intensity range. The HE adjusted image intensity to enhance contrast. The dynamic range and contrast of an image are modified by altering the image. The modification is achieved by using cumulative distribution function as the mapping function. The intensity levels are changed such that the peaks of the histogram are stretched and the troughs are compressed.

The histogram of a digital image with gray values r0, r1…rL-1 is the discrete function written in equation (3.1), where n_k denotes the number of pixels with gray value r_k, n denotes the number of pixels in the given image. p(r_k) represents the fraction of the total number of pixels with gray value r_k.. The global appearance of the image can be represented by the histogram (Reddy et al., 2017).

$$p(r_k) = \frac{n_k}{n} \tag{3.1}$$

HE offers the advantage of full automation, since HE automatically determines a transformation function to produce a new image with a uniform histogram (Reddy et al., 2017). Equation (3.2) is to determine the new pixel value of the HE.

$$S_k = T(r_k) = \sum_{j=0}^{k} \frac{n_j}{n} = \sum_{j=0}^{k} P_r(r_j) \tag{3.2}$$

3.2 Gaussian Filter

The Gaussian Filter is included in a linear type filter with a weight value for each pixel set in it by using the Gaussian function. The Gaussian Filter is used to blur images and remove unwanted detail and noise. The linear process in the Gaussian filter is done by multiplying each adjacent neighbor pixel and summing the result so that it gets the result for a certain coordinate point. The mechanism of the linear spatial filter is to move the center of a filter mask from 1 point to another. In each pixel, the result of the filter at that point is the sum of the multiplication of the filter coefficients and the corresponding neighbor pixels in the filter mask range.

There are 2 components to note on the Gaussian filter that is correlation and convolution. Correlation is the process of passing mask to the image. While the definition is defined as a process for obtaining pixel values based on their own pixel values, neighboring pixels and kernel matrices. In the process, the kernel will be shifted along the rows and columns of the input image used so that the new pixel value of the resulting image will be obtained. In the Gaussian filter itself, the convolution process first rotates the filter mask of 180º and then passes the image (Putra et al., 2017).

Gaussian filter has 2 types of filters: 1-dimensional Gaussian filter and 2-dimensional Gaussian filter, where the equation (3.3) as the 1-dimensional Gaussian function and the equation (3.4) as the 2-dimensional Gaussian function. The standard deviation of the distribution is expressed as σ and x and y in equation (3.4) is expressed as coordinate points (rows and columns) in image pixels. As the value of σ gets larger, then the distribution curve of the Gaussian gets wider and the peak decreases.

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{(\frac{-x^2}{2\sigma^2})} \tag{3.3}$$

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{(\frac{-x^2-y^2}{2\sigma^2})} \tag{3.4}$$

3.3 Otsu’s Method

Otsu's 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 falls in foreground or background. The aim is to find the threshold value where the sum of foreground and background spreads is at its minimum. Otsu’s method, works based on the very simple idea that minimizes the weighted within-class variance which can be seen in equation (3,5). Otsu algorithm will consider some assumptions such as the input image is bimodal, uniform illumination etc.

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t) \tag{3.5}$$

The class probabilities q1 and q2 are estimated as equation (3,6) and equation (3,7) (Reddy et al., 2017).

$$q_1(t) = \sum_{i=1}^{t} P(i) \tag{3.6}$$

$$q_2(t) = \sum_{i=t+1}^{I} P(i) \tag{3.7}$$

The class means are computed as equation (3,8) and equation (3,9) (Reddy et al., 2017).

$$\mu_1(t) = \sum_{i=1}^{t} \frac{iP(i)}{q_1(t)} \tag{3.8}$$

$$\mu_2(t) = \sum_{i=t+1}^{I} \frac{iP(i)}{q_2(t)} \tag{3.9}$$

The individual class variances are computed as equation (3,10) and equation (3,11) (Reddy et al., 2017).

$$\sigma_1^2(t) = \sum_{i=1}^{t} [i - \mu_1(t)]^2 \frac{P(i)}{q_1(t)} \tag{3.10}$$

$$\sigma_2^2(t) = \sum_{i=t+1}^{I} [i - \mu_2(t)]^2 \frac{P(i)}{q_2(t)} \tag{3.11}$$

P represents the image histogram. The problem of minimizing within class variance can be expressed as a maximizing the between class variance. It can be written as a difference of total variance and within class variance as shown in equation (3.12). Where $\sigma_w^2(t)$ is the is the within-class variance and $q_1(t)[1 - q_1(t)][\mu_1(t) - \mu_2(t)]^2$ is the between-class variance (Reddy et al., 2017).

$$\sigma^2 = \sigma_w^2(t) + q_1(t)[1 - q_1(t)][\mu_1(t) - \mu_2(t)]^2 \tag{3.12}$$

3.4 Canny Edge Detection

The Canny edge detector is an edge detection operator that uses a multi-stage algorithm to detect a widerange of edges in images. Canny algorithm smoothest the image by Gaussian filter, calculates the magnitude and direction of gray level gradient, which has the non-maxima suppression on gradient magnitude, and detects and connects the edge from the candidate points by the high and low thresholds. Canny algorithm finds the image gradient to highlight regions with high spatial derivative.

The process of Canny edge detector algorithm is as follows (Feng et al., 2017):

1. **Smoothing**: Blurring of the image to remove noise by convolving the image with the Gaussian filter.
2. **Find the intensity gradients of the image**: The edges should be marked where the gradients of the image has large magnitudes, finding the gradient of the image by feeding the smoothing image through a convolution operation with the derivative of the Gaussian in both the vertical and horizontal directions. The intensity gradients is determine using equation (3.13) and equation (3.14) (Borker, 2015).

$$M(m,n) = \sqrt{(g_m^2(m,n) + g_n^2(m,n))} \tag{3.13}$$

$$\theta(m,n) = \tan^{-1}[g_n(m,n)/g_m(m,n) \tag{3.14}$$

3. **Non-Maximum Suppression**: Only local maxims should be marked as edges. Finds the local maxima in the direction of the gradient, and suppresses all others, minimizing false edges.
4. **Double Thresholding**: potential edges are determined by thresholding, instead of using a single static threshold value for the entire image, the Canny algorithm introduced hysteresis thresholding, which has some adaptively to the local content of the image. There are 2 threshold levels, high and low values, where (high value > low value) the pixel values above the high value are immediately classified as edges.
5. **Edge Tracking by Hysteresis**: Final edges are determined by suppressing all edges that are not connected to a very strong edge.

3.5 Artificial Neural Network

Artificial Neural Network (ANN) is a computational model that was inspired by the neurological connection within a biological brain. The model consists of simple connections called neurons. The neuron send signals at spikes of electrical activity through a long thin stand known as an axon and an axon splits this signals through synapse and send it to the other neurons. A neural network can be trained to perform a particular function by adjusting the values of the connections (weights) between elements (Tur, 2012).

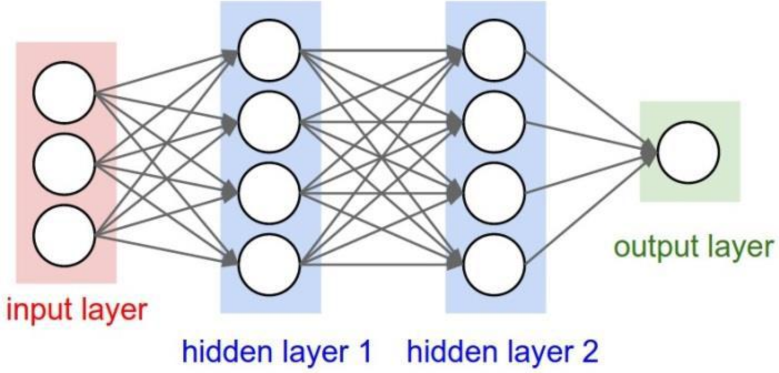

Figure 3.1 A simple 3 input units, 2 hidden layers with 4 hidden units, and 1 output units (I Wayan Suartika E. P et al., 2016)

Figure III.1 represents a simple ANN with 4 layers, an input layer, 2 hidden layers, and an output layer. The circles denote each neuron, and each connection (the lines connecting each neuron) is called synapses. The 3 neurons in the input layer represent the number of inputs into the ANN and neuron for the output layer represents the number of output from the system.

A neural network takes a vector as input, and it will be multiplied by an initially random assign weights within the synapses which will become the input signal for the next neuron layer until the output layer. If the output result does not match the expected value, then error value is calculated and weights on the synapses are renewed and optimized to minimalize the error of the output value in each iteration.

3.6 Activation Function

An activation function is a function within the neurons that takes its inputs and defines its output. There are multiple activation functions such as the sigmoid function, TanH, and rectified linear unit (ReLu) etc. Activation functions are used to convert the input signal into an output signal, and converting it from a linear.

3.6.1 Rectified Linear Unit

Rectified Linear Unit (ReLu) is used in CNN for activation function. ReLu works by thresholding values at 0, i.e. f(x) = max (0, x). Simply put, it outputs 0 when x < 0, and conversely, it outputs a linear function when x ≥ 0. Figure 3.2 is the visual representation. The predicted class for ReLu classifier is determine in equation (3.15) (Agarap, 2018).

$$y = \arg\max\ \max(0, x) \tag{3.15}$$

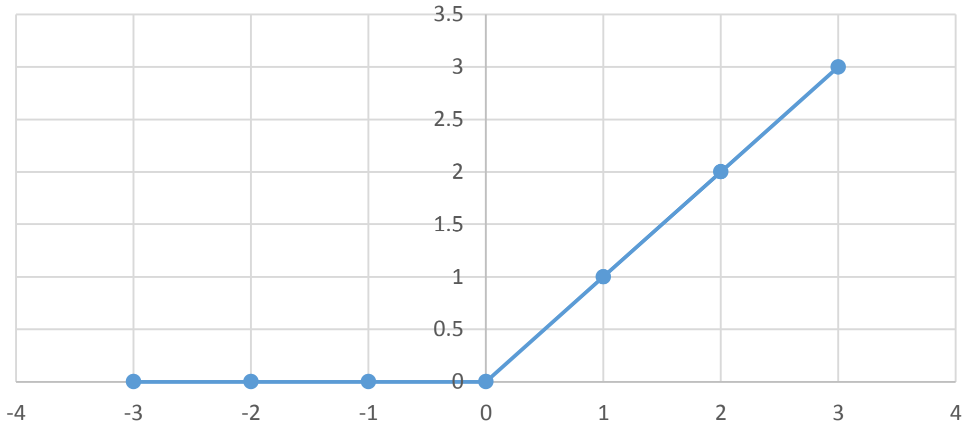

Figure 3.2 A Graph representation of the ReLu Activation Function (Agarap, 2018).

3.7. Stochastic Gradient Decent

Gradient Decent is an optimization algorithm that can be used to minimalize a function, in a Neural Network this function is the cost function of the model. This cost function represents the error of the neural network model and gradient decent performs the optimization by applying small changes to the weights and gradually minimalizing the cost function.

The Gradient Decent minimalize by obtaining the gradient of the cost function with respect to the current weights, and by going down the gradient, its negative value, the minimum point can be approach. However there is a learning rate to be considered, since if the learning rate is too high, this means that the steps taken to reach the minimum is bigger, so there is a chance that it can be missed, but if the learning rate is too small, then reaching the minimum will be too long and the minimum will never be reached.

The gradient (the rate of change) of the cost function can be obtain by deriving the cost function, and renewed the weights of the neural network with every iteration, however calculating the derivative for every data can be computationally expensive, and therefore the Stochastic Gradient Decent (SGD) is used. Unlike traditional gradient decent where every data is used to renew the weights, SGD updates the weights in batches of data. The sum of the gradients for a batch of data is used to update the weights (Hansson and Olsson, 2017).

3.8 Convolution Neural Network

Convolution Neural Network also known as CNN is a form of artificial neural network. It consists of an input layer which takes in an image of different dimensional matrix as the input, a hidden layer which consist of 3 type of layer convolution layer, the pooling layer, the rectified linear unit (ReLu) layer, and lastly the classifying layer which is an optional layer as the hidden layer, and the last type of layer the output layer which produces the result of the CNN (Johnson and Karpathy, 2015).Unlike Artificial Neural Network that uses weights in the synapses, in CNN it uses filters applied on the input matrix. Filter is a matrix that contains adjustable values that are used to extract the features on the matrix into the next layer.

A CNN can have different architecture design which consists of different order of the different type of layers within the hidden layer, different filter sizes and properties of each layer. By having different layers in the hidden layer, it can affect the unique results obtain from the CNN. Figure 3.3 shows a full model of the CNN architecture.

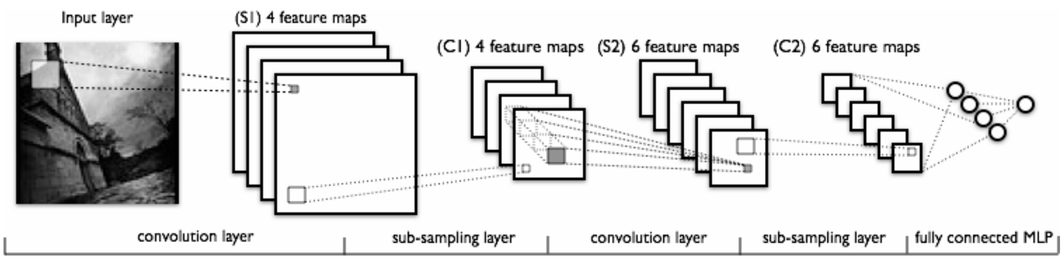

Figure 3.3 Full Model of The CNN Architecture (A. et al., 2013)

3.8.1 Convolution Layer

The aim of Convolutional layer is to learn feature representations of the inputs. Convolutional layer is consists of several feature maps. Each neuron of the same feature map is used to extract local characteristics of different positions in the former layer, but for single neurons, its extraction is local characteristics of same positions in former different feature map. In order to obtain a new feature, the input feature maps are first convolved with a learned kernel and then the results are passed into a nonlinear activation function. A different feature map can be obtained by applying different kernels (Guo et al., 2017).

Figure 3.4 displays a visual representation of the convolution layer. The 3x3 pixel window in green is the 3x3 kernel and the 7x7 pixel window in white is the object.

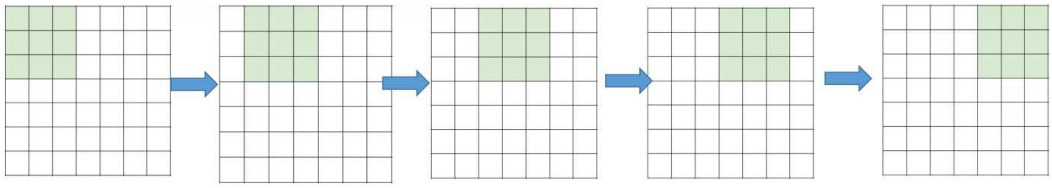

Figure 3.4 The explanation of how convolutional works (Albawi et al., 2017)

3.8.2 Pooling

The Pooling Layer will perform a down sampling operation along the spatial dimensions (width, height), resulting in a small volume of output data. These down sampling works by having a window of fixed size (usually 2 or 3 pixel) and stride the window across the each image in the stack and for each window then take the maximum or average value from within the window. This will shrink the image stack (Down sampling). This is perform to obtain the specific features within the input signal and this will greatly reduce the volume and computational time and controls the problem of overfitting (Aloysius and Geetha, 2017). Figure 3.5 is the process of max pooling layer.

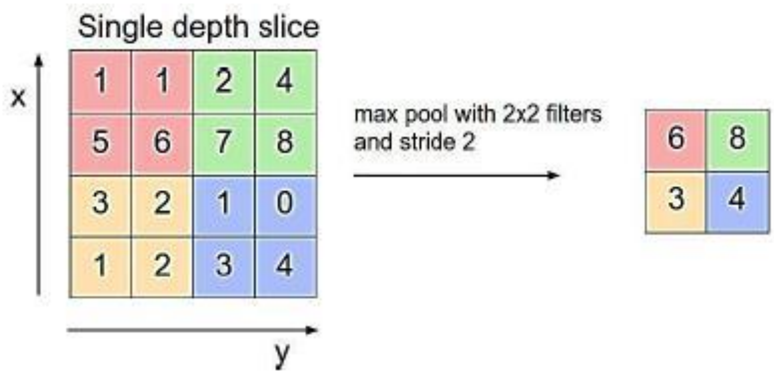

Figure 3.5 Result of computing the output values of a max pooling layer (I Wayan Suartika E. P et al., 2016)

CHAPTER IV RESEARCH METHODOLOGY

4.1 Research Description

Data acquisition is the first steps of image processing to collect the data which will be process. Data acquisition is done by capturing the video from FMIPA UGM CCTV. From figure 4.1 the images are obtained from a fixed camera position. The plate will be detected automatically using Sliding Window algorithm which then the plate will be cropped and the license plate detection process and the license plate recognition can be done. Figure 4.2 shows the image after being cropped.

Figure 4.1 Samples of the Image taken by FMIPA UGM CCTV

Figure 4.2 Samples of the cropped Image use for further process

4.2 Propose Method

4.2.1 Data Acquisition

Data Acquisition done by capturing the video from FMIPA UGM CCTV will obtain over1000 data that will be use to training, testing and validating the license plate detection process and the license plate recognition process. The camera can be access from software named iVMS-4200 which can be access only using UGM internet. The camera will capture the video on the position of 1 meter height. The output of data acquisition is raw images which then will be process in the license plate detection stage.

4.2.2 License Plate Detection

The images from data acquisition will be processed using the Sliding Window algorithm to detect the location of the license plate. The Sliding Window algorithm works by scanning every area using a boundary box until the license plate is detected. Before Sliding Window is applied, the data first must be annotated to make the algorithm know which object is the license plate and which object is not the license plate. The next step is to crop the image to separate the license plate and the other object using automatic cropping image from OpenCV library.

The cropped images then will process in the Enhancement process. The principal objective of enhancement is to process an image so that the result is more suitable than the original image for a specific application. The image first will be resized into 600x400, so all of the images will have the same size. The new size image then converted into grayscale to simplify the enhancement process because it only contains 2 main colors, black and white. The Histogram Equalization (HE) is applied to normalize the pixel values of the grayscale image. HE makes the bright image darker and the dark image brighter.

The image occasionally will have many noises which will disturb the Segmentation process. The deficiency of the segmentation process will be prevented by applying the Gaussian filters. The Gaussian filter that will be used is the Gaussian low-pass filter which will make the image becomes blurred reducing the detail of the image (Makandar and Halalli, 2015). The result of the Gaussian low-pass filter is Bokeh effect where only the vehicle's plate object becomes blurred.

The segmentation aims to classify the background pixels and the foreground pixel. The segmentation will first separate the dark pixel and the white pixel by applying Otsu's thresholding method after Gaussian low-pass filter is applied to make the Otsu's thresholding method become efficient. The separation is used to detect the plate which will have white pixel. The Otsu's method is applied to replace the double thresholding method from traditional Canny edge detection threshold method. Hatem, (2013) mention that the Otsu method is the best method of selecting an optimal threshold value automatically by the discriminant criterion.

The Otsu's thresholding method will make the image has 2 colors black and white, where the white color from the high pixel value object and the black color from the low pixel value object. The Canny will detect the edge of each character from the license plate which will be recognized. The Each character will be recognized by using Convolutional Neural Network (CNN) method for better results (Surekha et al., 2018).

4.2.3 License Plate Recognition

The Convolutional Neural Network (CNN) will be used in the recognition process to recognize each character on the vehicle's plate. The image after Segmentation process will be used as input. The plate will consist of 3 regions where the first region is the first 1 or 2 letters depend on what city the vehicle comes from. Then the number of the plate where also has a different amount of numbers and the last 2 letters at the end of the license plate. Figure 4.3 shows the example of the data that will be recognized by the CNN.

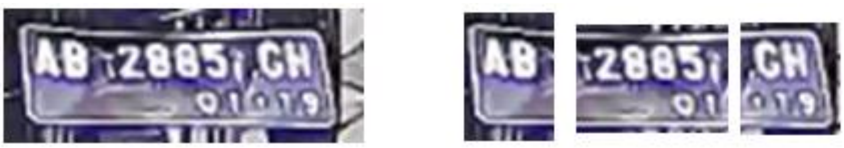

Figure 4.3 The data example that separate based region to be recognize by CNN.

The CNN architecture will use ReLu as activation function to transform the input into a positive value which it will activate the neurons or 0 value which it will not activate the neurons. The different architecture will be implemented to the CNN model for the evaluation to obtain the best result. The difference will be the amount of the hidden layer, the size of the kernel, the amount of the neurons and for the training, the amount of the epochs also will be changed to evaluate them and since there is many shapes of character uses in the license plate, the output will be also modified. The input from the segmentation process will be 300 images. The first CNN architecture will use 10 hidden layers with 37 output neurons consist of 26 letters of the alphabets, 10 digits and 1 object which is the license plate. Figure 4.2 shows the experiment diagram.

The data from data acquisition will be annotated first as the training data for the Sliding Window algorithm to detect the vehicle's plate. Training for the detection and the recognition process will use Stochastic Gradient Descent (SGD) with 100 epochs in the beginning. The initial weight will be initialized randomly, and the number of weight will depend on the architecture used. Since the CNN for the recognition process uses Backpropagation method, the weight will be updated automatically after the end of the previous process. The SGD will find the optimum way to classify the license plate character with help of the backpropagation process. The training process will be evaluated using validation data to make sure there is no overfitting.

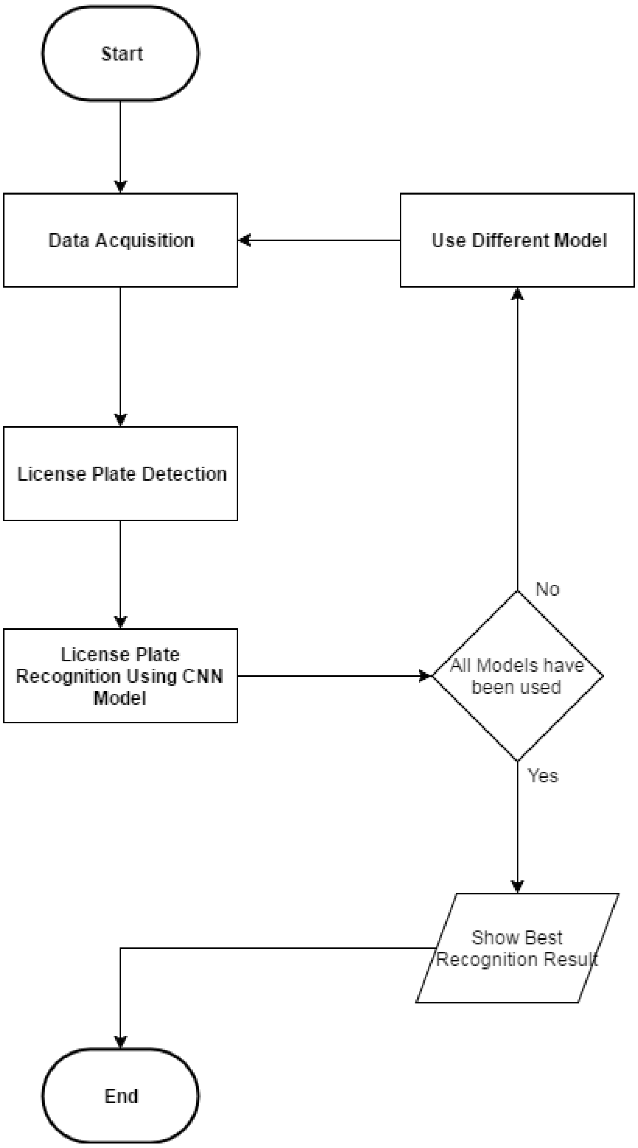

Figure 4.4 Diagram of the Experiment

4.3 Evaluation

The Sliding Window algorithm will be evaluated by calculates the average precision. The model of the Sliding Window architecture like the amount of layer and the amount of the hidden layer uses in the architecture will be changed to compare the results. The number of the epoch to train the Sliding Window algorithm will be 100 epochs. The result from the different model will be compared and the best result will be used.

The research will be evaluated process by process. The Enhancement process will be evaluated by comparing the result when using HE and the result when not using HE. The difference of the result of using HE and not using HE can be used to evaluate the next process. Evaluation will also use to evaluate the next process. The Recognition process will be evaluate by comparing the accuracy when using HE and when not using HE in the enhancement process, the evaluation will show the effect of HE for the image with various contrast and lighting. The evaluation will also show the best way of using HE. The evaluation for the thresholding process will be done by changing the Gaussian kernel to get the best thresholding result.

The CNN will be evaluated by how many hidden layers were used and how many epochs the model will be train. The result will show the performance of the CNN and from the result can be concluded which architecture or model is better for these research. The Evaluation will be performed using different data than the training and validating data obtain from the server.

4.2.1 Average Precision

The average precision is the average of the precision across the different values of the model threshold. By using different model threshold, the result will obtain different precision and recall values. The precision is the ratio of true positive and the total of positive predictions. The recall value is the ratio of true positive and the total number of objects.

$$TP = True\ Positive$$
$$TN = True\ Negative$$
$$FP = False\ Positive$$
$$FN = False\ Negative$$

$$Precision = \frac{TP}{TP + FP} \tag{4.1}$$

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

Equation (4.1) and (4.2) represents the formula to calculate the precision and recall of a model respectively. In the context of object detection, true positive is the correct object that is being detected, true negative represents when there is detection when there is no object within the image, false positive represents the number of incorrect detection, and false negative represents when there is an object but it is not detected. The average precision can be said to be the average calculated maximum precision for every top recall calculated with respect to the order of highest confidences value.

4.2.2 Mean Average Precision (MAP)

The MAP is used of evaluate the detection algorithms and it is the product of precision and recall of the detected bounding boxes. The MAP can be calculated by the AP (Average Precision) values for each class, then the average over the class and over all the IoU (Intersection over Union) threshold. A Detection is considered to be true positive only if the MAP is above 0.5.

4.4 Implementation

In this research, the following computer specifications and library used are:

1. PC Specification

   a. CPU :Intel Core i7
   b. RAM :8GB
   c. GPU :Nvidia GTX 820m 8GB

2. Library

   a. Numpy :A python Library used for numerical computation
   b. Matplotlib :A python Library used to plot graphs for visualization
   c. Tensorflow :A machine learning library
   d. OpenCV :A python Library used to image processing

CHAPTER V RESEARCH SCHEDULE

Schedule of research is made as a reference in completing the study in accordance with the stages and targets to be achieved within a predetermined time. The study schedule for this study is listed in the inside of Table 5.1

Table 5.1 Research Timeline

No	Activity	Time	Output
1	Literature Study	November 1 – November 30, 2018	Determine the problem from previous works
2	Analysis	November 18 – December 8, 2018	Define the method for the problem from
3	Data Acquisition and Annotation	December 9, 2018 – January 26, 2019	Collect 1000 data for testing, training and validating, and collet annotated data for
4	CNN Model Design	December 16, 2018 – January 19, 2019	Define the model uses to experiment and evaluation
5	Experiment and Implementation	January 20 – February 23, 2019	Collect the results from the experiment process
6	Evaluation	February 17 – March 9, 2019	Find the best model and method with high accuracy
7	Thesis Report	November 1, 2018 – March 31, 2019	Final Thesis Report

CHAPTER VI REFERENCE

A., M., A., Y., A., M., 2013. Automated Edge Detection Using Convolutional Neural Network. International Journal of Advanced Computer Science and Applications 4. https://doi.org/10.14569/IJACSA.2013.041003

Agarap, A.F., 2018. Deep Learning using Rectified Linear Units (ReLU). arXiv:1803.08375 [cs, stat].

Albawi, S., Mohammed, T.A., Al-Zawi, S., 2017. Understanding of a convolutional neural network, in: 2017 International Conference on Engineering and Technology (ICET). Presented at the 2017 International Conference on Engineering and Technology (ICET), IEEE, Antalya, pp. 1–6. https://doi.org/10.1109/ICEngTechnol.2017.8308186

Aloysius, N., Geetha, M., 2017. A review on deep convolutional neural networks, in: 2017 International Conference on Communication and Signal Processing (ICCSP). Presented at the 2017 International Conference on Communication and Signal Processing (ICCSP), IEEE, Chennai, pp. 0588–0592. https://doi.org/10.1109/ICCSP.2017.8286426

Borker, S., 2015. Implementation of Vehicle License Plate Recognition Using Canny Edge Detection. Engineering Technology 6, 4.

Feng, Y., Zhang, J., Wang, S., 2017. A new edge detection algorithm based on Canny idea. Presented at the 2ND INTERNATIONAL CONFERENCE ON MATERIALS SCIENCE, RESOURCE AND ENVIRONMENTAL ENGINEERING (MSREE 2017), Hubei Province, China, p. 040011. https://doi.org/10.1063/1.5005213

Guo, T., Dong, J., Li, H., Gao, Y., 2017. Simple convolutional neural network on image classification, in: 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)(. Presented at the 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA), IEEE, Beijing, China, pp. 721–724. https://doi.org/10.1109/ICBDA.2017.8078730

Hansson, M., Olsson, C., 2017. Feedforward neural networks with ReLU activation functions are linear splines 36.

Hatem, A.J., 2013. Adaptive Canny Algorithm Using Fast Otsu Multithresholding Method 26, 8.

I Wayan Suartika E. P, Arya Yudhi Wijaya, Rully Soelaiman, 2016. Klasifikasi Citra Menggunakan Convolution.

Johnson, J., Karpathy, A., 2015. CS231n Convolutional Neural Networks for Visual Recognition [WWW Document]. Stanford CS class. URL https://cs231n.github.io/convolutional-networks/ (accessed 5.5.18).

Keong, W.W., Iranmanesh, V., 2016. Malaysian automatic number plate recognition system using Pearson correlation, in: 2016 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE). Presented at the 2016 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), IEEE, Penang, Malaysia, pp. 40–45. https://doi.org/10.1109/ISCAIE.2016.7575034

Ktata, S., Khadhraoui, T., Benzarti, F., Amiri, H., 2015. Tunisian License Plate Number Recognition. Procedia Computer Science 73, 312–319. https://doi.org/10.1016/j.procs.2015.12.038

Makandar, A., Halalli, B., 2015. Image Enhancement Techniques using Highpass and Lowpass Filters. International Journal of Computer Applications 109, 21–27. https://doi.org/10.5120/19256-0999

Massoud, M.A., Sabee, M., Gergais, M., Bakhit, R., 2013. Automated new license plate recognition in Egypt. Alexandria Engineering Journal 52, 319–326. https://doi.org/10.1016/j.aej.2013.02.005

Putra, R.D., Purboyo, T.W., Prasasti, A.L., 2017. A Review of Image Enhancement Methods 12, 8.

Reddy, A.S., Rani, C.U., Sudhakar, M., 2017. An Effective Thresholding Technique for Otsu's Method using Contrast Enhancement 6, 9.

Surekha, P., Gurudath, P., Prithvi, R., Ananth, V.G.R., 2018. AUTOMATIC LICENSE PLATE RECOGNITION USING IMAGE PROCESSING AND NEURAL NETWORK. ICTACT JOURNAL ON IMAGE AND VIDEO PROCESSING 08, 7.

Tabrizi, S.S., Cavus, N., 2016. A Hybrid KNN-SVM Model for Iranian License Plate Recognition. Procedia Computer Science 102, 588–594. https://doi.org/10.1016/j.procs.2016.09.447

Tur, M., 2012. Network Routing by Artificial Neural Network. 2012 Military Communications and Information Systems Conference (MCC) 5.

Wang, Y., Ban, X., Chen, J., Hu, B., Yang, X., 2015. License plate recognition based on SIFT feature. Optik - International Journal for Light and Electron Optics 126, 2895–2901. https://doi.org/10.1016/j.ijleo.2015.07.040

Xiang, H., Zhao, Y., Yuan, Y., Zhang, G., Hu, X., 2019. Lightweight fully convolutional network for license plate detection. Optik 178, 1185–1194. https://doi.org/10.1016/j.ijleo.2018.10.098

Xiaohong Long, Jing Zhou, 2013. An Efficient Method For Dark License Plate Detection. International Journal of Digital Content Technology and its Applications 7, 477–485. https://doi.org/10.4156/jdcta.vol7.issue1.55