Ministry of Education and Science of the Republic of Kazakhstan
Suleyman Demirel University

Nurassyl Yerzhanov

# Development and implementation of license plate recognition algorithm software for automation of the checkpoint

THESIS

Presented in Partial Fulfillment for the
Degree of Master of Science in Computer Science
(degree code: 7M06102)
**Department of Computer Sciences**
Faculty of Engineering and Natural Sciences

Supervisor: **Meiram Murzabulatov**

Kaskelen, 2022

**Suleyman Demirel University**

***Faculty of Engineering and Natural Sciences***

**Department of Computer Science**

Dean of Faculty

Associate Professor, PhD Zhamanov A.

_______________

«20» 06 2022

**Topic of the thesis:**

**Development and implementation of license plate recognition algorithm software for automation of the checkpoint**

Thesis submitted as part of the requirements for the award of the MSc in "7M06102 - Computer Science"

Head of Department Assoc. Prof. Cemil Turan

Academic Supervisor Assist. Prof. Meiram Murzabulatov

Master's student Yerzhanov Nurassyl

Kaskelen, 2022

# Abstract


Technologies that identify cars by license plates are an important aspect of traffic control and safety, and are used in various fields: protected areas, enterprises, traffic control, gas stations, parking lots, entry and exit control, etc. The purpose of the work is to research and develop methods of segmentation and character recognition, providing information processing on the image for identification of license plates of cars.

Achieving this goal involves solving the following tasks:

1. research of the main methods of processing, segmentation and recognition of objects in images;

2. development and implementation of the number plate segmentation algorithm;

3. development and implementation of a character recognition algorithm;

4. investigation of the reliability of the developed algorithms for the recognition of a car license plate.



**Key words:** image processing, segmentation, recognition, Laplacian, erosion, dilation, binarization, median filtering, contour analysis, Viola-Jones method, support vector machine method, histogram equalization, Haar signs.

# Аңдатпа

Нөмірлік белгілері бойынша автомобильдерді анықтайтын технологиялар жол қозғалысы мен қауіпсіздікті басқарудың маңызды аспектісі болып табылады және әртүрлі салаларда қолданылады: қорғалатын табиғи аумақтар, кәсіпорындар, жол қозғалысын басқару, жанармай құю станциялары, автотұрақтар, кіру және-шығу бақылау пункттері және т. б. Жұмыстың мақсаты - автокөліктердің нөмірлік белгілерін анықтау үшін суреттегі ақпаратты өңдеуді қамтамасыз ететін сегменттеу және таңбаларды тану әдістерін зерттеу және дамыту жолдарын көрсету.

Осы мәселелерді шешу арқылы мақсатқа қол жеткіземіз:

1. суреттердегі нысандарды өңдеудің, сегментациялаудың және танудың негізгі әдістерін зерттеу;
2. нөмірлік белгілерді сегменттеу алгоритмін әзірлеу және енгізу;
3. таңбаларды тану алгоритмін жасау және енгізу;
4. әзірленген автомобиль нөмірлерін тану алгоритмдерінің сенімділігін зерттеу.

**Түйін сөздер:** суретті өңдеу, сегментация, тану, лапласиан, эрозия, дилатация, бинаризация, медианалық сүзу, контурлық талдау, Виола-Джонс әдісі, тірек векторларының машиналық әдісі, гистограммаларды туралау, Хаар белгілері.

# Аннотация

Технологии, которые идентифицируют автомобили по номерным знакам, являются важным аспектом управления дорожным движением и безопасности и используются в различных областях: охраняемые территории, предприятия, управление дорожным движением, заправочные станции, автостоянки, контроль въезда и выезда и т.д. Целью работы является исследование и разработка методов сегментации и распознавания символов, обеспечивающих обработку информации на изображении для идентификации номерных знаков автомобилей.

Достижение этой цели предполагает решение следующих задач:

1. исследование основных методов обработки, сегментации и распознавания объектов на изображениях;

2. разработка и внедрение алгоритма сегментации номерных знаков;

3. разработка и внедрение алгоритма распознавания символов;

4. исследование надежности разработанных алгоритмов распознавания номерного знака автомобиля. ателя.

**Ключевые слова:** обработка изображений, сегментация, распознавание, лапласиан, эрозия, дилатация, бинаризация, медианная фильтрация, контурный анализ, метод Виолы-Джонса, машинный метод опорных векторов, выравнивание гистограмм, знаки Хаара.модель

# Contents

# Chapter 1

# Introduction

## 1.1 Motivation

One of the most rapidly increasing fields in the world of information technology is object detection in photographs. Such systems are in demand in a range of fields, ranging from security to various forms of medical diagnostics. Item recognition in photographs is the assignment of a class to an object based on the basic traits that distinguish it from the others.

Identification of automobiles by license plates is a vital part of traffic management and safety, and it is used in a variety of settings, including protected areas, businesses, traffic control, gas stations, parking lots, entrance and exit control, and so on.

## 1.2 Aims and Objectives

The goal of this project is to investigate and develop segmentation and character recognition algorithms for image processing in order to identify license plates on automobiles.

Achieving this goal involves solving the following tasks:

1. research of the main methods of processing, segmentation and recognition of objects in images;
2. development and implementation of the number plate segmentation algorithm;
3. development and implementation of a character recognition algorithm;
4. investigation of the reliability of the developed algorithms for the recognition of a car license plate.

## 1.3 Thesis Outline

The first section of the study provides a problem specification, an analytical assessment of the literature, and an overview of number plate segmentation and recognition methods, as well as picture preprocessing approaches.

The second section discusses the chosen segmentation and recognition algorithms, as well as the principles of their operation, as well as the training and test samples generated throughout this project. The suggested method and its software implementation are also described in this section.

The third section presents the results of testing the developed algorithms for number plate recognition. As a result of testing, algorithms were selected that show the best accuracy of segmentation and recognition of license plates.

The scientific novelty of the work lies in the development of an algorithm for recognizing number plates in the image.

# Chapter 2

# ANALYTICAL REVIEW

The recognition of detected license plates is the subject of this research. The steps for automatic recognition are as follows:

1. picture preprocessing;
2. license plate segmentation into individual characters;
3. segmented character recognition

The goal of picture preprocessing is to remove noise, improve quality, increase important information while suppressing undesired data, conduct geometric changes, and adjust brightness and contrast.

The image is separated into familiar ones in the second step, i.e. the sections of particular characters are segmented. A segmented character recognition process is used in the third step, which produces a string of characters.

## 2.1 Analytical review of similar programs and systems

### 2.1.1 Software "Marshal"

Car park is an independent application developed by the research and production company Mullen Systems for automatic recognition and registration of state registration signs of cars. Figure 1.1 shows the composition of the system equipment for typical single-track control.

Automarshal software only works on the Windows operating system. One or more IP cameras are connected to the computer via Ethernet. The system recognizes car numbers and analyzes the account of the video coming from the cameras and sent all the Information (time, direction, car photo, car number, camera name, reviews, etc.) to the database.

System functionality: allows users to manage access lists, generate various reports, download databases, manage gate and traffic lights, interact with access control and management systems (ACS), and CCTV systems.

The basic version of the auto racing software includes:


Figure 2.1: Typical composition of equipment for single-track control

- recognition of state registration marks of cars on images;
- support for the car's own database * ability to download the database from formatted files in manual mode .xls, * .xlsx and * .csv;
- automatic verification of the number recognized by the user;
- visual and audio notification to the operator if the database matches;
- search in the database of vehicles defined for the specified user, create and print a report on the search results;

Main features of the car park system:
- Vehicle speed recognition capability up to 98 % to 150 km/h;
- Support for CIS and European Union (29 countries) numbers;
- 2 parallel recognition algorithms;
- 8 recognition channels in one system (connected cameras);
- quick start without additional settings;
- control of external devices: gates, traffic lights, etc.;

### 2.1.2 SecurOS Auto

The SecurOS Auto intelligent video analysis system pro- vides recognition of state registration plates (GRZ) of vehicles (TS). The neural network classifier determines the make, model, color of each passing car, as well as the vehicle category (A, B, C, D).

The scope of application covers a significant range of tasks: from ensuring security in parking lots to controlling traffic flows on a regional and federal scale.

SecurOS Auto operates on the basis of the SecurOS video management integration platform, which allows you to create a security complex with the functionality necessary for the customer.

Main features:
- Intelligent algorithms Confident recognition of the GRZ of cars at speeds up to 320 km/h in a wide range of external conditions;

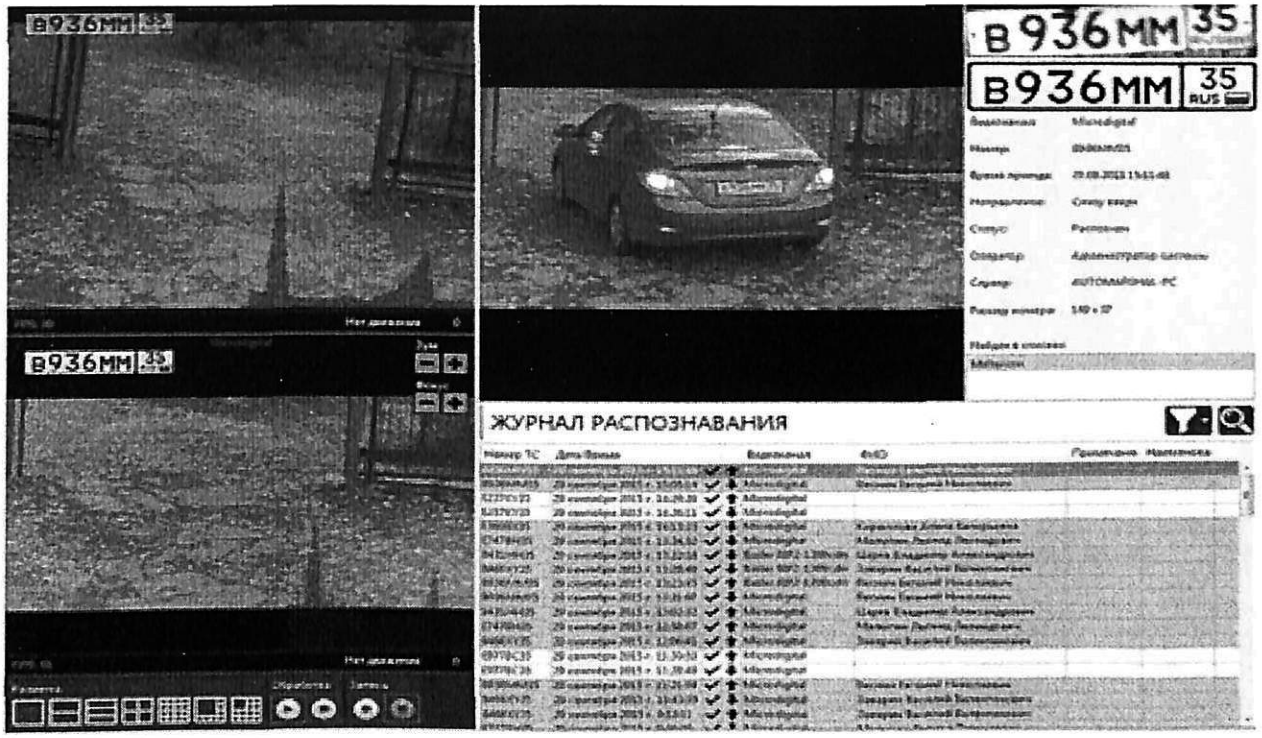


Figure 2.2: An example of the system

- Neural Network Classifier Determination of the make, model and color of the car. Determination of the vehicle category by driver's license: A (motorcycles), B (cars with a permitted maximum mass of less than 3.5 tons), C (cars with a permitted maximum mass of more than 3.5 tons), D (vehicles designed to carry 8 or more passengers);
- Hardware-independent system Lack of binding to the equipment of certain manufacturers;
- Control of several traffic lanes in 2 directions at the same time Detection of all number plates in the detection zone of one camera.

## 2.2 Overview of image preprocessing methods

Image processing helps to improve the speed and quality of character identification and segmentation. Image conversion to grayscale, binarization, Gaussian smoothing, median filtering, erosion, dilation, boundary selection based on the Laplace operator, and histogram equalization were all investigated to achieve this aim.

### 2.2.1 Grayscale image translation and binarization

Since it allows you to execute a compact description of information about the intensity of pixels, the procedures of image conversion to grayscale and subsequent binarization are of relevance in solving issues of object recognition in pictures.

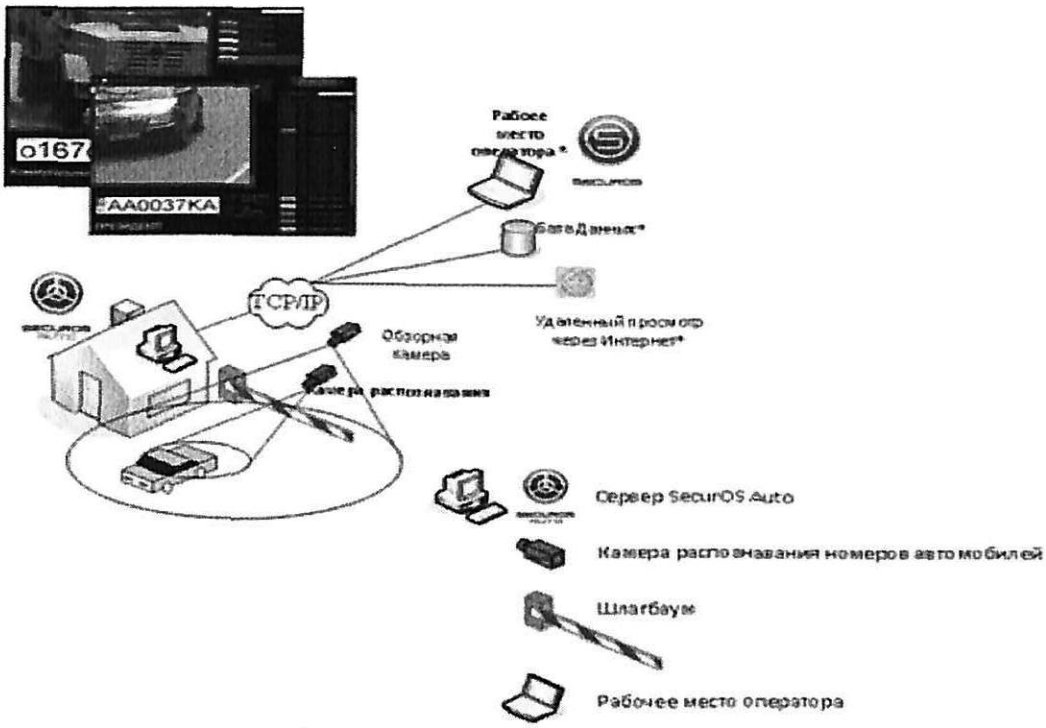

Figure 2.3: Operation of the SecurOS Auto system

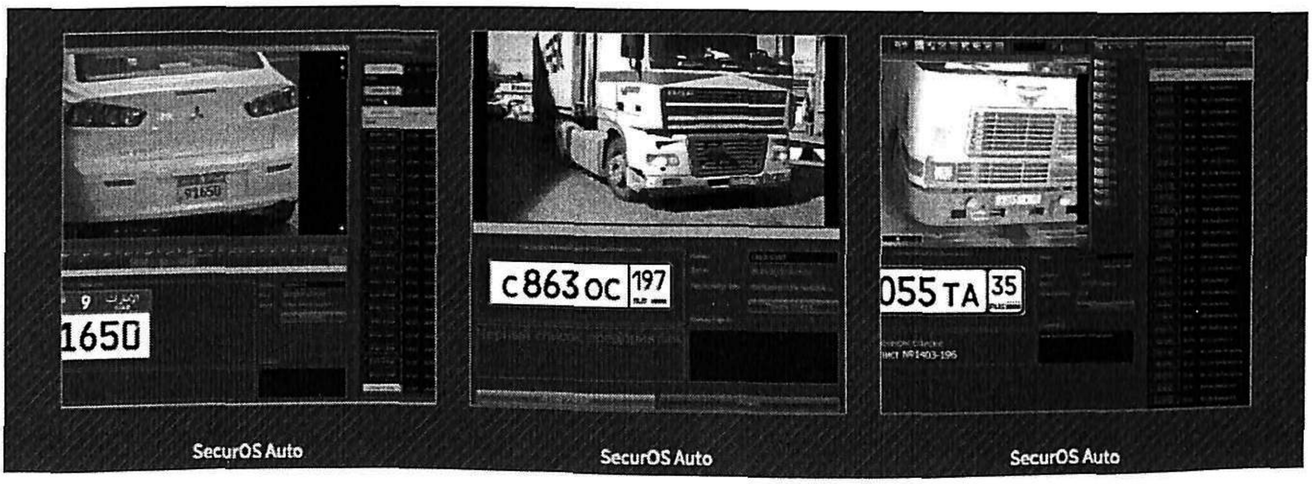


Figure 2.4: Example of the composition

Grayscale conversion decreases the quantity of data processed while keeping brightness and removing extraneous features.

To convert a color image to grayscale, use the formula (1) to calculate the brightness of each point, then transfer the resultant image via all three channels (R = B = G = Y), where R, G, B is the value of red, green, and blue colors at the processed point [3].

As a result, the image is converted to gray, with 256 different shades, from black (0) to white (255). Figure 1 shows an example of the original image (a) and the result of its translation into grayscale (b).

The process of transforming a grayscale image to a binary image is known as image binarization. Only one of two-color values, conditionally equal to 0 or 1, can be used in a binary picture. The backdrop is made up of pixels with a value

$$Y = 0.3R + 0.59G + 0.11B$$


Figure 2.5: Application of Grayscale

of 0 while the foreground is made up of pixels with a value of 1 [4].

The halftone image's spatial distribution of brightness is somewhat reproduced by the arrangement of 0 and 1. As a result of effective binarization, the binary picture more correctly reproduces the features and borders of the halftone image's objects [4].


Figure 2.6: The result of binarization of the image in Fig. 2.5

## 2.2.2 Removal of high frequency noise

Images of license plates shot in natural settings are frequently polluted, necessitating noise reduction beforehand. High-frequency noise is removed using median filtering and Gaussian blurring.

**Median filtering**

The brightness value of the current pixel is replaced by the median brightness value of all items in its neighborhood during median filtering. When compared to linear smoothing filters of similar sizes, median filters function effectively for various forms of random noise and have a very little defocusing impact. In addition, median filters perform well while filtering pulse noise.

The median filtering technique consists of the following steps:

1. ranking the brightness values of pixels in the vicinity of the current pixel in ascending order;

2. determining the median value among the ordered values; 3.

3. Assign the received value to the element that has been processed.

For example, the median for a 3x3 neighborhood will be the fifth magnitude value, while the median for a 5x5 neighborhood will be the thirteenth magnitude value. If the values of numerous items in the neighborhood are equal, they will be grouped together [3].[1]

Figure 2.2.2 shows an example of a numbered plate after applying median filtration with a neighborhood of 5.

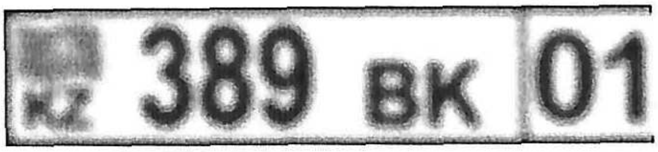


Figure 2.7: The result of applying median filtering

**Gaussian blur**

Gaussian blur is the convolution of an image with the function Gauss:

$$g(x,y) = Ae^{-\frac{x^2+y^2}{\sigma^2}}$$

where $\sigma$ is a parameter specifying the degree of blurring, A is a parameter providing rationing. The resulting pixel brightness depends on the brightness of neighboring pixels according to the law given by the function Gauss. For $\sigma =$ const, the degree of blurring is proportional to the size of the matrix filter.

It is not recommended to use this filter if it is necessary to emphasize the borders, as they can blur important details along the borders. But it is possible to modify this method to better adapt to the boundaries. To do this, you need:

1. search for the largest blur direction in each window;
2. apply a directed Gaussian along the found boundary.

As a result, there will be no blurred contour due to the fact that smoothing was carried out along the borders of the image [2]. Figure 2.8 shows an example of a numbered plate after applying a Gaussian blur with a neighborhood size of 5×5.

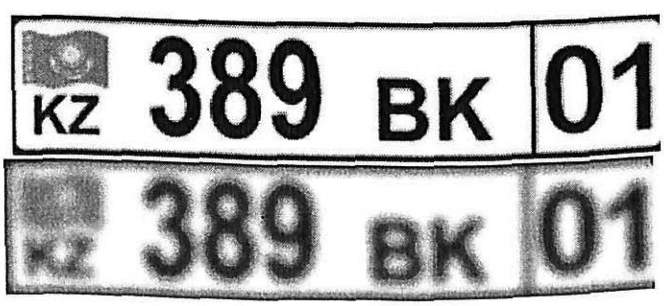


Figure 2.8: The result of applying Gaussian blur: a) the original image in grayscale; b) the result of applying the Gauss filter

### 2.2.3 Equalization of histograms

This method is based on the analysis of the histogram of the image. An image histogram is a discrete function of the following form:

$$f(i) = \frac{n_i}{n}$$

where $f(i)$ is the number of pixels of the image having the current brightness value $i(i = 0...255$, from black to white); $n$ is the total number of pixels contained in the image. The function $f(i)$ is normalized: $0 \leq f(x) \leq 1$. Brightness values are plotted along the abscissa axis: $0 \leq i \leq 255$, along the ordinate axis – values $0 \leq f(x) \leq 1$.

The equalization method uniformly increases the range between the brightness values present in the image, thereby increasing its contrast, thereby the difference in the brightness of the semitones will be greater. According to the formula (4), the histogram is transformed so that it accepts the maximum brightness range.

$$S_k = g(x_k) = \sum_{i=0}^{k} f(x_i) = \sum_{i=0}^{k} \frac{n_i}{n}$$

where $S_k$ is the total pixel brightness, $f(x_i)$ is the histogram value at $x_i$ point, $k$ is the range. The values of $S_k$ are in the range $0 \leq S_k \leq 1$. To change the value of the output pixels in the range $0 \leq S_k \leq 255$, you need to multiply by 255 [10]. After the conversion, the image will have the following appearance as in Figure 2.9.

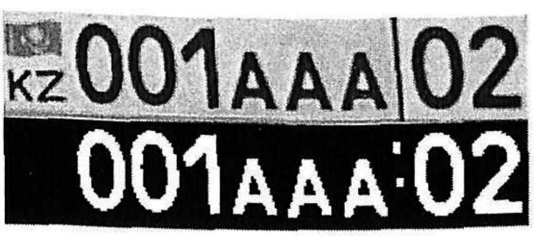


Figure 2.9: The result of applying histogram equalization: a) the original image in grayscale; b) the result of applying histogram equalization

### 2.2.4 Morphological operations: erosion and dilation

**Dilation**

Let $A$ and $B$ be sets of space $Z$. The dilation of the set $A$ by the set $B$ is defined as:

This relation is based on obtaining the central reflection of the set $B$ relative to its origin and then shifting the resulting set to the point $z$. In this case, the

$$A \oplus B = \{z | (\hat{B})_z \cap A \neq \emptyset\}$$

dilation of the set $A$ by $B$ is the set of all displacements $z$ in which the sets $B_1$ and $A$ coincide in at least one element. The set $B$ is called a structure-forming set or a dilation primitive [6].

**Erosion**

For the sets $A$ and $B$ from the space, $Z^2$ the erosion of $A$ by $B$ is defined as:

$$A \ominus B = \{z | (\hat{B})_z \subseteq A\}$$

The erosion of the set $A$ by the primitive $B$ is the set of all points $z$, when shifted to which the set $B$ is entirely contained in $A$. As in the case of dilation, this equation is not the only way to determine the erosion procedure[6].

All items smaller than structural objects connected by thin lines get detached as a result of the erosion procedure, and their sizes decrease. Erosion screens out any things smaller than the structural element while also preventing a significant drop in object size. After erosion, dilation is used to fill gaps and extend the surviving shapes.

### 2.2.5 Laplacian

The Laplacian operator based on the second derivative of the yar-bone image function is isotropic, and for two variables $f(x, y)$ is defined as:

$$\nabla^2(f) = \left(\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}\right)$$

The use of Laplacian emphasizes gaps and suppresses weak changes between brightness levels. The result is an image containing gray lines in place of contours and other breaks [3].

Figure 2.10 shows an isotropic mask for turns at angles multiple of 45°. They are based on the definition of the Laplacian being "negative".

## 2.3 Segmentation Methods

The technique of splitting a digital image into distinct sections (different objects) with the same texture or color is known as image segmentation. A set of regions (contours) is retrieved from the picture as a consequence of segmentation. The color, texture, and intensity of all pixels from the same region are comparable

| | | | | | | |
|---|---|---|---|---|---|---|
| 1 | 1 | 1 | | -1 | -1 | -1 |
| 1 | -8 | 1 | | -1 | -8 | -1 |
| 1 | 1 | 1 | | -1 | -1 | -1 |

Figure 2.10: Example of Laplace operator masks

in certain ways. The qualities listed above distinguish adjacent places from one another. Different methods for determining region borders are based on inhomogeneities non brightness intensity levels. As a result, the method of picture segmentation used is determined by the problem to be solved.[2]

Segmentation algorithms are classified as follows:

1. Threshold algorithms;
2. Algorithms for the expansion of regions;
3. Boundary algorithms;
4. Segmentation based on clustering.

### 2.3.1 Threshold algorithms

Threshold algorithms define certain upper and lower limitations. The image elements with a brightness level greater than the threshold will have a value of 1, which is less than the threshold value of 0. After the transformation in the image matrix, the image elements with a brightness level greater than the threshold will have a value of 1, which is less than the threshold value of 0. Histogram analysis is used to determine the optimal threshold value. Then there's multi-threshold segmentation [3].

Threshold algorithms can segment basic images, but they don't work well with images that include uneven illumination, flashes, or a variety of interference. There are methods that assess the weighted values of the extremes, or rather the intensity and gradient, to reduce the impact of the stated drawbacks [5].[3] For the correct use of the algorithm, it is necessary to avoid "deviation" when choosing a threshold value:

1. It is necessary to strictly control the identity of the distribution in the light and dark areas of the brightness histogram.

2. It is necessary to divide the image into small elements so that the brightness histogram has a vivid expression of extremes.

3. The elements should be quite large. The volume of the statistical sample should allow us to satisfactorily assess the positions of extremes and describe the neighborhood.

### 2.3.2 Regional sprawl algorithms

On pictures with areas with reliable connection within individual segments, the regional sprawl algorithms work well. Neighboring components with the same or comparable brightness levels are clustered as a result of the operation of such algorithms, and finally merge into homogenous areas [2].

Algorithms for growing areas have a number of drawbacks, including the fact that they typically emphasize similar pieces while omitting information regarding variations in brightness inside the areas and probable limits. When working with noisy photos, however, these algorithms have shown to be more successful.

### 2.3.3 Boundary algorithms

The term "boundary algorithms" refers to algorithms that discover points on the edges of areas [8]. The most important parameters for establishing borders are pixel brightness levels. Texture and gradient are other often utilized qualities.

The borders of the items in the image limit the amount of data that must be processed while preserving critical information about the objects in the image, such as their shape, size, and number. Segmentation algorithms have specific applications that depend on:

- input data
- recognition requirements
- calculation volume
- speed

When choosing a segmentation algorithm, the most popular methods were considered. These include the Roberts operator, Sobel, Previtt, and the Canny boundary detector.

**Sobel operator**

The Sobel operator is a discrete differential operator. This operator calculates the approximate value of the brightness gradient of the image. As a result of applying the Sobel operator, an image is obtained, each point of which will be either a brightness gradient vector or a norm at this point [9].[4]

This operator consists of two matrices, 3×3 in size. The second matrix differs from the first one in that it looks for vertical boundaries, and the first one is horizontal.

**Robert's operator**

Robert's operator performs simple and fast calculations of a two-dimensional spatial dimension on an image. This method emphasizes areas of high spatial frequency, which often correspond to edges. A grayscale image is fed to the operator's input. The value of the pixels of the output image at each point assumes a certain value of the spatial gradient of the input image at the same point.

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

$G_x$

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

$G_y$

Figure 2.11: Example of Sobel operator masks

| | |
|---|---|
| -1 | 0 |
| 0 | 1 |

$G_x$

| | |
|---|---|
| 0 | -1 |
| 1 | 0 |

$G_y$

Figure 2.12: Example of Robert's operator masks

The detection of boundaries by this computational method is much simpler than the Sobel method, but leads to frequent false triggering of the filter in point bursts of brightness, which leads to noise in the resulting image.

**Canny Boundary Detector**

The Canny boundary detector is one of the most popular contour detection algorithms. An important step in this algorithm is to eliminate noise on the contours, which can significantly affect the result, while it is necessary to preserve the boundaries as much as possible. This requires careful selection of the threshold value during processing.

Algorithm:

- blurring the output image $f(r, c)$ using the GAUSS function:

$$\hat{f}(r,c).\hat{f}(r,c) = f(r,c) * G(r,c,6)$$

- gradient search. The boundaries are outlined where the gradient takes the maximum value;

- totals of the borders are determined by the time of the suppression of all the edges that are not connected with certain boundaries.

### 2.3.4 Segmentation based on clustering

The benefit of clustering-based segmentation algorithms is that they automate, the process of determining the value of parameters for class separation. Many clustering methods exist, including k-means, CURE, BIRCH, and Viola-Jones. These algorithms look for clusters that match a static model [10].

The model's specified parameters have a direct impact on the mentioned algorithms. The software may fail if the parameters are incorrectly selected in respect to the classified data. Even if the model covers a substantial number of cluster properties, it may fail. When the data consists of clusters of varying sizes: form, size, and density, algorithms might make mistakes.

**Regional histogram analysis**

The first and easiest way to understand the algorithm is to analyze regions on a histogram. The method is based on the fact that the characters in the number, background, and frame are contradictory. Simply put, after dividing the boundaries of the image, a histogram is constructed. This number is also the maximum value of points. The example is shown in Figure 2.13.

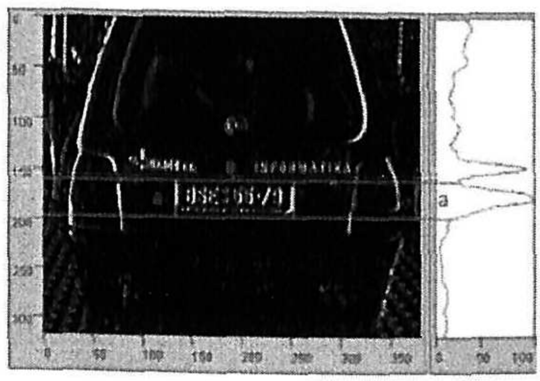

Figure 2.13: Regional histogram analysis

As you can see, the global maximum is fixed. Next, knowing the proportions of the numbers, you can draw conclusions about the length of the frame. After that, the candidate searches for a number - a vertical line, after which the brightness often changes.

According to GOST, the dimensions of the room are 520 millimeters and the width is 112 millimeters. The figure of GOST can be seen in Figure 2.14:

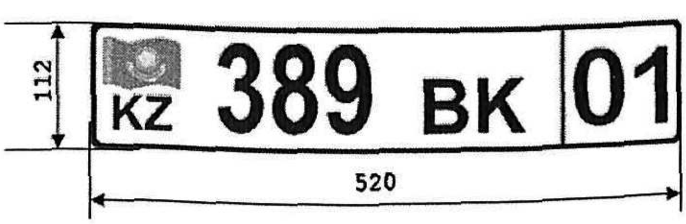


Figure 2.14: Sample of the state transport registration number of the Republic of Kazakhstan

Moving the distance b from the vertical boundary of the number. This will be the desired frame. After that, the resulting rectangle proceeds to the next stage.

Obviously, this algorithm is unstable to noise and dirt. In addition, in the case of a large perspective (the angle of inclination of the camera when shooting is large) or a small image compared to objects around the machine, it is practically difficult and impossible to separate the maximum projection.

To sum up, this algorithm is easy to understand and implement, but it only works when the machine captures foreign objects as the main object of the image, without shooting them and at a right angle. In other cases, the algorithm does not work. In this particular case, this algorithm cannot be used.

**The Viola Jones method**

The Viola-Jones method is one of the most effective and fast-acting of the existing methods. The method finds objects even when rotated by 30 degrees, but when the slope increases, detection occurs with errors. The method is characterized by the use of integral representation of images, Haar and boosting features. Haar signs or cascades are a set of black and white rectangular masks of different shapes. The mask is superimposed on some part of the frame, then the algorithm adds up the brightness of all the pixels of the image that appeared under the black and under the white part of the mask, after which the difference of these values is calculated, that is, the convolution of the frame with the mask is calculated. Next, the obtained result is compared with a certain threshold value and, thus, objects are found.

Haar signs or Cascades are a set of rectangular masks of various shapes, black and white. The mask is placed opposite one part of the frame, then the algorithm adds the brightness of all pixels of the image under the black and white parts of the mask, and then the difference between these values is calculated, i.e., the conversion frame with the mask is calculated. Next, the result is compared with a certain threshold value and, accordingly, objects are found.

In 2001, Paul Viola and Michael Jones proposed an algorithm for finding pages using Haar primitives. The essence of the method is to find in the image the characteristic features of a person's face - eyes, nose, mouth. The Haar primitives in Figure 2.15 show exactly these signs.

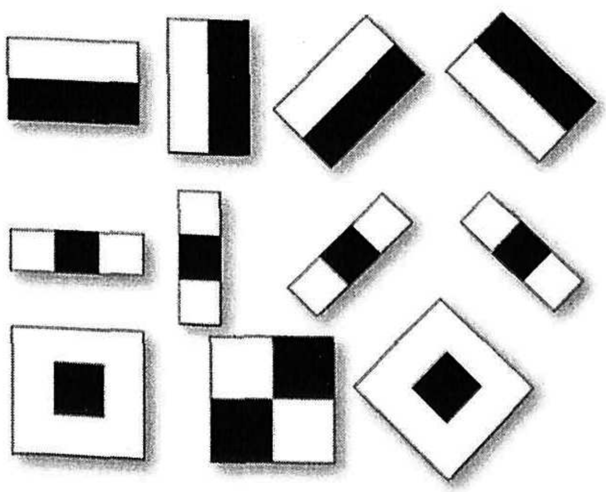

Figure 2.15: Haar primitives

Simply put, the essence of this algorithm is that the window will move to the image where any primitives will be used.

This window is fixed if signs are found that meet the task condition. The Haar cascade, trained in number recognition, has been added to the OpenCV library - a library of image processing algorithms, computer vision, and general-purpose open-source numerical algorithms.

As an algorithm for finding numbers, we got the result of 90 % correct definition. The operation of the algorithm is usually quite complex. The main steps are shown in Figure 2.16.[5]

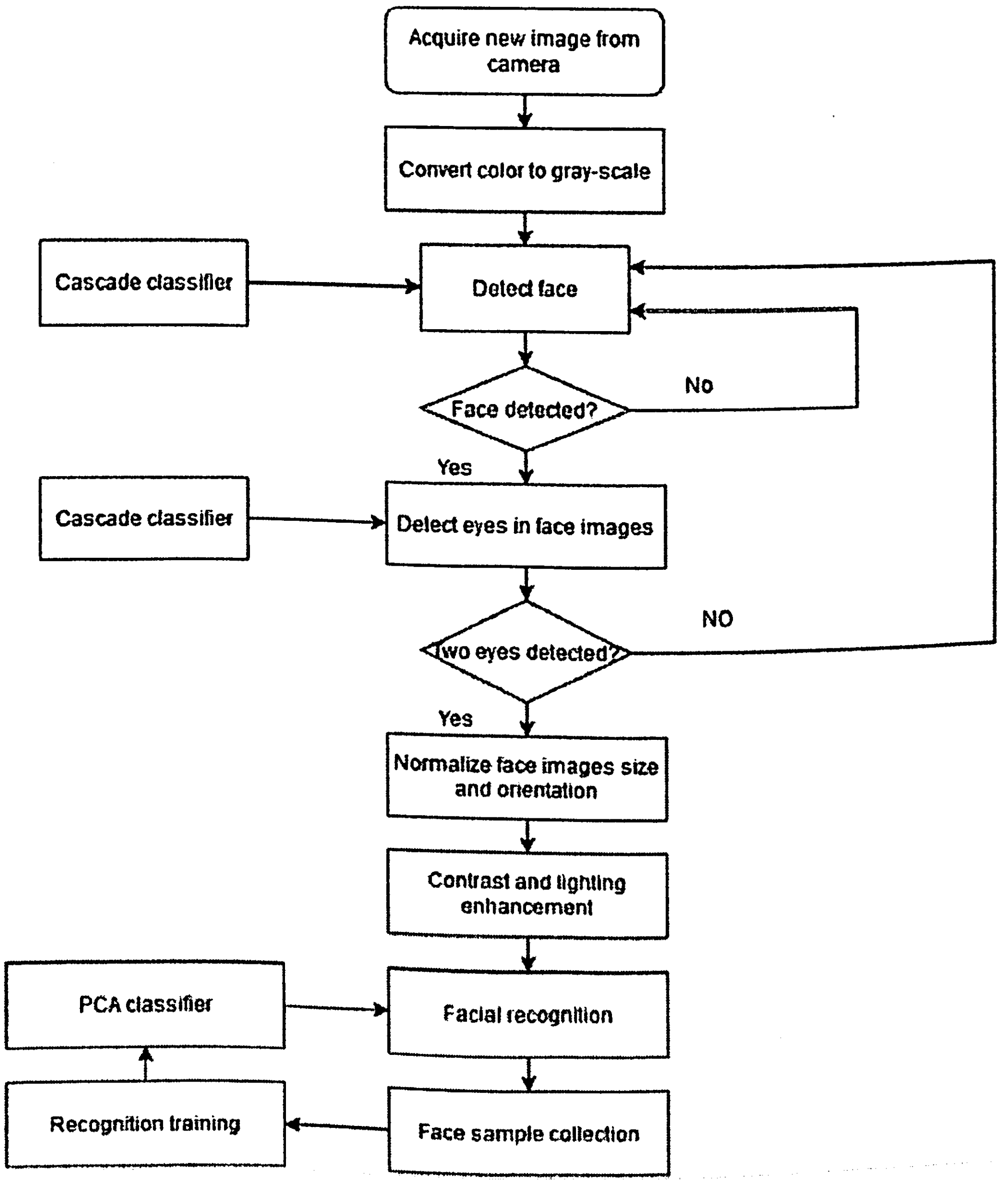


Figure 2.16: Recommended algorithm

**Neural network method**

The NVIDIA Deep Learning GPU Training System (DIGITS) gives data scientists and researchers access to deep learning's potential. DIGITS allow you to do standard deep learning operations including data management, network definition, parallel training, real-time monitoring of training performance, and selecting the best model from the results browser. DIGITS is fully interactive, allowing you to concentrate on network design and training rather than programming and troubleshooting.

DIGITS 4 has a novel object detection approach that allows you to train networks to recognize items in pictures (such as persons, automobiles, or pedestrians) and identify constraints in their vicinity. See the Deep Learning message to identify items using numbers for a step-by-step instruction to using this new feature.

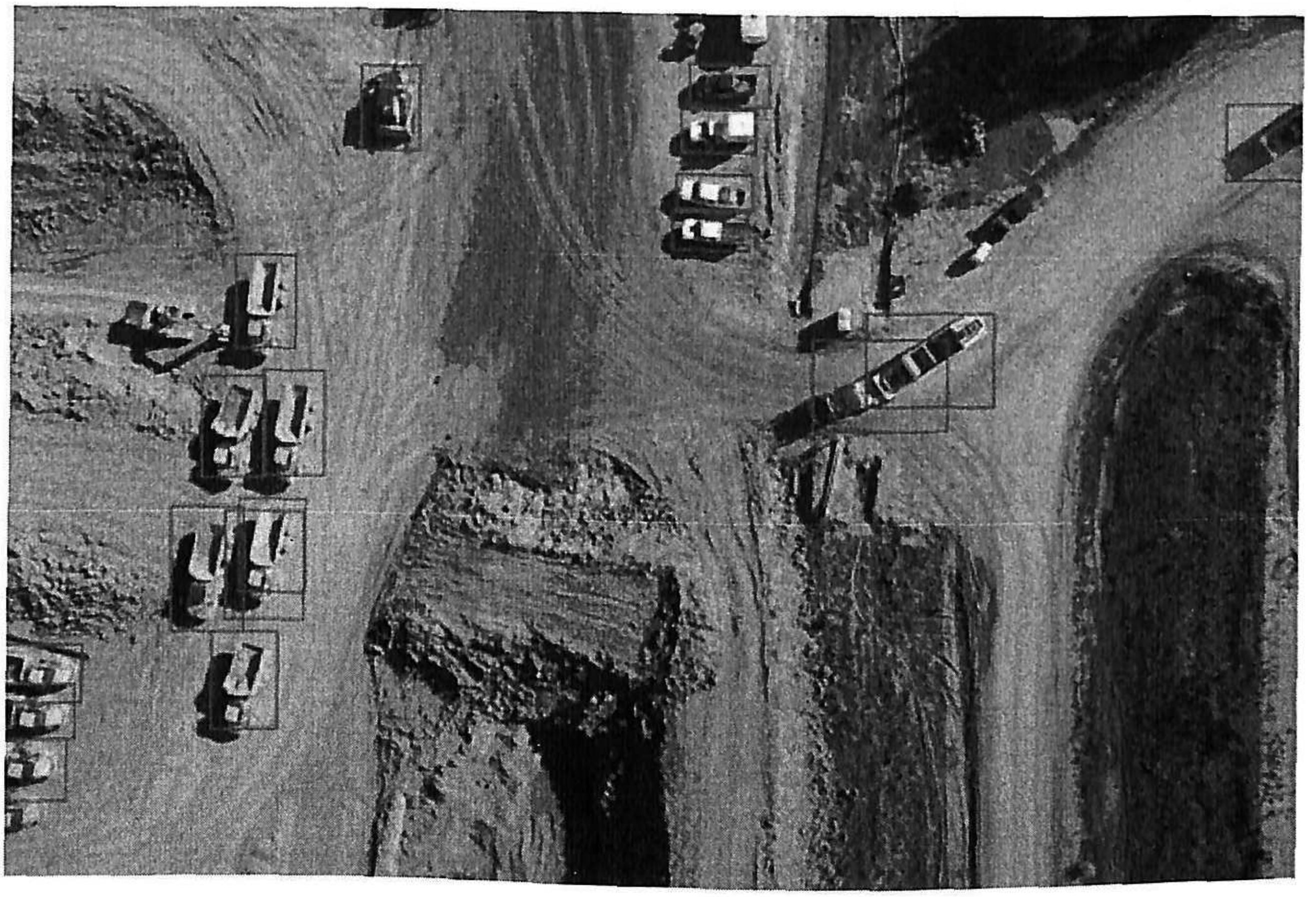

Figure 2.17: The result of the DetectNet neural network is shown, which is trained to detect vehicles in aerial images.

DIGITS now contain a new example neural network model architecture called DetectNet to help you get up and running with this new approach as quickly as feasible. Figure 2.17 is an example of DetectNet's output after being trained to recognize automobiles in aerial images.

As a class dimension builder, DetectNet employs a three-dimensional label format. As a result, the method can function with photos of various sizes and with any number of objects.

The image processor algorithm is shown in Figure 2.18 by the DetectNet training designation.

1. In the original image, a grid smaller than the size of the item to be specified is added.

2. The class of the item within each square in the grid, as well as the coordinates of the pixels of the rectangle's corners relative to the grid's central rectangle, are used to identify it.

3. If there are no things on the cage's site, it is tagged as "unattended." In the absence of an item, the "coverage" value is recorded in the input data format as

0; in the case of detection, it is 1. If multiple objects fall inside the grid window, the largest one is picked. If the pixel size is the same, the smallest coordinate along the Y-axis is chosen.[6]

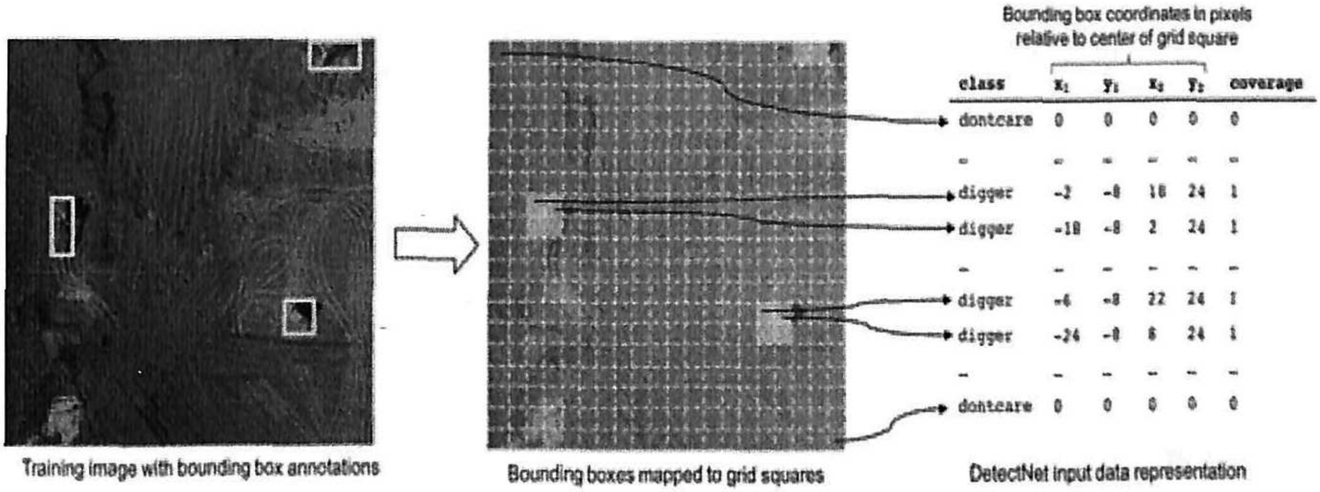


Figure 2.18: Input values.

Simply explained, DetectNet uses exercises to try to predict if an item is square or not. Figure 2.19 depicts the DetectNet learning neural network architecture.

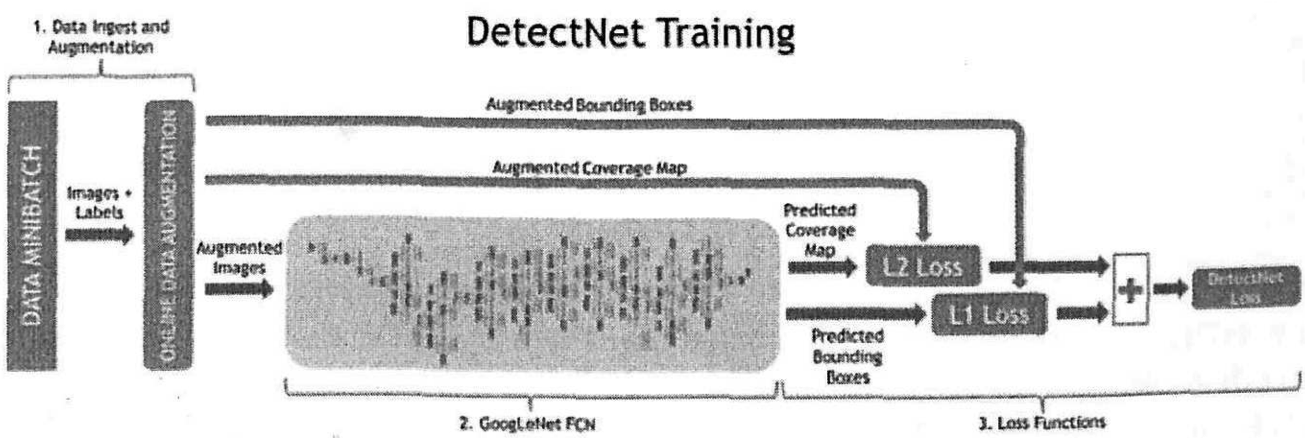


Figure 2.19: DetectNet structure for training.

The neural network has five layers and is built on the Caffe platform.

1. Put the picture and symbols in the appropriate places.

2. The Full Convulsion Network (FCN) is a network that accepts functions, anticipates exception classes, and links rectangles.

3. Cost functions also account for mistakes in issues such as anticipating an object's closure and the angles of connection between rectangles and squares in a grid.

4. The system's,estimated clustering rectangles

5. Calculation of the sample's simplified average accuracy, which defines the model's efficiency. You may achieve a pretty excellent result by using this technique to find numbers:

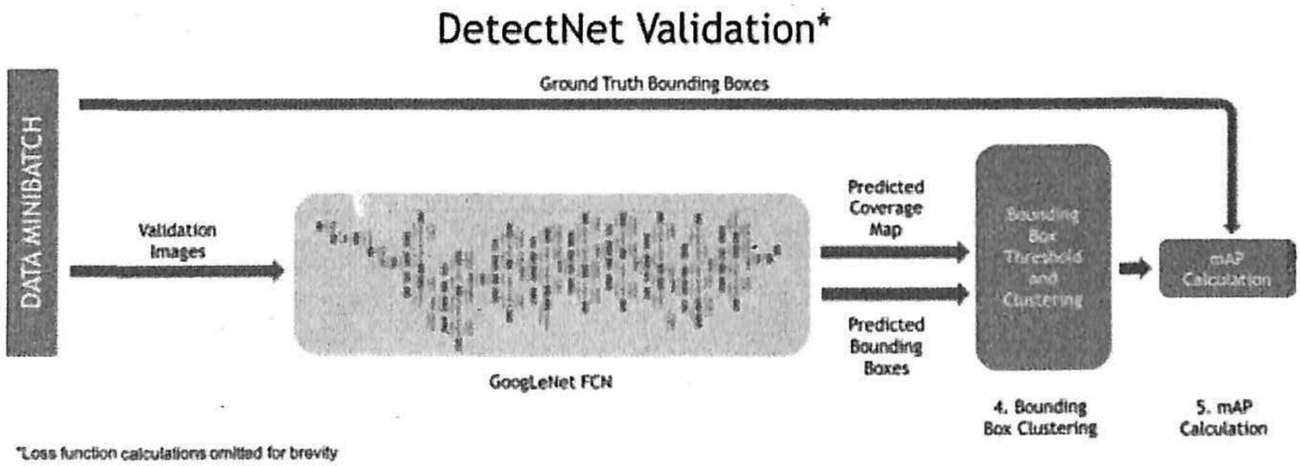


Figure 2.20: DetectNet structure for validation.

The stride in pixels in the *detectnetgroundtruthparam* layer can be used to describe the spacing of the grid squares in the training labels. Consider the following scenario:

```
detectnet_groundtruth_param {
    stride: 16
    scale_cvg: 0.4
    gridbox_type: GRIDBOX_MIN
    min_cvg_len: 20
    coverage_type: RECTANGULAR
    image_size_x: 1024
    image_size_y: 512
    obj_norm: true
    crop_bboxes: false
  }
```

You may additionally provide an image training patch size (image size x, image size y) in this layer. When these settings are configured, DetectNet will use a random crop of this size as input every time an image is sent into it during training. This might be beneficial if you have a lot of photographs with little things that you want to detect.

The *detectnetaugmentationparam* layer defines the parameters for online data augmentation. For example:

```
detectnet_augmentation_param {
    crop_prob: 1.0
    shift_x: 32
    shift_y: 32
    scale_prob: 0.4
    scale_min: 0.8
    scale_max: 1.2
    flip_prob: 0.5
    rotation_prob: 0.0
    max_rotate_degree: 5.0
    hue_rotation_prob: 0.8
    hue_rotation: 30.0
    desaturation_prob: 0.8
    desaturation_max: 0.8
  }
```

To successfully train a high sensitivity and accurate object detector using DetectNet, data augmentation is required. The detectnet augmentation param parameters specify how many random transformations, such as pixel shifts and flips, should be done to training pictures and labels each time they are ingested. Because the network never views the same training picture twice when utilizing online augmentation, it is significantly more resistant to overfitting and natural variation in the appearance of objects in test images than if we utilized a one-time static augmentation procedure.

## 2.4 Recognition methods

Currently, many different methods are used for recognition: neural networks, decision trees, support vector machines, k-nearest neighbor method, coverage algorithms, etc., in the construction of which training with a teacher is used when the output variable is set for each observation.

### 2.4.1 Decision tree

Decision trees were proposed by Quinlan as a solution to the problem of studying concepts. The method was supposed to present information within the framework of machine learning.

In the problem of building a decision tree, there are many attributes and many classes:

$$X = X_1 \times X_2 \times \ldots \times X_N$$
$$A = \{a_1, \ldots, a_d\}$$

where $N$ is the number of attributes, d is the number of classes.

By training sample:

$$D = ((x_1, c_1), (x_2, c_2), \dots, (x_M, c_M))$$

where, $x_i \in X$, $c_i \in A$, it is necessary to reconstruct the mapping acting from $X$ to $A$.

In the method of constructing trees, the classification algorithm is given in the form of a tree. In each node of the tree, one attribute is checked and, depending on the attribute value, a transition is made to the corresponding child node and is shown in Figure 2.21. Each leaf of the decision tree is assigned a class number. This class is the result of classification. The nodes of the tree are called decision nodes, and checking the attribute value in the node is a test procedure [14].

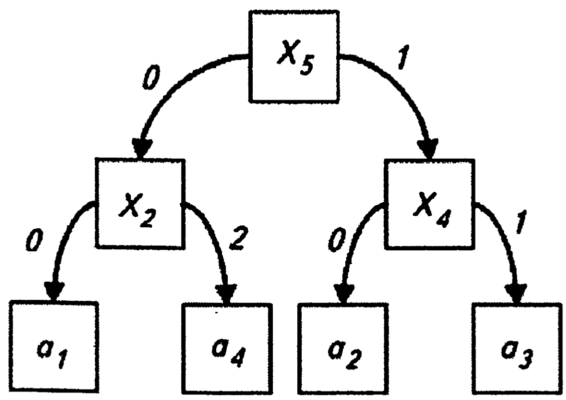


Figure 2.21: Example of a decision tree

### 2.4.2 k-nearest neighbor algorithm

The k-nearest neighbor algorithm is a metric method designed to classify objects. The basic principle of the k-nn method is that an object is assigned a class that is common among the neighbors of this element [6].

The algorithm needs to specify the number of clusters, the attributes of which are denoted as follows:

$$A = \{a_1, \dots, a_k\}$$

where $k$ is the number of classes, and $A$ is the set of classes.

The algorithm can be described in the following steps:

1) each of the k clusters are randomly assigned their reference images $\vec{y}_i$; these centers are usually the first $k$ images of the training sample:

$$\vec{y}_i = \vec{x}_i, i = 1, \dots, k$$

2) the selection elements are assigned to the class whose distance to the reference image is minimal:

$$A_i = \underset{a_j \in A}{\operatorname{argmin}}(s(\vec{x}_i, \vec{y}_i)), i = 1, \ldots, M,$$

3) cluster standards are recalculated, taking into account which images were assigned to them:

$$\vec{y}_i = \frac{1}{M_j} \sum_{(\forall i) A_i = a_j} \vec{x}_i$$

where $M_j$ is the number of images included in the class $a_j$;
4) steps 2 and 3 are repeated until the classes stop changing.

### 2.4.3 Support vector machine

The support vector machine technique (hence SVM, support vector machine) is a teacher-assisted learning algorithm that belongs to the linear classifiers family and is used for regression and classification applications. Because it constantly decreases the empirical classification error while increasing the gap, the support vector machine approach is a classifier method with a maximum gap.

The concept behind the approach is to convert the starting vectors into a high-dimensional space and then search for a separating hyper-plane. This hyperplane has the greatest gap in the space and two parallel hyperplanes on both sides. The hyperplane with the greatest distance between two parallel hyperplanes will be the one that separates them. The difference or distance between two parallel hyperplanes determines the average error of the technique results; the higher the distance, the smaller the error.[7]

## 2.5 Conclusion on the first chapter

Images captured in natural settings feature uneven lighting, flashes, and different distractions such as shadows, dirt, and so on. Threshold algorithms do not perform well in such photos. In the face of distortion, the build-up algorithms using numbered plates will fail. Unlike the Roberts and Sobel operators, the Canny algorithm is less vulnerable to picture noise and performs better than the others listed because it employs additional checks. The Viola-Jones approach, a clustering-based segmentation algorithm, is better suited for this purpose, since it has the capacity to recognize more than one region in the picture and uses simple classifiers while still being fast.

As a result, the Canny border algorithm and the Viola-Jones clustering-based segmentation approach were chosen as segmentation algorithms.

The k-nn algorithm and the support vector machine approach were chosen to address the character recognition problem.

K-nn has the following positive features:

1. simplicity of the software implementation of the algorithm;

2. the algorithm is resistant to abnormal outliers, since the characters not represented in the training sample will not fall into the number of k-nearest neighbors;

3. the result of the algorithm is easy to analyze;

4. the algorithm can be modified for a specific task, the use of the most appropriate function and metrics allows you to adjust the algorithm to a specific task.[8]

The SVM method has the following positive features:

1. resistant to abnormal emissions;

2. much faster among similar methods;

3. solves the problem of quadratic programming in a convex domain, resulting in a unique solution.

4. finds the dividing strip of the maximum width, which allows more accurate classification.[9]

To improve the quality of segmentation and recognition, it was decided to test combinations of additional filters, including grayscale image conversion, binarization, Gaussian smoothing, erosion, dilation, Laplacian-based border selection, histogram equalization and median filter.

# Chapter 3

# ALGORITHMS FOR SEGMENTATION AND RECOGNITION OF THE CAR LICENSE PLATE

The Viola-Jones algorithm and the contour analysis approach were investigated in order to tackle the challenge of segmenting an automobile license plate.

As a consequence of the research, a new contour analysis approach was presented, which produced superior segmentation accuracy results.

The methods of contour analysis and the method of support vectors were investigated in order to solve the problem of segmented symbol recognition.

Figure 3.1. depicts a block diagram of the suggested picture preprocessing.

## 3.1 Number plate segmentation algorithm

### 3.1.1 Contour analysis method

The contour analysis approach is a collection of transformation techniques and picture contour selection. The contour completely defines the image's shape and provides all of the information needed to continue working on it. Contour analysis has the benefit of providing chosen forms in an image without taking into account the picture's interior points, which decreases the quantity of data analyzed.

A contour is a dramatic shift in brightness levels across a large area. There are issues with choosing the image's contours:

- contour breaks in places where the brightness change is not too fast; - false contours in the presence of noise in the image; - blurring or noise leads to an increase in contour lines.

A contour is represented by a sequence of complex numbers in contour analysis.

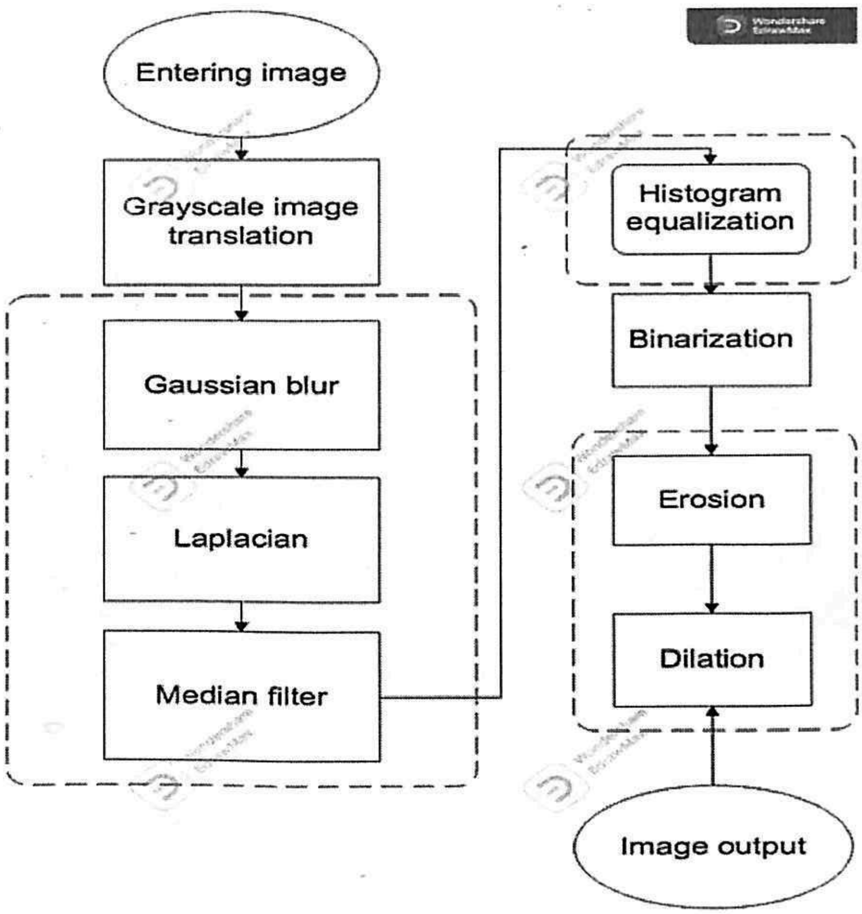


Figure 3.1: Block diagram with sequential application of filters.

The beginning point is a location on the contour that is fixed. The contour is then skipped, and each displacement vector is expressed as a complex integer $a + ib$, where $a$ represents the point's displacement along the $x$ axis and $b$ represents the point's displacement along the $y$ axis. [15] The displacement is measured in relation to the prior position (Fig. 3.2).

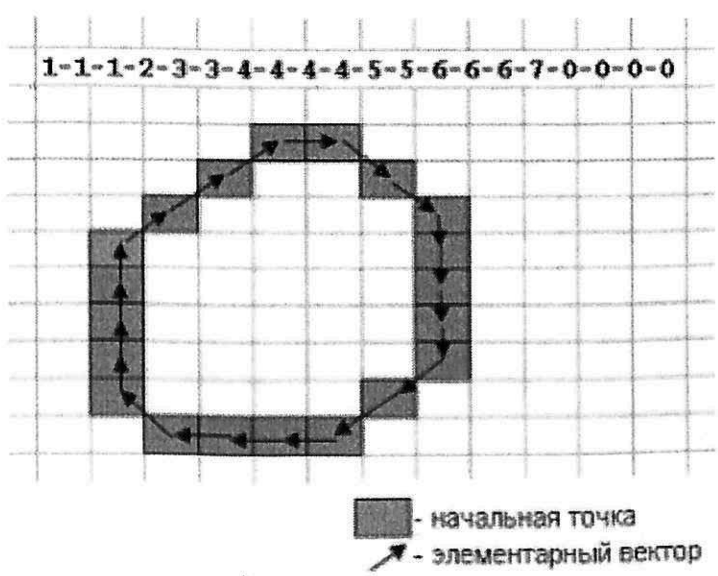


Figure 3.2: Encoding the contour.

To effectively search for contours, the original color image must first be converted to grayscale, and then methods for selecting the boundaries of objects in the image must be subjected. Of all the methods of boundary selection discussed in the first chapter, the Canny detector is the most accurate.

Figure 3.3: The original image used to check the Sobel, Canny, Roberts, and Previtt operators.

Let's look at the methods for selecting contours: Sobel, canny, Roberts, and Previtt operators.

**Sobel operator**

Simply put, the result of a point with a constant brightness of this differential operator is a zero vector, and if it belongs to the boundary of regions with different brightness, then the vector is directed to increase the brightness by crossing the boundary. Officially, its work can be described by the following formulas:

$$1.\, G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * A$$

$$2.\, G_y = \begin{bmatrix} 1 & 2 & -1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A$$

$$3.\, G = \sqrt{G_x^2 + G_y^2}$$

Where $A$ is initial image.


Figure 3.4: Filtering using the Sobel operator.

As can be seen from figure 3.4, the operator highlighted the boundaries. It should be noted that the vehicle's license plate and its signs are clearly visible on the territory of the vehicle.

**Canny edge detector**

In general, the work of this operator consists of five stages:

1. Smoothing - the image is blurred to eliminate noise.

$$B = \frac{1}{159}\begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} * A$$

2. Look for gradients - contours where the brightness difference is maximum.

$$G = \sqrt{G_x^2 + G_y^2}$$

$$\theta = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

is rounding and takes values 0, 45, 90, 135.

3. Maximum limit - only the local maximum is defined as the limit.

4. Double limit filtering - here the possible limits are set as the limit.

5. Search for an undefined region - all borders that are not related to previously defined borders are deleted.

Figure 3.5: Filtering using the Canny edge detector.

Looking at Figure 3.5, it is clear that this filter is superfluous for the problem of finding a number - it detects too many external contours. In the future, such information in the image will harm the algorithm.

**Roberts cross**

This operator is faster than the Sobel operator:

$$G_x = \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} * A$$

$$G_y = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} * A$$

$$G = \sqrt{G_x^2 + G_y^2}$$

$$\theta = \tan^{-1}(\frac{G_y}{G_x})$$

For this example, the frame of numbers and symbols are also clearly visible. If the car is dirty, then the calculation speed will not help to determine the frame of the license plate. After all, dusty cars with dirty numbers often drive on the roads. For the task of finding a number, this operator is suitable only if the clean number is a high-quality image.

Figure 3.6: Filtering using the Roberts cross.

**Pruett operator**
In general, this is the same as the Sobel operator, only the Sobel operator has twice its central weight:

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} * A$$

$$G_y = \begin{bmatrix} 1 & 1 & -1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} * A$$

$$G = \sqrt{G_x^2 + G_y^2}$$

As can be seen from figure 3.7, the boundaries of numbers are also well separated, as in the Sobel operator. When you take a low-quality picture of the frontal surface of the car at an angle, you will not see the same bright silhouettes in the license plate frame.

Figure 3.7: Filtering using the Pruett operator.

### Canny edge detector

J. Canny investigated the mathematical issue of generating an ideal filter based on the criteria of selection, localisation, and reduction of many edge clicks. This implies the detector must react to true borders while ignoring false ones, precisely locate the boundary line, and react to each boundary only once, preventing the impression of broad bands of brightness change as a collection of boundaries.

The algorithm includes:

1. smoothing — reduces the sensitivity of the algorithm to noise. The first derivative of the Gaussian is used for smoothing;
2. gradient search — searches for the maximum gradient value to highlight the boundaries;
3. suppression of non—maximums - marks only local maximums as boundaries;
4. double threshold filtering — marks potential boundaries with thresholds;
5. tracing the area of ambiguity — suppresses all edges that are not connected with certain boundaries to establish the final boundaries (are established by suppressing all edges) [16].

The selection of each symbol is as follows:

a. Using the Canny boundary detector, the boundaries of all objects are highlighted in the image.

b. Based on the contour analysis method, we obtain the contours of objects in the form of a set of points with coordinates.

c. Having a set of points for each contour, the smallest enclosing rectangle containing the area inside the contour is determined.

d. The rectangle is represented as the coordinate of the upper-left corner of the contour and its width. Based on this information, we select the contour of each rectangle on the initial image and save the inner area as a new image.

Figure 3.8 shows an example of license plate segmentation based on the contour analysis method.


Figure 3.8: Segmentation of number characters.

### 3.1.2 Viola–Jones object detection framework

Paul Viola and Michael Jones presented the Viola-Jones approach in 2001 as the first object identification system that offers real-time detection of competing items. The technique is based on sliding window technology: this is a tiny frame that goes across the image and discovers the item with the assistance of a trained cascade of weak classifiers. This approach is frequently employed in a variety of computer vision applications.

There are two algorithms in the method: a learning algorithm and a recognition algorithm. Figure 3.9 depicts a generalized recognition system in the Viola-Jones method.

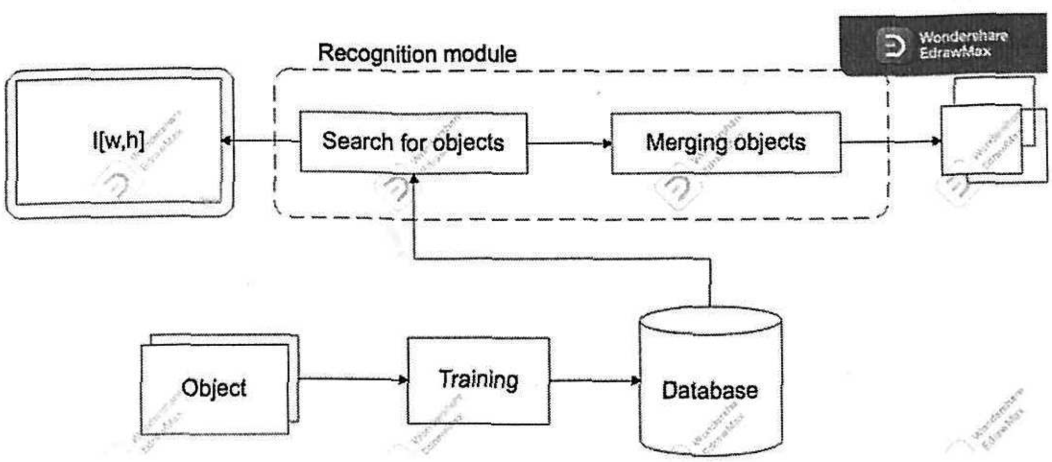


Figure 3.9: Generalized recognition scheme of the Viola-Jones method.

The generalized scheme of the algorithm looks like this:

1. The algorithm generates a database consisting of features.

2. Next, the recognition algorithm searches for objects at different image scales using the created database.

3. The Viola-Jones algorithm outputs the entire set of unconnected objects found at different scales.

4. The next task is to decide which of the found objects are really present in the frame, and which are duplicates.

As features for the recognition algorithm, the authors proposed Haar-like features based on Haar-like wavelets.

With the help of Haar-like signs, you can set a point value for the brightness differences along the axis $X$ and $Y$. The value of Haar-like features can be calculated by the formula:

$$F = X - Y$$

where $X$ is the sum of the brightness values of the points covered by the light part of the feature; $Y$ is the sum of the brightness values of the points covered by the dark part of the feature.

It can be seen that if we count the sum of the intensity values for each sign, this will require significant computing resources. Viola and Jones proposed using an integral representation of the image.

The classifier is based on the boosting algorithm. Boosting selects the most suitable features for the desired object in this part of the image.

Boosting is a collection of techniques for improving the accuracy of analytical models in general. A "powerful" classifier is a model that permits minimal classification mistakes. A "weakness" on the other hand, a classifier is a model that cannot consistently separate classes or make correct predictions. A model like this produces a lot of inaccuracies. As a result, boosting refers to the "strengthening" of "weak" models and is a step-by-step technique for putting together a machine learning algorithm composition. The algorithm attempts to compensate for the inadequacies of the preceding algorithms' composition.

At each iteration of the algorithm, a "weak" classifier of the type:

$$h_{j(z)} = \begin{cases} 1, & if\ p_j f_j(z) < p_j \theta_j \\ 0, & otherwise \end{cases}$$

where $p_j$ is the direction of the inequality sign, $\theta_j$ is the threshold value, is the calculated value of the attribute, $z$ is the 24×24-pixel image window.

The classifier obtained by formula has a minimum error relative to the current values of the weights involved in the training procedure to determine the error.

A trained classifier, supplied in xml format, is used to look for an item in a digital picture. Haar-like characteristics are used to build the classifier.

A cascade of classifiers is constructed on this foundation to determine whether or not an object in the image is identified. The difference between the attribute's value and the threshold determines whether the required item is present in the window [17].

### 3.1.3 Modification of the contour analysis method

The contour analysis method showed to be superior during the tests. For future research, this method was chosen. It was determined to develop this method in order to increase segmentation accuracy, as this method does not produce a good enough result when segmenting a collection of integers with distortion. There are a few drawbacks that have been addressed with the inclusion of extra filters:

1. Excessive color information – eliminated by transforming the image in grayscale;
2. Small noises - eliminated by subsequent Gaussian blur;
3. The boundaries were distinguished by the Laplacian operator;
4. Point noise is eliminated using a median filter;
5. To increase the contrast of the image, the equalization of histograms was used;
6. Background. Binarization was used to get rid of it;
7. The noise in the binary image was removed by erosion;
8. Using dilation fills the voids. Recognition methods.

### 3.1.4 k-nearest neighbor algorithm

The method uses the distance function to minimize the global measure, which is the total standard deviation of pictures from their class centers.

One of the disadvantages of the algorithm is the need to specify the number or size of clusters. The number and size of clusters carry approximately the same amount of important information. For example, the following algorithm requires setting the cluster size:

1. form one cluster (k=1) from the first image of the training sample and put $\vec{y}_i = \vec{x}_i$;
2. select the vector $\vec{x}_i$ not considered and determine the minimum distance by the formula:

$$s' = \min s(\vec{x}_i, \vec{y}_i)$$

3. repeat step 2 for the entire training sample.

This method assigns each image to a certain class once and is sufficient for efficient calculation, but its result depends on the order of the sequence of input images.

Figure 3.10 shows the results of applying the method.


Figure 3.10: Generalized recognition scheme of the Viola-Jones method.

### 3.1.5 Supported vector machine

The supported vector machine is an algorithm belonging to the family of linear classifiers. The algorithm is a learning algorithm with a teacher, for solving the problem of classification and regression analysis. The method can also be considered as a special case of Tikhonov regularization. The supported vector machine method is a classifier method with a maximum gap, since it continuously reduces the empirical classification error and increases the gap.

The method works by translating the original vectors into a high-dimensional space and looking for a separating hyperplane. This hyperplane has the greatest gap in the space and two parallel hyperplanes on both sides. The hyperplane with the greatest distance between two parallel hyperplanes will be the separating hyperplane. The difference or distance between two parallel hyperplanes determines the average error of the technique results; the higher the distance, the smaller the error [5].

In machine learning algorithms, there is a need to classify data. Each data object is represented as a vector in space with a sequence of p numbers. Only one of the two classes applies to each item. A hyperplane of size $p-1$ is required to split objects. To do this, the gap is increased, resulting in a more confident categorization. To do so, we calculate the hyperplane distance from which the distance to the nearest item is the greatest. When such a hyperplane is discovered, it is dubbed the optimum separating hyperplane, and the linear classifier matching to it is dubbed the optimal separating classifier [7].

**Formal description of the task**

It is assumed that the objects have the form:

$$\{(x_1, c_1), (x_2, c_2), \dots, (x_n, c_n)\}$$

where $c_1$ is a value (1 or -1) that depends on whether the object $x_1$ belongs to the class. Each $x_1$ is a two—dimensional vector with values of [0,1] or [-1,1].

If the objects are not normalized, then an object with large deviations from the average values of the object's coordinates will have a huge impact on the classifier.

It is assumed that a class has already been set for each object. To do this, we construct a separating hyperplane that has the form:

$$w * x - b = 0$$

where $w$ - a vector perpendicular to the separating hyperplane. A reference vector is an object located close to parallel hyperplanes.

The parameter $b/(||w||)$ is equal modulo the distance from the hyperplane to the origin coordinates. If parameter $b$ is zero, the hyperplane passes through the origin, which limits the solution [18].

We are interested in optimal separation, support vectors and hyperplanes parallel to the optimal and closest to the support vectors of the two classes. Parallel hyperplanes can be described by the following equations [16].

$$w * x - b = 1$$
$$w * x - b = -1$$

If the training sample is linearly separable, then we can choose hyperplanes in such a way that no object of the training sample lies between them and then maximize the distance between the hyperplanes. The width of the strip between them is equal to $2/w$. The task is to minimize $w$. To exclude all objects from the strip, you need to make sure that for all $i$:

$$w * x - b \geq 1, c_i = 1$$
$$w * x - b \leq 1, c_i = -1$$

Also can be written as:

$$c_i(w * x_i - b) \geq 1, 1 \leq i \leq n$$

**The case of linear separability**

To solve the problem, it is necessary to construct an optimal separating hyperplane. This reduces to minimizing $w$, under the conditio. Such a task is called quadratic optimization:

$$\begin{cases} \|w\|^2 \to min \\ c_i(w * x_i - b) \geq 1, 1 \leq i \leq n \end{cases}$$

By the Karush–Kuhn–Tucker conditions, this problem is equivalent to the dual task of finding the saddle point of the Lagrange function:

$$\begin{cases} L(w, b; \lambda) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \lambda_i c_i((w * x_i) - b - 1) \to min_{w,b} min_\lambda \\ \lambda_i \geq 0, 1 \leq i \leq n \end{cases}$$

where $\lambda = (\lambda_1, \ldots, \lambda_n)$ is the vector of dual variables.

We reduce this problem to an equivalent quadratic programming problem containing only dual variables:

$$\begin{cases} L(\lambda) = -\sum_{i=1}^{n} \lambda_i + \sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j c_i c_j (x_i * x_j) \to min_\lambda \\ \lambda_i \geq 0, \ 1 \leq i \leq n \\ \sum_{j=1}^{n} \lambda_i c_i = 0 \end{cases}$$

When solving problem, we can find $w$ and b by the following formula:

$$w = \sum_{i=1}^{n} \lambda_i c_i x_i,$$
$$b = w * \lambda_i c_i x_i, \lambda_i > 0$$

As a result, the classification algorithm can be written as:

$$a(x) = sign\left(\sum_{i=1}^{n} \lambda_i c_i x_i * x - b\right)$$

At the same time, the summation does not go over the entire sample, but only over the reference vectors for which $\lambda_i \neq 0$[16].

**The case of linear inseparability**

In order for the algorithm to work if the classes are linearly inseparable, let's let it make mistakes on the training sample. We introduce a set of additional variables $\xi_i \geq 0$, that characterize the magnitude of the error on the objects $x_i, 1 \leq i \leq n$. Let's take (21) as a starting point, soften the constraints of inequality, and also introduce a penalty for the total error into the minimized functional:

$$\begin{cases} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \to min_{w,b,\xi_i} \\ c_i(w * x_i - b) \geq \xi_i, 1 \leq i \leq n \\ \xi_i \geq 0, 1 \leq i \leq n \end{cases}$$

The coefficient $C$ is a parameter of the method setting that allows you to regulate the ratio between maximizing the width of the dividing strip and minimizing the total error [1] [10].

Similarly, by the Karush–Kuhn–Tucker conditions, we reduce the problem to finding the saddle point of the Lagrange function:

$$L(w, b, \xi; \lambda, \eta) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \lambda_i (c_i(w * x_i) - 1) - \sum_{i=1}^{n} \xi_i(\lambda_i + \eta_i - C) \to min_{w,b} min_{\lambda,\eta}$$

$$\xi_i \geq 0, \lambda_i \geq 0, \eta_i \geq 0, 1 \leq i \leq n$$

$$\begin{cases} \lambda_i = 0 \\ (c_i(w * x_i - b) = 1 - \xi_i \end{cases}, 1 \leq i \leq n$$

$$\begin{cases} \eta_i = 0 \\ \xi_i = 0 \end{cases}, 1 \leq i \leq n$$

By analogy, we reduce this problem to the equivalent:

$$\begin{cases} -L(\lambda) = -\sum_{i=1}^{n} \lambda_i + \sum_{i=1}^{n}\sum_{j=1}^{n} \lambda_i \lambda_j c_i c_j (x_i * x_j) \to min_\lambda \\ \lambda_i \geq 0, \ 1 \leq i \leq n \\ \sum_{j=1}^{n} \lambda_i c_i = 0 \end{cases}$$

In practice, it is this problem that is solved to build a support vector machine, and not (27), since it is generally not possible to guarantee the linear separability of points into two classes. This variant of the algorithm is called a soft gap algorithm, whereas in the linearly separable case, they talk about a hard gap.

The formula (29) is preserved for the classification algorithm, with the only difference that now not only the reference objects, but also the offending objects have nonzero $\lambda_i$ values. In a certain sense, this is a disadvantage, since noise emissions often turn out to be violations, and the decisive rule based on them, in fact, relies on noise.

The constant $C$ is usually chosen according to the criterion of sliding control. This is a time-consuming method, since the task has to be solved anew with each value of $C$.

If there is reason to believe that the sample is almost linearly separable, and only outlier objects are classified incorrectly, then you can apply outlier filtering. First, the problem is solved at some C, and a small proportion of objects with the largest error value $\xi_i$ is removed from the sample. After that, the problem is solved anew using a truncated sample. It may be necessary to do several such iterations until the remaining objects are linearly separable [19].

The disadvantage of the supported vector machine method is that in the case of linear inseparability of classes, there is no general approach to automatic kernel selection [19].

## 3.2 Description of the training. Description of training and test samples

### 3.2.1 Training sampling of the k-nn algorithm

Let the set of objects $X$ set a function of distance $\delta : X \times X \to [0, \infty)$. There is a target dependency $y^* : X \to Y$, the value of which is known only on the training sample feature

$$X^\ell = (x_i, y_i)_{i=1}^\ell, y_i = y^*(x_i)$$

Lots of classes $Y$, of course. It is required to construct a classification algorithm $a : X \to Y$, approximating the target dependence $y^*(x)$ on the entire set $X$.

The algorithm assigns the classified object $u \in Xl$ to the class to which the nearest training object belongs:

$$w(i,u) = [i = 1]$$
$$a(u; X^{\ell}) = y_u(1)$$

For the training sample, an image with Latin letters and Arabic numerals was used, a total of 22 characters (see Figure 3.11).

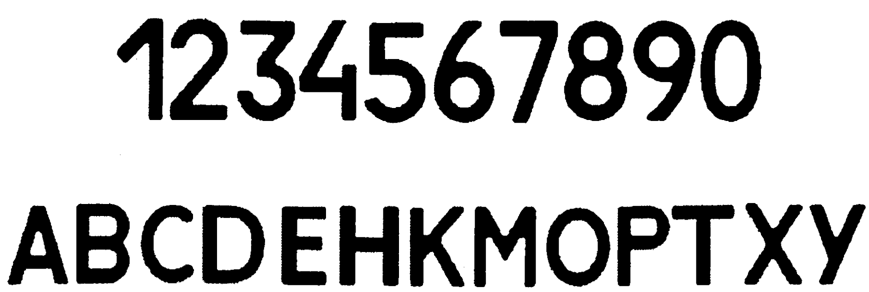


Figure 3.11: Training image.

According to the program, we generate a training sample in two files: classifications.xml – ASCII of each character, images.xml – actual data for each character. The program finds letters and classifies them.[11]

### 3.2.2 Viola-Jones teaching and learning method

The learning algorithm of the Viola-Jones method is carried out in the following way: A set of positive and negative images $N$ with their attributes $(x^i, y^i)$ is fed to the input. If the image $i$ is the desired object, then $y^i = 1$, if not, then $y^i = -1$;

1. Initialization: assigning to each image $i$ weight $w_1^i = 1/N$;
2. For each function $f_j$ with $j = 1, \ldots, M$:

a. Reshaping the weights so that they add up to one.

b. Applying function (16) to each image in the training set, then find the optimal threshold and polarity $\theta_j$, $s_j$, which minimizes the weighted classification error. That is:

$$\theta_j, s_j = \arg min \sum_{i=1}^{N} w_j^i \varepsilon_j^i$$

$$\varepsilon_j^i = \begin{cases} 0, if\ y^1 = h_j(x^i, \theta_j, s_j) \\ 1, \quad otherwise \end{cases}$$

c. Assignment the weight of $a_j$ to $h_j$ which is inversely proportional to the error coefficient. Thus, the best classifiers are considered more important.

d. The weight of the next iteration, that is, $w_(j+1)^i$ will decrease for images that have been correctly classified.

Setting the last classifier to (19)

$$h(x) = sign\left(\sum_{j=1}^{M} a_j h_j(x)\right)$$

Learning the Viola-Jones algorithm is learning the algorithm with a teacher. The learning algorithm works with grayscale images. The algorithm uses 16 masks, where a weak classifier is trained for each configuration, giving the smallest error. A weak classifier is added to the training base.

A training set of 440-character images was used for the training. The images were from a positive and negative set, that is, images containing the desired characters $l$ and images not containing the characters $m$:

- $l$ = 1542 images of a positive set
- $m$ = 896 images of the passive set

Weak classifier to guess the presence of an object 50% of the time. Using the AdaBoost learning process, a strong classifier is created consisting of T weak classifiers.

As a result of the training, we received the cascade database.xml of 16 weak classifiers.

### 3.2.3 Training and training sampling for the SVM algorithm

The support vector machine's training procedure consists of the following steps:

1. At the input, there is a training sample: $(x_1, c_1), (x_2, c_2), \ldots, (x_n, c_n)$;

2. The method of support vectors is to construct a classifying function F in the form:

$$F(x) = sign(w, x) + b$$

where $(w, b)$ is the scalar product, $w$ is the normal vector to the separating hyperplane, $b$ is an auxiliary parameter.

3. Those objects for which $F(x) = 1$ fall into one class, and objects with $F(x) = -1$ fall into another. The choice of such a function is not accidental: any hyperplane can be given in the form $(w, x) + b = 0$ for some $w$ and $b$.

4. Such $w$ and $b$ are selected that maximize the distance to each class. It can be calculated that this distance is equal to $1/w$ and is calculated by the formula (25).

For the training sample, detected images of license plates were used, a total of 22 characters of 10 images (Fig. 3.12). According to the program, we generate a training sample. The SVM learning algorithm is presented on page 23.

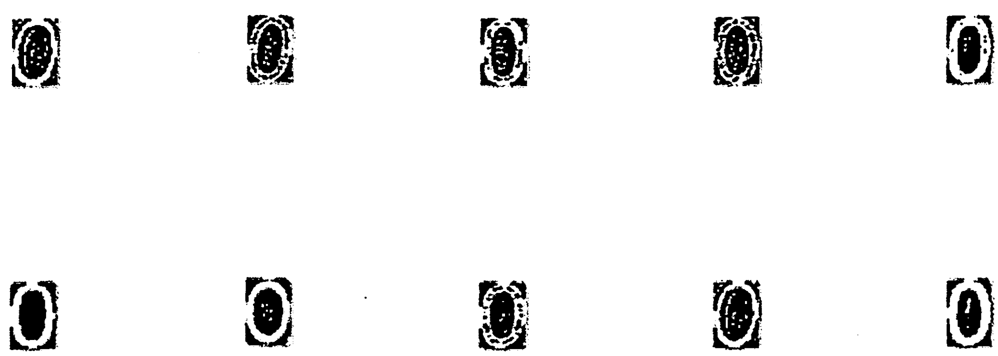

Figure 3.12: Training sample for the digit 0.

### 3.2.4 Description of the test sample

For the test sample, 200 photos of license plates containing only 1,620 characters were taken and tested. The numbers were detected manually and were divided into 3 groups: normal – numbers where the slope is less than 30° numbers and letters are clearly visible (100 number plates); at an angle – numbers where the angle of inclination is more than 30° (50 number plates); with the presence of distortion – numbers that have a defect, dirty, or numbers that are shot without focus or blurred during shooting (50 number plates).

## 3.3 General segmentation and recognition algorithm

The block diagram of the general segmentation and recognition algorithm is shown in Figure 18. As a result of the program execution, an image with segmented characters on the image and a recognized number were obtained.

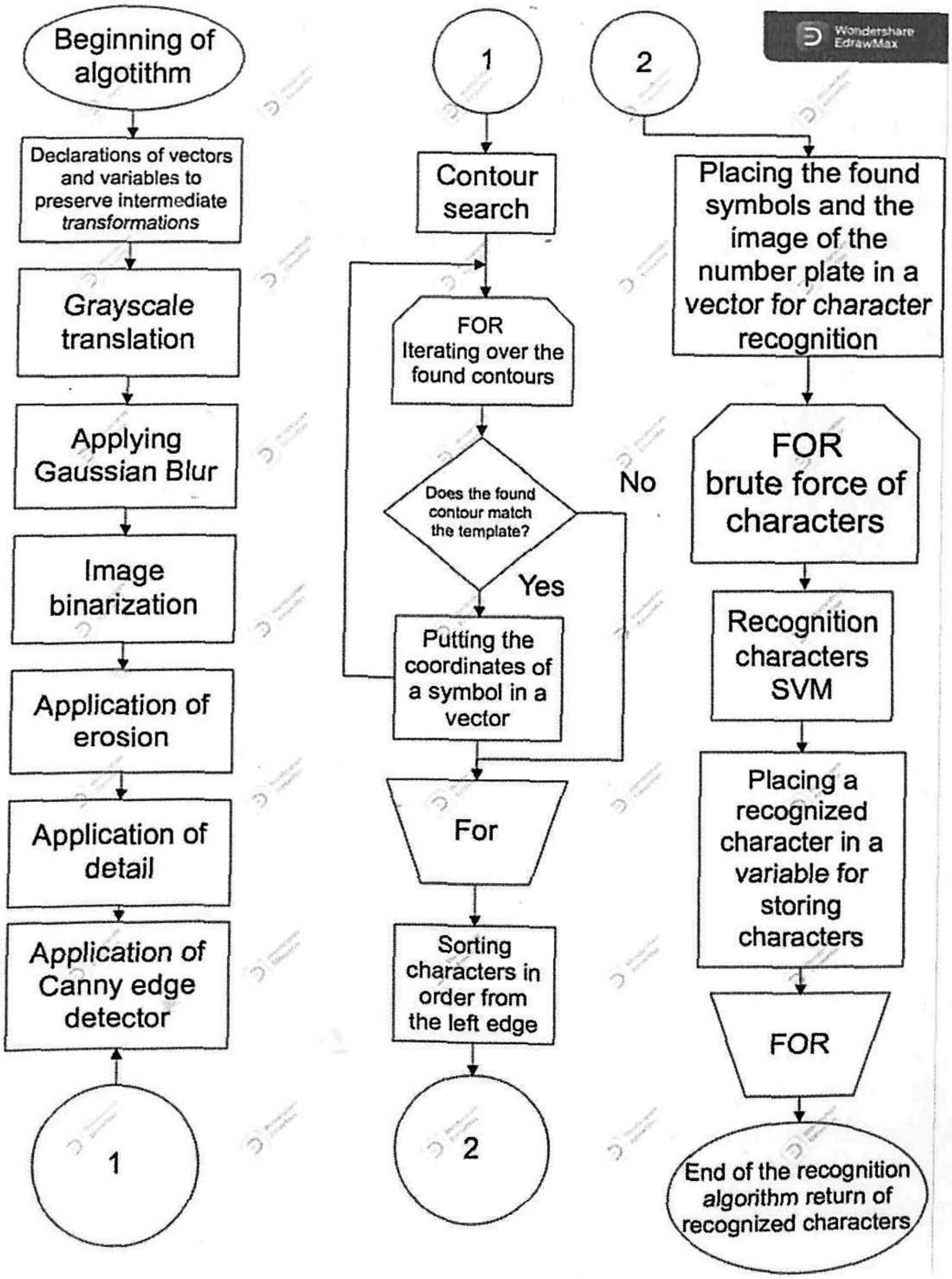


Figure 3.13: General segmentation and recognition algorithm.

## 3.4 Software implementation

### 3.4.1 Choosing the development environment and language

The OpenCV library and the C++ programming language were selected to implement the algorithm.

OpenCV supports fundamental image analysis and processing (computer vision) algorithms, including machine learning-based algorithms such as drawing, filtering, transformation, and pattern recognition. Open-Source Computer Vision

Library is its name. The libraries are free and open source, and are distributed under the BSD license [21].

The programming environment was chosen to be Microsoft Visual Studio and Qt Creator.

The CMake tool was developed to automate the process of constructing a program from source code. Kitware is the major developer of CMake, which is a free and open-source tool.

The SMake tool's concept of operation is as follows: a file is computed from the source code directory CMakeLists.txt with the project description, and the tool's output creates project files for one of the numerous construction end systems.

The presence of the make utility and the script interpreter of the compiled CMake tool, as well as a C++ compiler, are all prerequisites for creating CMake. CMake source codes, on the other hand, employ solely the language, standard, and library's capabilities.[12]

### 3.4.2 Program Description

On a computer, the algorithms were developed, trained, and tested using the following settings: 8 GB RAM; Ati Radeon HD 3470 512 MB graphics card; Intel Core i7-4510U CPU with clock frequency of 2.00 GHz and 2.60 GHz.

Figure 34 depicts the program's user interface. One primary window with three buttons makes up the user interface:

1. Select an image. The normal dialog window for opening the Select File to Open file appears when you click the Select Image button. In the main window, the selected license plate image is displayed;

2. Recognition. The segmentation and identification procedure begins when you click the button. The segmented image and recognition result are shown as a result.

3. SVM training. The SVM learning/retraining process begins when you click the button.


Figure 3.14: Program interface.

The application is made up of two classes: MainWindow and ContourWithData, and it makes use of the Qt and OpenCV libraries. Methods and class fields are shown in Tables 1-4.

The following classes make up the program being developed:

1. The MainWindow class implements the program's graphical interface and triggers the call of the main methods.

2. ContourWithData is a class that analyzes the contour that has been discovered.

Table 2.1 Methods and fields of the MainWindow class

| Methods of the class | |
|---|---|
| ContourWithData() | Constructor |
| checkIfContourIsValid() | Check if the contour is correct in width |
| checkIfContourIsValid() | Check if the contour is correct in width and height |
| sortByBoundingRectXPosition() | Sorting by the position of the rectangle bounding the contour |
| **Class Fields** | |
| rect | Fields |
| ptContour | The vector of points that organizes the contour |
| fltArea | Contour area |

Table 3.2: Methods and fields of the ContourWithData class

| Methods of the class | |
|---|---|
| MainWindow() | Constructor |
| ~MainWindow() | Destructor |
| on_pb_run_clicked() | Event handler for pressing the Run button to recognize the license plate |
| on_pb_open_clicked() | Handler for the Open button click event. The function opens a folder for selecting the number plate |
| on_pb_traning_clicked() | Handler for the button click event Training, for SVM training |
| IplImage2QImage() | Conversion between IplImage to QImage |
| QImage2IplImage() | Conversion between QImage to IplImage |
| Mat2QImage() | Converting Mat to QImage |
| characterRecognition() | Segmented character recognition |
| count_pixel() | Counting the number of pixels of a given intensity |
| calculate_feature() | Calculation of the ratio of the number of pixels of a certain intensity in the window to the total number of pixels of the corresponding intensity in the entire image |
| Class Fields | |
| contoursWithData | Contours with data |
| matBegin | Matrix of the original image |
| matWork | Matrix for working with |
| matCopy | A copy for findContours to work with |
| ptContours | Monetary, units |
| v4iHierarchy | Declare contour hierarchies |

Table 3.1: Methods and fields of the MainWindow class

| | |
|---|---|
| QImage | The class provides a hardware-independent representation of the image that allows direct access to pixel data and can be used as a drawing device |
| QPixmap | This class is an off-screen image that can be used as a drawing device |

Table 3.3: Qt Library Classes

| | |
|---|---|
| Mat | Data type |
| cvtColor | Conversion to Grayscale |
| equalizeHist | Equalization of histograms |
| Laplacian | Laplacian |
| adaptiveThreshold | Image Blur |
| medianBlur | Median filter |
| erode | Performing morphological erosion operation |
| dilate | Performing morphological dilation operation |
| findContours | Contour Search |
| SVM | Is a discriminatory classifier formally defined by a separating hyperplane |

Table 3.4: OpenCV Library

## 3.5 Conclusion on the second chapter

In this chapter, picture training and test samples were gathered in order to train segmentation and recognition algorithms as well as conduct numerical experiments. Segmentation and recognition algorithms such as the contour analysis method, Viola-Jones method, support vector machine method, k-nn method, and image preparation approaches were examined using the OpenCV and Qt libraries. These approaches were put in place and taught on.

# Chapter 4

# Segmentation of text characters in found license plates

The multi-stage algorithm for determining vehicle registration plates, discussed in chapter 3, can be used in an integrated text character recognition system. To do this, it is necessary to supplement the detector with segmentation and character classification algorithms.

The following primary tasks are provided in this chapter in order to develop a fully effective car license plate recognition system in digital photos based on the suggested detection algorithm:

- development and analysis of symbolic segmentation algorithms for a successfully found number, taking into account the information component of the digital image;

- search for the working parameters of the proposed algorithms using the analysis of visual and quantitative results of character segmentation evaluation;

- modification and analysis of classification algorithms applied to the problem of car license plate recognition.

Analysis of the literature on noise reduction in digital images shows that digital images can contain noise of various kinds. This means the need for expanded use of a number of digital image processing methods in the number recognition system.[13]

An example of such tasks may include: RGB image color filtering, filtering of "unedited" images, preventing blurring of object borders and increasing sharpness in the image, etc.

The solution to these problems can be found in the relevant scientific and technical literature.

## 4.1 Segmentation of text characters

Today, the segmentation of objects in the image should be carried out taking into account its information component. General methods are functionally limited, as they have significant geometric limitations.

It is recommended to take into account the information in the image as a basis for highlighting the objects of the textual content of car registration plates and the final differentiation of characters. To determine the information value of a pixel, we introduce the concept of "cost function".

This function is a conditional value that describes how important a given image is in the current image.

The general scheme of the proposed algorithm is as follows:

1. Calculation of the energy function for each pixel of the image. There are many variants of the energy function. When calculating it, the most important elements in the image will be the outlines and contours of objects, which are shown in Figure 4.1.

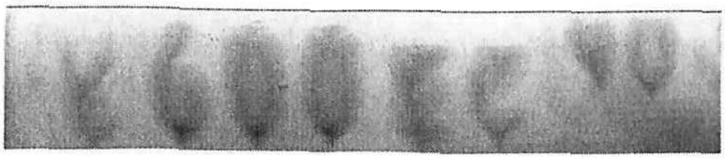

Figure 4.1: Power function for test video car license plates.

Taking into account the energy function, it is necessary to solve the problem of symbol distribution. There may be several ways to solve this problem. For example, you can set a condition for maximum energy savings and divide the minimum energy into pixels. This approach destroys the structure of closed objects and greatly deforms labels.

2. In order to avoid damage to the image, it is necessary to observe the conditions for adding a pixel, according to which the division into segments is performed. All lines are inserted into the image-a set of 8 pixels drawn from top to bottom on the image. The method of assembling such rows is shown in Figure 4.2.

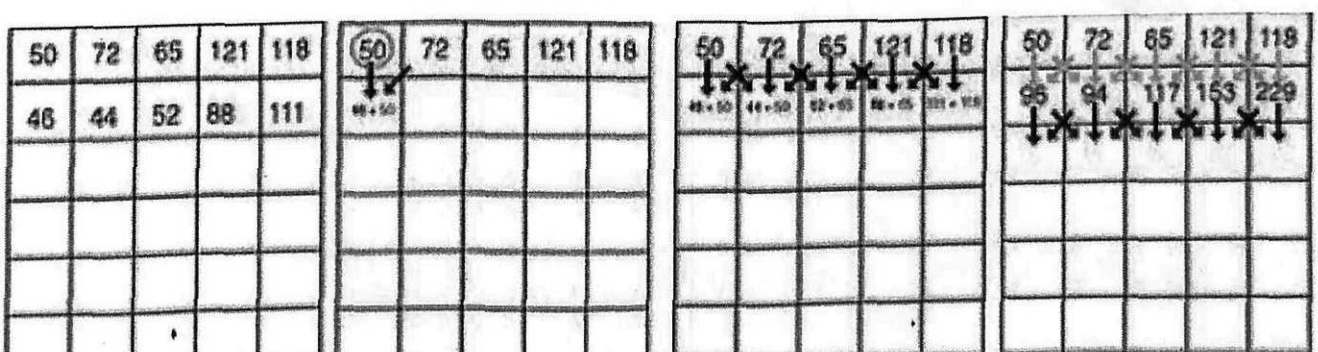


Figure 4.2: The principle of calculating the additional weight coefficient.

The rules for the development of the annexed areas are drawn up as follows.

- The original image is obtained by calculating the energy function (Fig 4.1.).
- for a pixel $(n+1)$ row, an additional weight is calculated - the smallest value of three neighboring pixels in the $n-th$ row.
- This is the general idea of the operation.

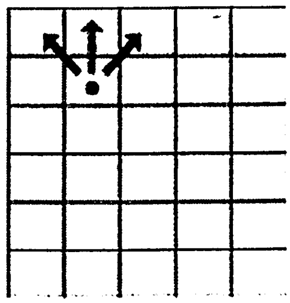

Figure 4.3: Determination of pixels with a neighboring effect for the analyzed element of the digital image.

- The operation of calculating the additional weight $(n+1)$ - for each pixel of the row (Fig 4.2.). These additional weights are added to the current intensity of the analyzed row pixel $(n+1)$.
- The operation is performed for all image paths. Since there is a function $E(I)$, we calculate the cost of all such rows:

$$E(s) = E(I_s) = \sum_{i=1}^{n} e(I(s_i))$$

The lines with the minimum cost are selected, which will be the dividing lines between the characters:

$$s^* = minE(s) = min\sum_{i=1}^{n} e(I(s_i))$$

The formation of the dividing line of characters, as shown in Figure 4.4, occurs in the reverse order of accumulation in a repeated form.

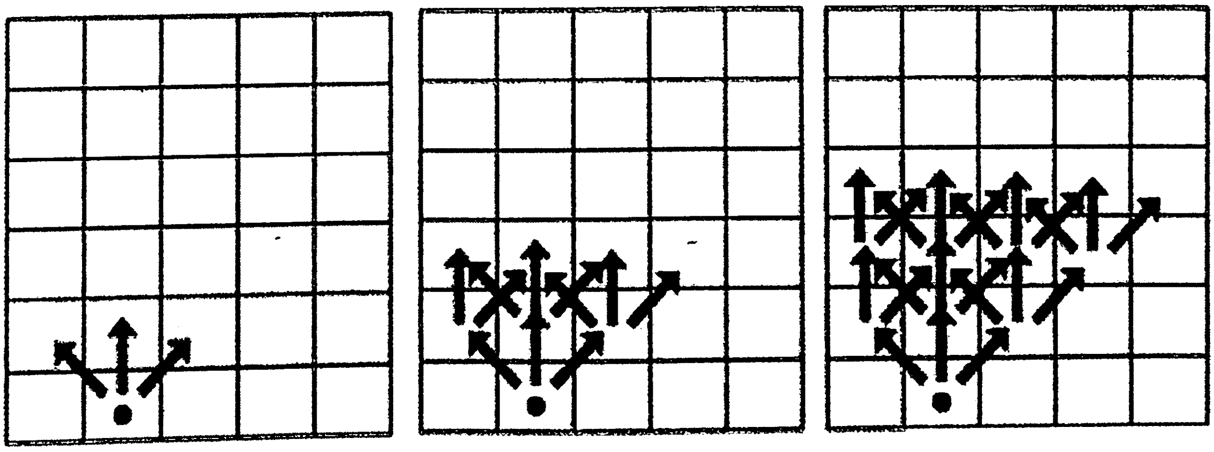

Figure 4.4: Possible routes that make up the passage.

Vertical sequences consist of pixels in which the following conditions are true:

- the total energy function of all pixels in the circuit is minimal;
- the chain crosses the image from the bottom up;
- the circuit is arranged in such a way that each row contains only one pixel, and adjacent pixels must be connected by sides and corners.

To evaluate the quality of the algorithm under consideration, the ROC analysis method described above was used. Within the framework of this methodology, the dependence of the number of correctly defined lines in the image on the number of false lines is studied.

It is known that in practice it is impossible to increase two doses at the same time: the sensitivity and specificity of the test. A compromise can be reached by adjusting the parameters of the segmentation algorithm, in particular the cost function. So, by changing parameters A and b in the cost function, the ratio of sensitivity and specificity can be changed. In addition, you can find the values of the parameters that are most suitable for solving the problem. There are a huge number of criteria, one of which is a condition for the balance between sensitivity and specificity: $Sensitivity \approx Specificity$.

Thus, during testing, the dependence of specifications and sensitivity on parameters A and b was calculated. The figure shows graphs of these dependencies. It can be seen that these surfaces have common points that intersect along certain curved generators. The specified intersection areas correspond to the values of the optimal parameters A and b according to the criterion of equilibrium between the sensitivity and specificity of the algorithm.

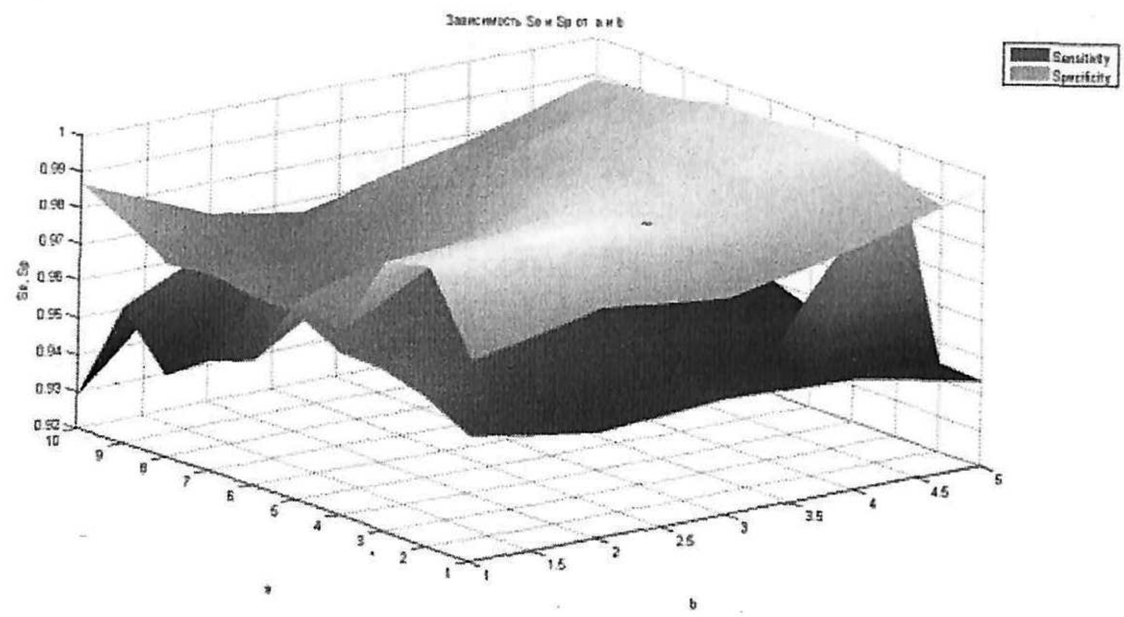

Figure 4.5: Dependence of sensitivity and originality on cost parameters

## 4.2 Additional correction of the segmentation algorithm

When identifying signs of license plates, it is necessary to separate the segments of signs from each other, but this is an insufficient condition for successful classification of signs. Each segment must be additionally cut automatically, while preserving only the Information symbol.

In the case of symbolic segmentation, the task of finding optimal parameters is to fully match the reference marking area and the automatic one. Since the linear dimensions of these objects are very small and the symbol occupies an area equal to about 20 * 25 pixels, all relative values, such as accuracy and recall, vary significantly with a small deviation of the shapes from each other.

At the same time, the cases of increasing the area of each automatically found segment (FP) or truncating incorrect characters (FN) are not equal to each other. In the first case, the noise level increases during classification, and in the second case, the information component of the symbol does not recover. Therefore, the task of finding the optimal parameters of the cost function should be expanded in comparison with the above.

A natural solution to this problem is to use the character classifier as an evaluation criterion. In this case, the classification algorithm is considered given and does not change during the experiment. Based on (3.1), the following matrix of conditions $T$ is constructed for each value of parameters $A$ and $B$:

$$\mathrm{T} = \mathrm{N}\left\{\begin{array}{l} k\left\{\begin{array}{lll} \mathrm{TP}_{11} & \mathrm{FP}_{11} & \mathrm{FN}_{11} \\ \mathrm{TP}_{12} & \mathrm{FP}_{12} & \mathrm{FN}_{12} \\ \mathrm{TP}_{13} & \mathrm{FP}_{13} & \mathrm{FN}_{13} \\ & \cdots & \\ \mathrm{TP}_{1k} & \mathrm{FP}_{1k} & \mathrm{FN}_{1k} \end{array}\right. \\ \qquad\qquad \cdots \\ \qquad\qquad \cdots \\ \qquad\qquad \cdots \\ k\left\{\begin{array}{lll} \mathrm{TP}_{1j} & \mathrm{FP}_{j1} & \mathrm{FN}_{j1} \\ \mathrm{TP}_{j2} & \mathrm{FP}_{j2} & \mathrm{FN}_{j2} \\ \mathrm{TP}_{j3} & \mathrm{FP}_{j2} & \mathrm{FN}_{j3} \\ & \cdots & \\ \mathrm{TP}_{jk} & \mathrm{FP}_{jk} & \mathrm{FN}_{jk} \end{array}\right. \end{array}\right. .$$

This matrix is created for $N$ test images, each of which has a $K$ symbol. Simultaneously with this matrix, a probability matrix of the correct classification of the symbol $P$ is constructed:

$$\mathrm{P} = \mathrm{N}\left\{\begin{array}{l} k\left\{\begin{array}{l} \mathrm{P}_{11} \\ \mathrm{P}_{12} \\ \mathrm{P}_{13} \\ \cdots \\ \mathrm{P}_{1k} \end{array}\right. \\ \quad \cdots \\ \quad \cdots \\ \quad \cdots \\ k\left\{\begin{array}{l} \mathrm{P}_{j1} \\ \mathrm{P}_{j2} \\ \mathrm{P}_{j3} \\ \cdots \\ \mathrm{P}_{jk} \end{array}\right. \end{array}\right. .$$

In it, each peak position contains a binary number that indicates whether the classifier solution of this segment coincides with the character value stored in the database. Since the compilation of such matrices for all images will be an intensive calculation task, we get a certain dependence of the recognition probability on the values TP, FP, FN.

In this case, the estimation of the specific classification of each symbol from the parameters of the symbol detector is indicated by the following dependence:

$$\alpha TP + \beta FP + \gamma FN = \tilde{P}$$

The coefficients $\alpha$,$\beta$ and $\gamma$ can be derived from the conditions for segments in the Matrix, and the matrix form has the corresponding condition:

$$T \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = P$$

The solution to this equation will be the following matrix:

$$\begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \frac{T^T T}{T^T P}$$

Given that for correct recognition of the license plate, an accurate classification of each character is required at the same time, for all images in the database, such an assessment takes the following form:

$$S(a,b) = \sum_{j=1}^{N} \left( \prod_{i=1}^{k} \tilde{P}_k \right) = \sum_{j=1}^{N} ( \prod_{i=1}^{k} (\alpha TP_{ik} + \beta FP_{ik} + \gamma FN_{ik}))$$

The resulting formula is a function of target optimization, since it clarifies the correct recognition of the entire number along with the parameters selected in the segmentation algorithm.

The maximum of this function $[a_0, b_0]$ reaches the point, so:

$$[a_0, b_0] = argmax(S(a,b))$$

Thus, it is possible to obtain the values of the parameters of the value function a and b, which are considered from the point of view of recognizing the actual number for the algorithm. In this case, the most favorable ratio between errors of the first and second types is achieved, and the algorithm works effectively. Thanks to this, it will be possible to accurately draw the dividing lines between adjacent characters in the digital image of the car license plate.

# Chapter 5

# Classification of text characters

Usually, the process of recognizing car license plates consists of three main stages: segmentation of license plates of individual signs on the marking and their subsequent classification. When classifying text car registration signs, the processes of plate detection and segmentation are considered successful.

All existing methods of doing this use pre-created templates or are based on training. In the first case, templates are created for each possible character and placed in the database. During recognition, a comparison is made at the system input with the newly received sign, all of which is present in the database. In this case, all characters must be brought to the same size. However, this method makes significant errors even with a slight change in the color of objects or light transmission, so some features of objects (perimeter, area, intensity projection) are distinguished, which leads to an increase in the reliability of recognition.[14]

Among the learning-based recognition methods, it is worth highlighting multilayer neural networks, decision trees, and committee methods based on the simultaneous use of several classifiers.

The use of methods of this class provides higher recognition accuracy compared to the template, but at the same time increases the calculation time. To reduce it, it is necessary to highlight information characteristic features. This paper discusses character-based learning methods for Character Recognition: logistic regression and Boltzmann's limited machines.[15]

## 5.1 Multiclass single classification

The usual logistic regression algorithm solves binary classification problems. In general, the number of recognized classes can be quite large, and the usual logistic regression cannot be used. Therefore, to solve such problems, a multiclass classifier is developed. One of the most popular schemes for implementing such a classifier is the "All Against All" scheme. In this case, n classifiers are trained, each of which separates one class from the others (figure 5.1).

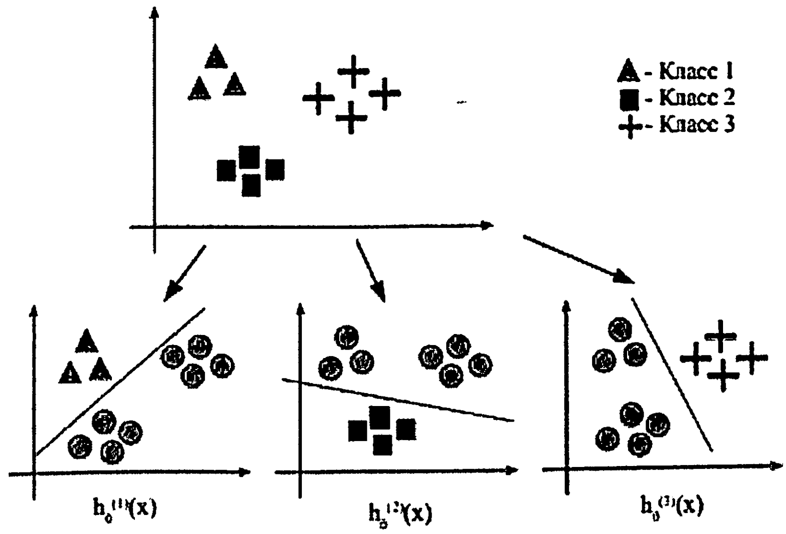


Figure 5.1: The principle of "against all".

## 5.2 The problem of reducing the size of the opportunity space

In research and experimental work, there are many cases when the total number of signs registered in each of the studied objects is quite large (in our case, four hundred signs are used). However, the available multidimensional data must undergo statistical processing. In this case, new auxiliary features can be selected from among the primary features, or according to some rule, a set of primary characteristics can be defined, such as their linear combination. When forming a new feature system, different requirements are imposed, such as informality, mutual correlation, minimal distortion of the geometric structure of the original data set, and so on. Depending on the option of formalization of these requirements, we come to an algorithm for reducing one or another dimension. There are at least three main types that allow you to switch from a large number of primary attributes to a much smaller number of more informative variables:

- duplication of information that occurs when there is a strong link between symptoms;
- low variability of symptoms;
- ability to accumulate, weighty set of features.

## 5.3 Analysis of the main components of solving the data classification problem

It is enough to project the dimensions on the plane of the first two main components to see a clear division of the studied set of dimensions into classes.[16]

It is often impossible to capture this division directly in the initial multidimensional space. In the interpretation of general multidimensional observation and the formulation of the classification problem using the main component method $x_i = ( x_i^1, \ldots, x_t^p)^T$, it is necessary to pay attention to the possible duality. Indeed, if in the matrix of observation:

$$(x_1, x_2, \ldots, x_n) = \begin{pmatrix} x_1^{(1)} & x_2^{(1)} & \cdots & x_n^{(1)} \\ x_1^{(2)} & x_2^{(2)} & \ddots & x_n^{(2)} \\ \vdots & \vdots & & \vdots \\ x_1^{(p)} & x_1^{(p)} & \cdots & x_2^{(p)} \end{pmatrix}$$

consider columns as observations, then the objects themselves will be classified objects, and if there are rows, then the signs will become classified objects. If the goals of the first classification are clear, then the goal of the second classification is to find similar characteristics. If some functions are combined into groups, the size of the initial factor space can be reduced. Another important point is that the signs are closely related to each other, for example, they can be proportional. Therefore, before using the method, it is necessary to normalize the initial control data. Also, for convenience, you can focus your attention on the center, which will ensure the convenience of presenting information.

# Chapter 6

# Test results

The detection and recognition algorithms were put to the test. 200 license plates with a total of 1,620 characters were subjected to testing.

The results are given in the table, Appendix A. The value of the columns:

- Total – the number of characters on the license plate;
- Segmented area – the total number of segments found;
- Correctly segmented – the number of correctly segmented areas;
- Recognition – the number of correctly recognized elements.[17]

## 6.1 Testing of image preprocessing methods

Grayscale image conversion, Gaussian blur and binarization were chosen as the main filters (MF).[18] Testing of image preprocessing methods was carried out on three groups of numbered plates:

- Normal – numbers where the slope is less than 30° numbers and letters are clearly visible;
- At an angle – numbers where the angle of inclination is greater than 30°;
- With the presence of distortion – defocused and blurred numbers.

Table 6.1 shows the results of evaluating image preprocessing methods for 408 characters of number plates with distortion.

| Image pre-processing methods | General segment | Correct segment | SVM | Accuracy segment | Recognition accuracy |
|---|---|---|---|---|---|
| Main filter (MF) | 320 | 305 | 177 | 0,97 | 0,58 |
| MF + erosion + dilation (erosion element = 2;dilation element = 3) | 278 | 277 | 238 | 0,99 | 0,85 |
| MF + erosion + dilation (erosion element = 1;dilation element = 1) | 406 | 402 | 384 | 0,98 | 0,96 |
| MF + Equivalent histogram + erosion + dilation | 299 | 278 | 192 | 0,92 | 0,69 |
| MF + Equivalent histogram + median + erosion + dilation | 209 | 189 | 132 | 0,90 | 0,69 |
| MF + Laplacian+ erosion + dilation | 297 | 285 | 236 | 0,95 | 0,82 |

Table 6.1: Image preprocessing methods

The following filter combinations produced the greatest results among the methods listed: MF + erosion + dilation and MF + Laplacian + erosion + dilation. Filtering with the Laplacian filter leaves filthy spots on the license plate, which the segmentation algorithm interprets as a symbol, lowering the identification accuracy. The method began to perform more properly once the value of the erosion and dilation components were changed.

When evaluating image preprocessing methods, the following filter combination produced the best results: basic filters + erosion + dilation, which produced the best results in terms of segmentation and recognition accuracy.

## 6.2 Testing segmentation methods

The results of segmentation based on the contour analysis method are shown in the table of Appendix A. Based on the results obtained (Appendix A):

- Total number of characters: x = 1620;
- Number of segmented areas: s = 1613;
- The number of correctly segmented: y = 1600.

$$\textit{Average segmentation accuracy} = \frac{y}{x} * 100\% = \frac{1600}{1620} * 100\% = 99\%.$$

$$\textit{Average segmentation error} = \left|1 - \frac{y}{s}\right| = \left|1 - \frac{1600}{1613}\right| = 0.01$$

Table 6.2 shows a report on individual groups of numbers.

| Group | Segmentation accuracy of the contour analysis method, % |
|---|---|
| Normal | 99 |
| At an angle | 99 |
| With the presence | 98 |

Table 6.2: Calculating the accuracy of groups

The results of the Viola-Jones method-based segmentation algorithm are provided in Appendix A table. Based on the results obtained (Appendix A):

- Total number of characters: x = 1620
- Number of segmented areas: s = 2247
- Number of correctly segmented: y = 1348

$$\textit{Average segmentation accuracy} = \frac{y}{x} * 100\% = \frac{1297}{1608} * 100\% = 80\%$$

$$\textit{Average segmentation error} = \left|1 - \frac{y}{s}\right| = \left|1 - \frac{1348}{2247}\right| = 0.4$$

Table 6.3 shows a report on individual groups of numbers.

| Group | Segmentation accuracy Viola-Jones method, % |
|---|---|
| Normal | 90 |
| At an angle | 80 |
| With the presence | 73 |

Table 6.3: Calculating the accuracy of groups

Accuracy is determined by the percentage of correctly recognized characters relative to the total number of characters. According to the two developed segmentation algorithms, the following results were obtained (Table 6.4).

| | Contour detection algorithm | The Viola Jones method |
|---|---|---|
| Number of characters | 1620 | 1620 |
| Number of segmentation area | 1613 | 2247 |
| Correct segmentation area | 1610 | 1348 |
| Segmentation accuracy | 99 | 80 |
| Segmentation fault | 0.01 | 0.4 |

Table 6.4: Calculating the accuracy of groups

The results of the tests revealed that the two algorithms had a significant difference in segmentation accuracy and error. Using the Viola-Jones method, over 40% of segmented regions are not symbols. As a segmentation algorithm, a contour-finding algorithm was employed.

## 6.3 Testing recognition methods

The segmented areas were recognized by the K-nn and SVM algorithm. The test results are shown on the table in Appendix A.

Based on the results of the K-nn recognition algorithm, the following results were obtained (Appendix A).

- The number of correctly segmented: $y = 1600$
- Number of recognized by the K-nn algorithm: $knn = 1429$
- Number of recognized by the SVM algorithm: $svm = 1580$

$$Average\ recognition\ accuracy\ of\ the\ algorithm\ k-nn = \frac{knn}{y} * 100$$

$$= \frac{1425}{1601} * 100\% = 89\%$$

$$Average\ recognition\ accuracy\ for\ the\ algorithm\ SVM = \frac{svm}{y} * 100\%$$

$$= \frac{1561}{1601} * 100\% = 98.7\%$$

Tables 6.5 and 6.6 provide a report on individual groups of numbers for the K-nn algorithm and the SVM support vector machine.

| Group | Average recognition accuracy of the knn algorithm, % |
|---|---|
| Normal | 93 |
| At an angle | 86 |
| With the presence | 81 |

Table 6.5: Calculation of recognition accuracy by groups using the k-nn algorithm

| Group | Average recognition accuracy of the knn algorithm, % |
|---|---|
| Normal | 99 |
| At an angle | 99 |
| With the presence | 98 |

Table 6.6: Calculation of group accuracy for the SVM support vector machine

| Algarithm | Processing time, sec | t, sec | Recognition percentage, % |
|---|---|---|---|
| K-nn | 5,11 | 1.50 | 89 |
| SVM | 8,56 | 0.88 | 98.7 |

Table 6.7: Calculation of group accuracy for the SVM support vector machine

The proportion of properly detected characters relative to the total amount of characters determines accuracy, and t is the one-dimensional plate recognition time.

The recognition results revealed that the SVM algorithm has a lower percentage of false positives (97%) than the K-nn approach (89%). Because there are numbers with distortion that the method cannot handle, a set of number plates with the presence of distortion of numbers is detected with a 96 percent accuracy.

Algorithms learn rapidly, and because the difference between them is just 3.45 fractions of a second, only one training is required. The knn algorithm takes 5 minutes to recognize 200 number plates, whereas the SVM takes 3 minutes.

## 6.4 Comparison of the obtained results with existing algorithms

There are several ways for identifying license plates in use today. The goal of this work was to develop a registration mark recognition algorithm with a high probability of success. Modules like: are the functional analogues of the produced software.

1. The Template Matching algorithm, which employs comparison with a reference as a recognition algorithm and dilation as a filter for picture preprocessing, is described in the paper [22]. The experiment included 96 photos of numbered plates, with 89 of them successfully segmented.

2. The MVLP algorithm, which comprises the Black Top-Hat method for symbol segmentation and a neural network trained using the error back propagation method for character recognition, is presented in the paper [23]. The experiment was carried out on 257 photographs of numbered plates, with 234 accurately segmented and 228 correctly identified.

3. The findings of the LPR algorithm are presented in the paper [24], which employed FPV (Floating Peak and Valleys) as a segmentation algorithm and the k nearest neighbors method as a recognition algorithm. The experiment was conducted on 439 license plate photos.

The results of the experiment on the software products listed above are shown in Table 6.8.

| Methods used | Segmentation speed, sec | Segmentation accuracy, % |
|---|---|---|
| Template Matching | 0.3 | 92.7 |
| VLPR | 0.4-0.6 | 91 |
| LPR | 2 | 97.5 |
| The developed algorithm | 0.4 | 99 |

Table 6.8: Comparison of software segmentation methods

| Methods used | Segmentation speed, sec | Segmentation accuracy, % |
|---|---|---|
| Template Matching | 0.6 | 90.6 |
| VLPR | 0.5-1 | 88.7 |
| LPR | 0.6 | 94.2 |
| The developed algorithm | 0.8 | 98.7 |

Table 6.9: Comparison of software product recognition methods

| Methods used | Overall segmentation accuracy, % |
|---|---|
| Template Matching | 78.3 |
| VLPR | 88.7 |
| LPR | 79.8 |
| The developed algorithm | 98.7 |

Table 6.10: Comparison of software product recognition methods

According to the tables 6.8-6.10, it can be concluded that the developed algorithms do not use the methods that were implemented within the framework of this high recognition accuracy. The algorithm proposed in this paper has achieved a high recognition accuracy of 98.7%.

## 6.5 Conclusion by chapter

The results of numerical experiments on the developed algorithms were acquired in this chapter. According to the results of the experiment, image conversion to grayscale, Gaussian blur, binarization erosion, dilation, histogram equalization, contour analysis algorithm as a segmentation algorithm, and SVM support vector method as a recognition algorithm were chosen for the recognition of car license plates. The whole algorithm's accuracy is 97.5 percent (filter bank + contour analysis technique + SVM method).

$$\textit{Average accuracy of the algorithm: contour analysis} + SVM = \frac{svm}{s} * 100\%$$

$$= \frac{1580}{1620} * 100\% = 97.5\%$$

# Chapter 7

# Conclusion

The challenge of segmenting and recognizing automobile license plates was set and implemented in this research. Two segmentation techniques, the contour finding algorithm and the Viola-Jones method, and two-character recognition algorithms, k-nn and SVM, were investigated throughout the research. Based on the findings, a combination segmentation and recognition method were proposed, consisting of a contour finding method and an SVM method with additional filters to increase segmentation and recognition accuracy.

The contour finding technique, the Viola-Jones method, the Haar cascade training, the k-nn algorithm, and the support vector approach are all examined in depth. Experiments on 200 photos were conducted using the created algorithms. The numbers were manually recognized and sorted into three groups: normal – numbers with a slope less than 30°, digits and letters plainly visible; at an angle – numbers with a slope more than 30°; with distortion – defocused and blurred numerals.

Grayscale image conversion, binarization, Gaussian smoothing, erosion, dilation, Laplacian-based boundary selection, histogram equalization, and the median filter were all used to increase the quality of segmentation and identification. During the tests, the combination of basic filters + erosion + dilation produced the greatest results, with an average segmentation accuracy of 99 percent and an average recognition accuracy of 86 percent.

Based on the results of this work, the following main conclusions can be drawn:

1. After doing research and comparing two segmentation methods, the Viola-Jones method and contour analysis, it was discovered that the contour analysis method has a 92.6 percent higher segmentation accuracy. Using the Viola-Jones technique, over half of the segmented areas are not symbols.

2. As a result of the analysis, the contour analysis method was chosen for further improvement.

3. It was feasible to improve segmentation accuracy to 99 percent by altering the contour analysis method using a combination of extra filters.

4. The approaches of support vectors and K-nn were researched when trying to solve the problem of character recognition. The support vector machine had

the greatest recognition results, with 96 percent accuracy compared to 89 percent accuracy for K-nn.

5. In terms of recognition speed, the reference vector method works faster by 0.62 fractions of a second on one license plate: Intel inside Core i7- 4510U CPU 2.00 GHz 2.60 GHz; 8 Gb RAM.

6. With an average accuracy of 97.5 percent, the created program can detect automobile license plates of various categories: clean, with distortion, and at an angle of more than 30 degrees.

# Appendix A

# The results of testing the algorithm for contour analysis.

Column value:

- S total – the number of characters on the license plate;
- segmented area – total number of segments found:
- CO KA – segmented area of contour analysis;
- CO V-D is a segmented area of the Viola-Jones method;
- KA – the number of correctly segmented areas by contour analysis;
- C-D – the number of correctly segmented areas by the Viola Jones method;
- Knn - the number of correctly recognized elements by the knn method;
- SVM – the number of correctly recognized elements by the reference vector method.

| № | Нормальные | | | | | | | Под углом | | | | | | | С наличием искажения | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | S | CO KA | KA | CO В-Д | В-Д | knn | SVM | S | CO KA | KA | CO В-Д | В-Д | knn | SVM | S | CO KA | KA | CO В-Д | В-Д | knn | SVM |
| 1 | 8 | 8 | 8 | 13 | 8 | 8 | 8 | 8 | 7 | 7 | 12 | 1 | 7 | 7 | 8 | 8 | 8 | 13 | 7 | 6 | 8 |
| 2 | 8 | 8 | 8 | 10 | 7 | 8 | 8 | 8 | 8 | 8 | 8 | 6 | 8 | 8 | 8 | 8 | 8 | 13 | 7 | 7 | 8 |
| 3 | 8 | 8 | 8 | 12 | 7 | 7 | 8 | 8 | 8 | 8 | 8 | 7 | 7 | 8 | 8 | 8 | 8 | 15 | 6 | 8 | 8 |
| 4 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 10 | 5 | 8 | 8 | 8 | 8 | 8 | 15 | 7 | 7 | 8 |
| 5 | 8 | 8 | 8 | 13 | 8 | 8 | 8 | 8 | 8 | 8 | 10 | 6 | 5 | 8 | 8 | 8 | 8 | 9 | 6 | 8 | 8 |
| 6 | 8 | 8 | 8 | 9 | 7 | 7 | 8 | 8 | 8 | 8 | 9 | 8 | 7 | 8 | 8 | 8 | 8 | 10 | 6 | 5 | 8 |
| 7 | 8 | 7 | 7 | 11 | 7 | 7 | 7 | 8 | 8 | 8 | 13 | 3 | 6 | 8 | 8 | 9 | 8 | 15 | 5 | 8 | 8 |
| 8 | 8 | 8 | 8 | 9 | 8 | 8 | 8 | 8 | 8 | 7 | 12 | 6 | 7 | 7 | 8 | 8 | 8 | 13 | 6 | 7 | 8 |
| 9 | 8 | 8 | 8 | 9 | 7 | 6 | 8 | 8 | 8 | 8 | 12 | 7 | 8 | 8 | 9 | 9 | 9 | 11 | 7 | 8 | 8 |
| 10 | 8 | 8 | 7 | 13 | 7 | 6 | 7 | 8 | 8 | 8 | 12 | 4 | 8 | 8 | 8 | 8 | 8 | 10 | 4 | 6 | 7 |
| 11 | 8 | 8 | 8 | 11 | 8 | 8 | 8 | 8 | 8 | 8 | 9 | 8 | 7 | 8 | 8 | 8 | 8 | 9 | 3 | 7 | 8 |

| | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 12 | 8 | 8 | 8 | 9 | 5 | 8 | 8 | 8 | 8 | 8 | 13 | 8 | 6 | 8 | 8 | 8 | 8 | 15 | 6 | 8 | 8 |
| 13 | 9 | 7 | 7 | 14 | 7 | 7 | 7 | 8 | 7 | 7 | 13 | 5 | 7 | 7 | 8 | 8 | 8 | 13 | 5 | 6 | 7 |
| 14 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 9 | 5 | 8 | 8 | 9 | 8 | 8 | 10 | 7 | 8 | 8 |
| 15 | 9 | 9 | 9 | 14 | 7 | 6 | 9 | 8 | 8 | 8 | 12 | 6 | 7 | 8 | 8 | 8 | 8 | 10 | 6 | 8 | 8 |
| 16 | 8 | 8 | 8 | 8 | 6 | 6 | 8 | 8 | 8 | 8 | 7 | 4 | 6 | 8 | 8 | 8 | 8 | 11 | 8 | 7 | 7 |
| 17 | 8 | 8 | 8 | 8 | 5 | 8 | 8 | 8 | 8 | 8 | 10 | 7 | 8 | 8 | 8 | 8 | 8 | 14 | 5 | 4 | 8 |
| 18 | 8 | 8 | 8 | 13 | 6 | 8 | 8 | 8 | 8 | 8 | 11 | 8 | 5 | 8 | 8 | 8 | 7 | 14 | 6 | 5 | 7 |
| 19 | 8 | 8 | 8 | 12 | 6 | 8 | 8 | 8 | 8 | 8 | 13 | 8 | 8 | 8 | 8 | 8 | 8 | 14 | 7 | 6 | 8 |
| 20 | 8 | 8 | 8 | 15 | 8 | 7 | 8 | 8 | 8 | 8 | 10 | 6 | 7 | 8 | 8 | 8 | 8 | 15 | 5 | 6 | 8 |
| 21 | 8 | 8 | 8 | 13 | 6 | 8 | 8 | 8 | 8 | 8 | 7 | 8 | 7 | 8 | 8 | 8 | 8 | 20 | 4 | 7 | 8 |
| 22 | 9 | 9 | 9 | 9 | 6 | 9 | 9 | 8 | 8 | 8 | 10 | 8 | 8 | 8 | 8 | 8 | 8 | 10 | 5 | 8 | 8 |
| 23 | 8 | 8 | 8 | 12 | 8 | 7 | 8 | 8 | 8 | 8 | 16 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 7 | 6 | 8 |
| 24 | 8 | 8 | 8 | 13 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 4 | 7 | 8 | 8 | 7 | 7 | 13 | 5 | 8 | 7 |
| 25 | 8 | 8 | 8 | 9 | 7 | 8 | 8 | 8 | 8 | 8 | 9 | 6 | 6 | 8 | 8 | 8 | 8 | 11 | 7 | 7 | 8 |
| 26 | 8 | 8 | 8 | 11 | 8 | 6 | 8 | 9 | 8 | 8 | 9 | 8 | 8 | 7 | 8 | 8 | 8 | 13 | 6 | 7 | 8 |
| 27 | 8 | 8 | 8 | 11 | 6 | 7 | 7 | 8 | 8 | 8 | 9 | 7 | 7 | 8 | 9 | 8 | 8 | 10 | 7 | 8 | 8 |
| 28 | 8 | 9 | 8 | 8 | 7 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 7 | 8 | 8 | 8 | 8 | 8 | 4 | 6 | 8 |
| 29 | 8 | 8 | 8 | 12 | 5 | 8 | 8 | 8 | 8 | 8 | 15 | 7 | 8 | 7 | 8 | 8 | 8 | 9 | 6 | 6 | 7 |
| 30 | 8 | 8 | 8 | 9 | 7 | 6 | 8 | 8 | 8 | 8 | 8 | 8 | 6 | 8 | 8 | 8 | 8 | 14 | 5 | 7 | 7 |
| 31 | 8 | 8 | 8 | 12 | 8 | 8 | 8 | 8 | 8 | 8 | 16 | 6 | 7 | 7 | 8 | 8 | 8 | 11 | 7 | 8 | 8 |
| 32 | 8 | 8 | 8 | 14 | 8 | 8 | 8 | 8 | 8 | 8 | 7 | 7 | 7 | 8 | 8 | 8 | 8 | 12 | 6 | 4 | 8 |
| 33 | 8 | 8 | 8 | 9 | 3 | 7 | 8 | 8 | 8 | 8 | 7 | 7 | 6 | 7 | 8 | 8 | 8 | 10 | 8 | 5 | 8 |
| 34 | 8 | 8 | 8 | 13 | 8 | 8 | 8 | 8 | 8 | 8 | 20 | 4 | 5 | 8 | 8 | 8 | 8 | 10 | 5 | 7 | 8 |
| 35 | 8 | 8 | 8 | 12 | 4 | 8 | 8 | 8 | 8 | 8 | 9 | 6 | 8 | 8 | 9 | 9 | 9 | 10 | 6 | 6 | 9 |
| 36 | 8 | 8 | 8 | 10 | 8 | 6 | 8 | 9 | 9 | 9 | 9 | 7 | 9 | 9 | 8 | 9 | 8 | 8 | 7 | 7 | 8 |
| 37 | 8 | 7 | 7 | 8 | 7 | 6 | 7 | 9 | 9 | 9 | 10 | 9 | 6 | 9 | 8 | 8 | 7 | 13 | 4 | 7 | 7 |
| 38 | 8 | 8 | 8 | 12 | 7 | 8 | 8 | 8 | 8 | 8 | 12 | 8 | 8 | 8 | 8 | 9 | 8 | 19 | 5 | 6 | 8 |
| 39 | 8 | 8 | 8 | 11 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 7 | 8 | 8 | 8 | 8 | 8 | 15 | 6 | 7 | 8 |
| 40 | 8 | 8 | 8 | 12 | 8 | 7 | 8 | 8 | 8 | 8 | 9 | 6 | 7 | 8 | 8 | 8 | 8 | 9 | 5 | 8 | 8 |
| 41 | 8 | 8 | 8 | 8 | 7 | 7 | 7 | 8 | 8 | 8 | 20 | 8 | 7 | 8 | 8 | 7 | 7 | 10 | 8 | 8 | 7 |

| | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 42 | 8 | 8 | 8 | 13 | 8 | 8 | 8 | 8 | 8 | 8 | 9 | 7 | 6 | 8 | 9 | 9 | 8 | 14 | 7 | 5 | 8 |
| 43 | 8 | 8 | 8 | 12 | 3 | 8 | 8 | 8 | 8 | 8 | 12 | 7 | 8 | 8 | 9 | 9 | 9 | 8 | 8 | 7 | 8 |
| 44 | 8 | 8 | 8 | 10 | 8 | 6 | 8 | 8 | 8 | 8 | 10 | 8 | 6 | 8 | 8 | 8 | 8 | 15 | 5 | 6 | 8 |
| 45 | 8 | 8 | 8 | 11 | 7 | 8 | 8 | 8 | 8 | 8 | 9 | 6 | 7 | 8 | 8 | 8 | 8 | 15 | 5 | 8 | 7 |
| 46 | 9 | 9 | 9 | 9 | 6 | 9 | 9 | 9 | 9 | 9 | 13 | 7 | 6 | 9 | 8 | 8 | 8 | 13 | 6 | 5 | 8 |
| 47 | 8 | 8 | 8 | 12 | 8 | 7 | 7 | 8 | 8 | 8 | 12 | 3 | 7 | 8 | 9 | 9 | 9 | 15 | 6 | 5 | 9 |
| 48 | 8 | 9 | 8 | 15 | 8 | 8 | 8 | 8 | 8 | 8 | 7 | 8 | 5 | 8 | 8 | 8 | 8 | 12 | 7 | 7 | 8 |
| 49 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 9 | 9 | 9 | 9 | 6 | 8 | 9 | 8 | 8 | 8 | 10 | 5 | 4 | 8 |
| 50 | 8 | 8 | 8 | 8 | 8 | 7 | 8 | 8 | 8 | 8 | 10 | 7 | 5 | 8 | 9 | 8 | 8 | 15 | 7 | 5 | 8 |
| 51 | 8 | 8 | 7 | 11 | 7 | 7 | 8 | | | | | | | | | | | | | | |
| 52 | 8 | 8 | 8 | 8 | 8 | 7 | 7 | | | | | | | | | | | | | | |
| 53 | 8 | 8 | 8 | 13 | 8 | 7 | 8 | | | | | | | | | | | | | | |
| 54 | 8 | 8 | 8 | 9 | 6 | 8 | 8 | | | | | | | | | | | | | | |
| 55 | 8 | 8 | 8 | 11 | 8 | 8 | 8 | | | | | | | | | | | | | | |
| 56 | 8 | 8 | 8 | 8 | 8 | 6 | 8 | | | | | | | | | | | | | | |
| 57 | 8 | 8 | 8 | 14 | 8 | 8 | 8 | | | | | | | | | | | | | | |
| 58 | 8 | 8 | 8 | 14 | 8 | 7 | 8 | | | | | | | | | | | | | | |
| 59 | 9 | 9 | 9 | 11 | 9 | 8 | 9 | | | | | | | | | | | | | | |
| 60 | 8 | 8 | 8 | 12 | 8 | 8 | 8 | | | | | | | | | | | | | | |
| 61 | 8 | 8 | 8 | 14 | 8 | 8 | 8 | | | | | | | | | | | | | | |
| 62 | 8 | 8 | 8 | 13 | 8 | 7 | 8 | | | | | | | | | | | | | | |
| 63 | 8 | 8 | 8 | 14 | 8 | 8 | 8 | | | | | | | | | | | | | | |
| 64 | 8 | 8 | 8 | 8 | 7 | 7 | 8 | | | | | | | | | | | | | | |
| 65 | 8 | 8 | 8 | 13 | 8 | 8 | 8 | | | | | | | | | | | | | | |
| 66 | 8 | 8 | 8 | 8 | 8 | 7 | 7 | | | | | | | | | | | | | | |
| 67 | 8 | 8 | 8 | 15 | 6 | 8 | 8 | | | | | | | | | | | | | | |
| 68 | 8 | 8 | 7 | 12 | 7 | 6 | 7 | | | | | | | | | | | | | | |
| 69 | 8 | 8 | 8 | 14 | 8 | 8 | 8 | | | | | | | | | | | | | | |
| 70 | 8 | 8 | 8 | 9 | 8 | 8 | 8 | | | | | | | | | | | | | | |
| 71 | 8 | 8 | 8 | 14 | 7 | 7 | 8 | | | | | | | | | | | | | | |

| | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 72 | 8 | 8 | 8 | 13 | 8 | 8 | 8 | | | | | | | | | | | | | | |
| 73 | 8 | 8 | 8 | 11 | 8 | 8 | 8 | | | | | | | | | | | | | | |
| 74 | 8 | 8 | 8 | 12 | 8 | 7 | 8 | | | | | | | | | | | | | | |
| 75 | 8 | 8 | 8 | 10 | 6 | 8 | 8 | | | | | | | | | | | | | | |
| 76 | 8 | 8 | 8 | 12 | 8 | 7 | 8 | | | | | | | | | | | | | | |
| 77 | 8 | 8 | 7 | 9 | 7 | 7 | 7 | | | | | | | | | | | | | | |
| 78 | 8 | 8 | 8 | 11 | 8 | 8 | 8 | | | | | | | | | | | | | | |
| 79 | 8 | 8 | 8 | 8 | 6 | 7 | 7 | | | | | | | | | | | | | | |
| 80 | 8 | 8 | 8 | 10 | 8 | 7 | 7 | | | | | | | | | | | | | | |
| 81 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | | | | | | | | | | | | | | |
| 82 | 8 | 8 | 8 | 9 | 8 | 8 | 8 | | | | | | | | | | | | | | |
| 83 | 8 | 8 | 8 | 15 | 8 | 6 | 8 | | | | | | | | | | | | | | |
| 84 | 8 | 8 | 8 | 11 | 6 | 8 | 8 | | | | | | | | | | | | | | |
| 85 | 8 | 8 | 8 | 13 | 8 | 8 | 8 | | | | | | | | | | | | | | |
| 86 | 8 | 8 | 8 | 10 | 8 | 7 | 8 | | | | | | | | | | | | | | |
| 87 | 8 | 8 | 8 | 12 | 8 | 8 | 8 | | | | | | | | | | | | | | |
| 88 | 8 | 8 | 8 | 11 | 5 | 7 | 7 | | | | | | | | | | | | | | |
| 89 | 8 | 8 | 8 | 11 | 8 | 7 | 8 | | | | | | | | | | | | | | |
| 90 | 8 | 8 | 8 | 12 | 8 | 8 | 8 | | | | | | | | | | | | | | |
| 91 | 8 | 8 | 8 | 8 | 6 | 8 | 8 | | | | | | | | | | | | | | |
| 92 | 8 | 8 | 8 | 10 | 8 | 7 | 7 | | | | | | | | | | | | | | |
| 93 | 8 | 8 | 8 | 12 | 8 | 8 | 8 | | | | | | | | | | | | | | |
| 94 | 8 | 8 | 8 | 11 | 5 | 8 | 8 | | | | | | | | | | | | | | |
| 95 | 9 | 9 | 9 | 12 | 9 | 9 | 9 | | | | | | | | | | | | | | |
| 96 | 9 | 9 | 9 | 10 | 9 | 9 | 9 | | | | | | | | | | | | | | |
| 97 | 8 | 8 | 8 | 15 | 8 | 8 | 8 | | | | | | | | | | | | | | |
| 98 | 8 | 8 | 8 | 10 | 6 | 8 | 8 | | | | | | | | | | | | | | |
| 99 | 8 | 8 | 8 | 11 | 8 | 8 | 8 | | | | | | | | | | | | | | |
| 100 | 8 | 8 | 8 | 11 | 8 | 8 | 8 | | | | | | | | | | | | | | |
| Итог | 807 | 805 | 799 | 1108 | 726 | 751 | 791 | 405 | 402 | 401 | 530 | 324 | 348 | 397 | 408 | 406 | 400 | 609 | 298 | 330 | 392 |

# Appendix B

# K - Nearest Neighbors

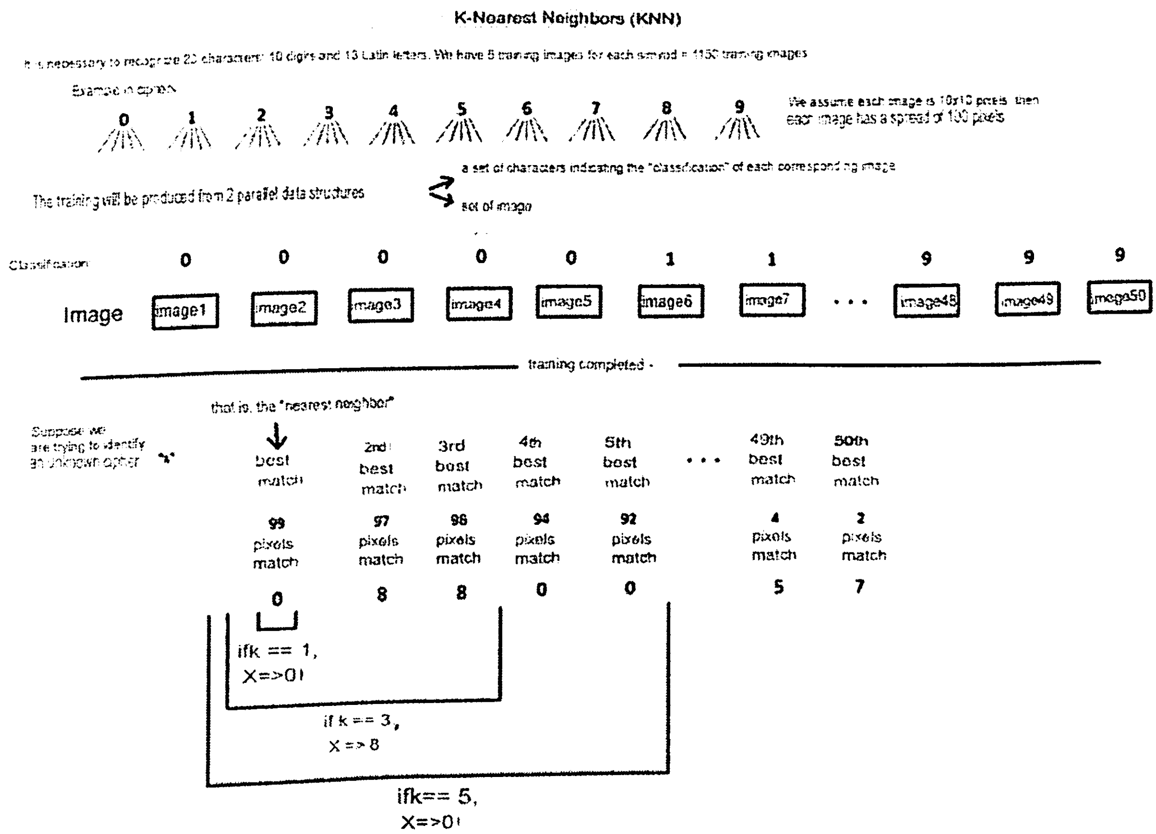


Figure B.1: Description of the algorithm.

# References

[1] Yu. A. Bolotova. "Recognition of license plates based on the method of connected components and hierarchical time network". In: *Computer Optics: scientific journal* 39.2 (2015), pp. 275–280.

[2] Khvedchenia I. David M. E. Shervin E. "Mastering OpenCV with Practical Computer Vision Projects". In: *Baggio Published by Packt Publishingl* (2012).

[3] D. V. Dubrov. "Programming: CMake project building system". In: *Yurayt Publishing House* 5.2 (2018), p. 422.

[4] Omer I. Saqib R. Asad N. "Automated Number Plate Recognition Using Hough Lines and Template Matching". In: *Proceedings of the World Congress on Engineering and Computer Science* 1.3 (2012), pp. 233–123.

[5] M. N. Rudometkina Yu. A. Bolotova V. G. Spitsyn. *The Principles of Quantum Mechanics.* International series of monographs on physics. Clarendon Press, 1981.

[6] Donald E. Knuth. "Fundamental Algorithms". In: Addison-Wesley, 1973. Chap. 1.2.

[7] Khanin A. Svirin I. "Some aspects of automatic car license plate recognition". In: *Security algorithms* 3 (2010), pp. 26–29.

[8] Siyu Zhu. "An End-to-End License Plate Localization and Recognition System". In: *Department of Electrical Engineering Kate Gleason College of Engineering Rochester Institute of Technology Rochester, New York* (2015).

[9] Hsieh C. Hung K. "A Real-Time Mobile Vehicle License Plate Detection and Recognition". In: *Tamkang Journal of Science and Engineering* 13.4 (2010), pp. 433–442.

[10] Makarov A. A. Yakovleva E. S. "On the properties of the block algorithm for binarization of digital images". In: *Computer tools in education* 5.4 (2015), pp. 26–36.

[11] Fisenko T.Yu. Fisenko V.T. *Computer processing and image recognition.* International series of monographs on physics. St. Petersburg State University ITMO, 2008.

[12] Atilla O. Emir T. *Edge Detection Using Steerable Filters and CNN.* Akman, 2002.

[13] D. Gaur A. Nagpal A. Jatain. "Review based on data clustering algorithms". In: *IEEE Conference, Information Communication Technologies (ICT)* (2013), pp. 298–303.

[14] D. Wunsch R. Xu. "Survey of clustering algorithms". In: *IEEE Transactions, Neural Networks* 16.3 (2005), pp. 645–678.

[15] Fahad A. "A survey of clustering algorithms for big data: Taxonomy and empirical analysis". In: *IEEE Transactions, Emerging Topics in Computing* 2.3 (2014), pp. 267–279.

[16] Canny J.F. *Finding edges and lines in images.* MIT, Cam- bridge, USA, 1983, pp. 50–67.

[17] P. Viola and M.J. Jones. "Robust real-time face detection". In: *International Journal of Computer Vision* 57.2 (2004), pp. 137–154.

[18] Khaykin S. "Neural networks". In: *Publishing House Williams* 2.21 (2006), pp. 351–356.