# TRaffic Sign Classifier

*by*

Kerem Ali Korkmaz, 116200088
Mehmet Berke Toraman, 116200026
Emre Yavuz, 116200059

*Supervised by*

Elena Battini Snmez

*Submitted to the*
Faculty of Engineering and Natural Sciences
*in partial fulfillment of the requirements for the*

Bachelor of Science

*in the*
Department of Computer Engineering

January, 2021

## ***Abstract***

*With self-driving cars arriving, it has become important to automatically recognize traffic and hand-held signs. Besides self-driving cars, drivers who do not know the meaning of a traffic sign may cause accidents or cause the traffic to get crowded. This project aims to prevent them. Recognizing the traffic signs correctly using intelligent systems can reduce the traffic accidents all around the world, help the environment be safer, healthier and the traffic be more ordered and less crowded.*
*The goal is to use cameras for recognizing the traffic signs. The first part of the task is to survey all traffic signs.*
*German Traffic Sign Recognition data-set (GTSRB) is being used as a base for the project. It reflects the variations in visual appearance of signs due to distance, illumination, weather conditions, partial occlusions, and rotations. In addition, this research plans to add local Turkish traffic signs to the German data-set, by collecting images from different positions and orientations, because variety allows simulating real-world situations. Ten new Turkish signs, which do not exist in the German data-set, will be added by the project developers.*
*This paper gives an example for efficient recognition of traffic signs from images. It is a challenging task because the same sign's appearances can be very different from one another in presence of disturbance elements such as rotations, abrasions, distance, and illumination.*

# Contents

# List of Figures

# List of Tables

# List of Symbols

| | |
|---|---|
| $a, b, c$ | Scalars |

# List of Abbreviations

| | |
|---|---|
| etc. | Et cetera (Latin: and such) |
| HOG | Histogram of Orianted Gradients |
| LBP | Local Binary Pattern |
| SIFT | Scale-invariant Feature Transform |
| BRISK | Binary Robust Invariant Scalable Keypoints |
| CNN | Convolutional Neural Network |
| ms | millisecond |
| GTSRB | German Traffic Sign Recognition Benchmark |
| BTSC | Belgium Traffic Sign Classifier Benchmark |
| DITS | The Database of Italian Traffic Signals |
| RevisedMastif | The Revised Mapping and Assessing the State of Traffic Infrastructure |
| i.e. | Id est (Latin: this means) |
| GHT | Generalized Hough Transform |
| STN | Spatial Trans-former Networks |
| ELM | Extreme Learning Machine |
| DCT | Discrete Cosine Transform |
| ACF | Aggregate channel features |
| SVM | Support vector machine |
| LDA | Latent Dirichlet allocation |
| MLP | multilayer perceptron |
| RGB | red-blue-green |

# 1 Introduction

Traffic means opening access roads to vehicles and pedestrians. Traffic is part of our lives. Nowadays, people who want to go from one place to another set off by private vehicles, public transportation vehicles or on foot. Therefore, traffic has an impact on our daily life. Time, energy, money, speed, security and comfort come to the fore in traffic and Traffic works according to the rules. Failure to comply with the rules will result in loss of life and property. Obeying to Traffic signs are one of these major traffic rules so understanding the meaning of signs is very important [2].It has been observed that signage violations in traffic accidents have played a huge role [3].

With the development of technology nowadays, autonomous vehicles entered our lives. The biggest reason for the success of these vehicles is that they have a more developed perception system than human intelligence. Today, traffic signs in countries are too many as a number. For example, there are 243 traffic signs at Turkey and there are 500 traffic signs at USA and countries have some signs which have different local meaning. Also, Signs of traffic show a big range of options between classes in terms of text, color, and shape but there are subsets of classes such as speed limit signs which are very similar. The classifier has to deal with wide variations in visual views due to lighting changes, partial blockages, rotations, weather, scaling, etc.[4].The topics are mentioned give a crucial reason why machine memory is preferred over human memory in traffic so having an advanced sign recognition and classification system of vehicles is significant for the financial and vital health of both the customer and other pedestrians.

The development of a successful system is determined by the presence of a good database; the used database should be rich and for Turkish traffic sign classifier, a traffic sign must have photographs taken in many different situations. In summary, the data set must be well prepared after that the use of specific machine learning and/or deep learning algorithms will create a model, which must be for real time situations. A well-prepared model won't cause any problems during driving [5].


Figure 1: Traffic Signs

The rest of the paper is organized as follows.
Section 2 introduces the topic.
Section 3 overviews the common used databases of traffic signs and introduces the Turkish traffic sign database, TRaffic Sign.
Section 4 details main algorithms used for feature extraction and classifications.
Section 5 reports the run experiments with performance.
Sections 6 and 7 explain the used design methodology and risk managements.
Section 8 and 9 highlight the main contribution of this work and conclude the thesis.

# 2 Literature Review

Traffic sign recognition has been a very interesting topic for programmers, when we examine the recent studies, we observe the following results; The work of Alexander and Pavel was published in 2017. [6] They used German Traffic Sign Benchmark (GTSRB) dataset; their best performance was reached with Generalized Hough Transform (GHT) method and Convolutional Neural Network (CNN, or ConvNet), equal to 99.94%.
In 2018, Alvaro's [7] challenged the German Traffic Sign database (GTSRB); their best performance was reached with single CNN with 3 Spatial Transformer Networks (STN), equal to 99.71%. Also they have an another performance on Belgium Traffic Sign Classificaton (BTSC) dataset [8] with same method and they reached the performance 98.87%.
In 2019, Zhang's [9] challenged the GTSRB dataset; their best performance was reached with special method that they call "Student Network"; this network is characterized by an end-to-end structure consisting of five convolutional layers and a fully connected layer, equal to 99.13%.
In 2018, Aziz's [10] challenged the GTSRB and BTSC datasets; their best performance was reached with Extreme Learning Machine (ELM), equal to 99.10% on GTSRB and best performance for BTSC is equal to 98.30%.
In 2020, Franzen's [11] challenged the German Traffic Sign database; their best performance was reached with Discrete Cosine Transform (DCT), equal to 99.6%.
As we extend our investigations into the middle period, we notice some of the widely used methods. They can be listed as for feature extraction; Histogram of Oriented Gradient (HOG), Local Binary Pattern (LBP), Scale-invariant Feature Transform (SIFT) and Binary Robust Invariant Scalable Key-points (BRISK), Aggregate channel features (ACF) and Gabor [12],[13], [14]. Also, Classifiers can be listed as N-Bayes,Support vector machine(SVM),Random forest, Extreme learning machine (ELM), Adaboost, latent Dirichlet allocation (LDA), multilayer perceptron (MLP) and Convolutional Neural Network (CNN).Table 3 summarizes the workings on GTSRB dataset with different combinations of methods.

| Authors | Released Year | Feature-Classifier | Accuracy(%) |
|---|---|---|---|
| Aghdam et al. [15] | 2015 | Visual attributes with N-Bayes | 98.04 |
| Berkaya et al. [16] | 2016 | Gabor with SVM | 93.90 |
| Ellahyani et al. [17] | 2016 | HOG with Random Forest | 97.43 |
| He and Dai et al. [18] | 2016 | LBP with SVM | 97.67 |
| Li and Yang et al.[19] | 2016 | HOG with ELM | 95.16 |
| Qu et al. [20] | 2016 | ACF with Adaboost | 95.97 |
| Stalkamp et al. [21] | 2012 | HOG with LDA | 93.18 |
| Aghdam et al.[22] | 2016 | CNN | **99.55** |
| Cirean et al. [23] | 2011 | MLP and HOG with CNN | 99.15 |
| Eickeler et al.[24] | 2016 | DNN | 97.5 |
| Qian et al.[25] | 2016 | CNN and MLP | 98.86 |
| Xie et al.[26] | 2016 | CNN | 97.94 |
| Youssef et al. [27] | 2016 | Single scale CNN | 97.2 |
| Zeng et al.[28] | 2015 | CNN and ELM | 99.40 |
| Tang et al.[29] | 2013 | HOG with SVM | 95.92 |

Table 1: Best Performances on the German Traffic Sign database from [1]. Best performance is in black.

As it can be understood from the table, using deep learning algorithms for classification has better accuracy, so we are going to use one of the deep learning algorithms which is CNN(Convolutional Neural Network) for the classification part of our project. [30] So looking at the literature review, the main algorithms used in traffic sign recognition are hand-crafted features and deep learning methods. This work will extend the GTSRB database with Turkish signs and it will compile the performance with the CNN algorithm.

# 3 Databases for TRaffic Sign

All researchers tackling the issue of traffic sign classification work with common databases.
The German Traffic Sign Recognition Data-Set (GTSRB) [31] database was released in 2011, it is made up of 43 signs, for a total of 50,000 images. For every symbol, there is an average of 1100 images. The size of the images are $32\times32$
The Belgium Traffic Sign Classification (BTSC) [8] database stores a total of 6000 images for 62 different symbols and roughly 100 images for a symbol.
The Revised Mapping and Assessing the State of Traffic Infrastructure (MASTIF) [32] database collects 6,000 image of 30 symbols with the average 200 images for a symbol.
The Database of Italian Traffic Signals (DITS) [33] database includes 58 different signs and averagely 160 images for a sign with the total of 9254 images.
Table 2 summarizes those databases.

| Dataset | Classes | Images | Country | Size |
|---|---|---|---|---|
| GTSRB | 43 | 50000 | Germany | $32 \times 32$ |
| BTSC | 62 | 6000 | Belgium | NA |
| Revised MASTIF | 30 | 6000 | Croatia | $64 \times 64$ pixels |
| DITS | 58 | 9254 | Italia | NA |

Table 2: Common used databases on Traffic Sign [1]

This work extends the German Traffic Sign Recognition Database [31] by adding Turkish signs; the resulting database is called **TR**affic Sign Classifying database, where **TR** stands for Turkey. We added 10 Turkish sign classes to the dataset and there are 200 images per class and their pixels are 32x32 as like other German traffic signs. This is because, GTSRB has sufficient labeled data for training from various environments and it is known that this data set works efficiently with the CNN algorithm, which is the algorithm to be used in this project.
Figure 2 details some German traffic signs.
Figure 3 shows the Turkish signs, which will be added to the original German database.


Figure 2: Figure of GTSRB dataset


Figure 3: Figure of some Turkish traffic signs

# 4 Algorithms

In this part we examined popular feature extraction methods.

## 4.1 Feature Extraction

**Histogram of Oriented Gradients (HOG)**
The Histogram of Oriented Gradients (HOG) is a function descriptor used for object detection in computer vision and image processing. The technique counts gradient orientation occurrences in localized portions of an image. This approach is similar to that of histograms of edge orientation, descriptors of scale-invariant feature transformation, and shape contexts, but differs in that it is measured for better precision on a dense grid of uniformly spaced cells and uses overlapping local contrast normalization.

The important concept behind the histogram of the descriptor of oriented gradients is that the distribution of intensity gradients or edge directions can define the appearance and shape of local objects within a picture. The image is divided into small connected regions called cells and a histogram of gradient directions is compiled for the pixels within each cell. A concatenation of these histograms is the descriptor. By measuring a measure of the intensity over a larger region of the image, called a block, and then using this value to normalize all cells within the block, the local histograms can be contrast-normalized for improved accuracy. This normalization leads to a greater variance of lighting and shadowing shifts [34] .

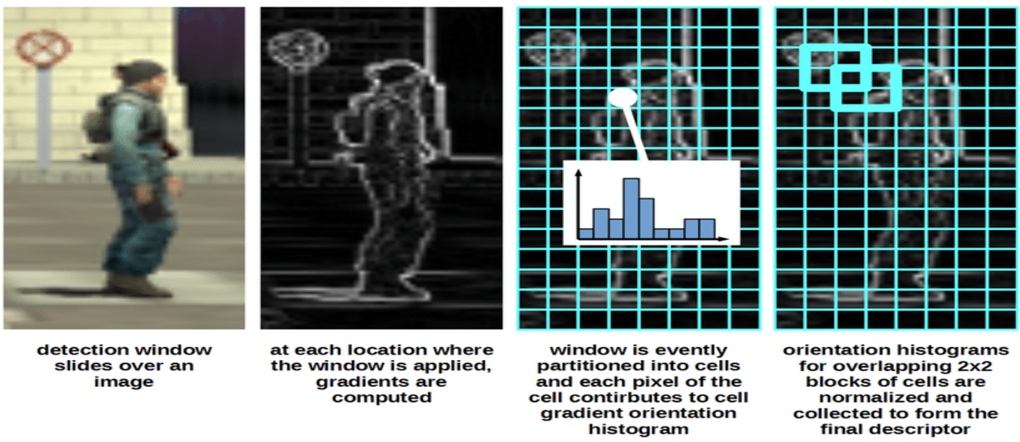


Figure 4: Figure of History Oriented Programming

**Haar-like Feature**

Haar-like features are digital image features used in object recognition.
Haar-like feature considers adjacent rectangular regions at a specific location in a detection window, sums up the pixel intensities in each region, and calculates the difference between these sums. This difference is then used to categorize subsections of an image.
For example, with a human face, it is a common observation that among all faces the region of the eyes is darker than the region of the cheeks. Therefore, a common Haar feature for face detection is a set of two adjacent rectangles that lie above the eye and the cheek region. The position of these rectangles is defined relative to a detection window that acts like a bounding box to the target object (the face in this case).

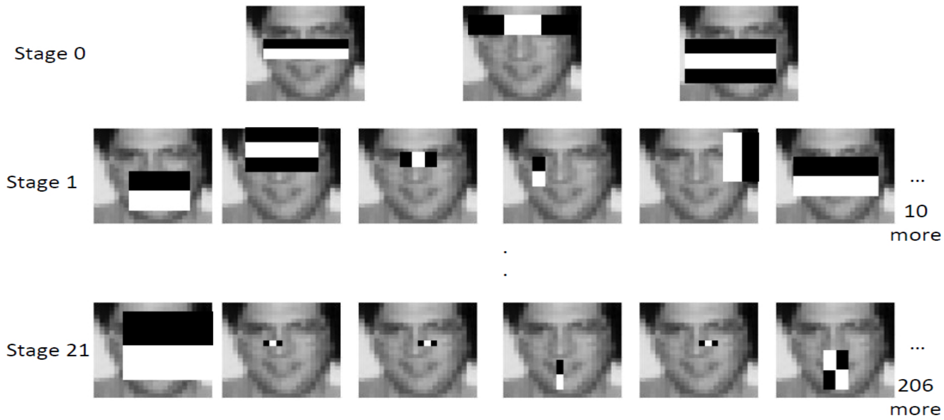


Figure 5: Figure of Haar-like Feature

**Hue Histogram**

When you refer to Hue, you are referring to its pure color or the visible spectrum of basic colors that can be seen in a rainbow. Hue-histogram can also be called a color histogram.
A color histogram of an image represents the distribution of the composition of colors in the image. For digital images, a color histogram represents the number of pixels that have colors in each of a fixed list of color ranges, that span the image's color space, the set of all possible colors. It shows different types of colors appeared and the number of pixels in each type of the colors that appeared.

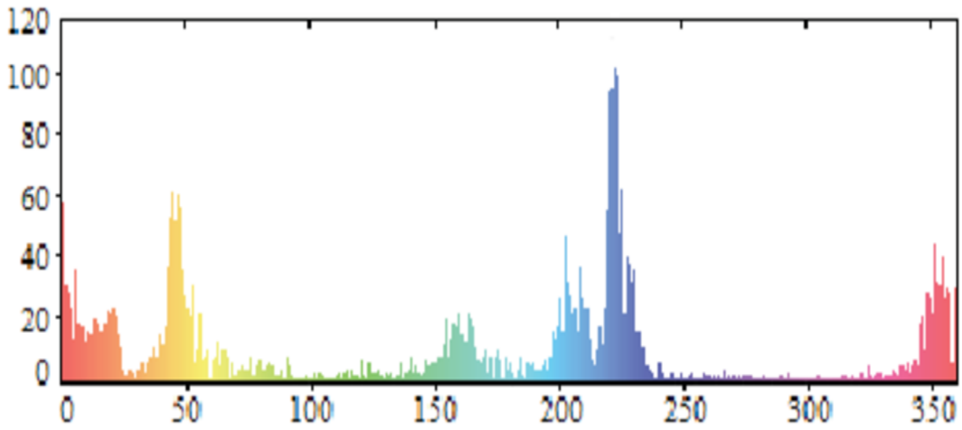

Figure 6: Figure of Hue-Histogram

**Local Binary Patterns**

Local Binary Patterns (LBP) is a simple yet very efficient texture operator which labels the pixels of an image by thresholding the neighborhood of each pixel and considers the result as a binary number. due to its discriminative power and computational simplicity, the LBP texture operator has become a popular approach in various applications. Perhaps the most important property of the LBP operator in real-world applications is its stability to monotonic gray-scale changes caused by illumination variations. Another important property is its computational simplicity, which makes it possible to analyze images in challenging real-time settings.

They produce rather long histograms, which slow down the recognition speed especially on large-scale databases which is the reason for us to not use it [35].

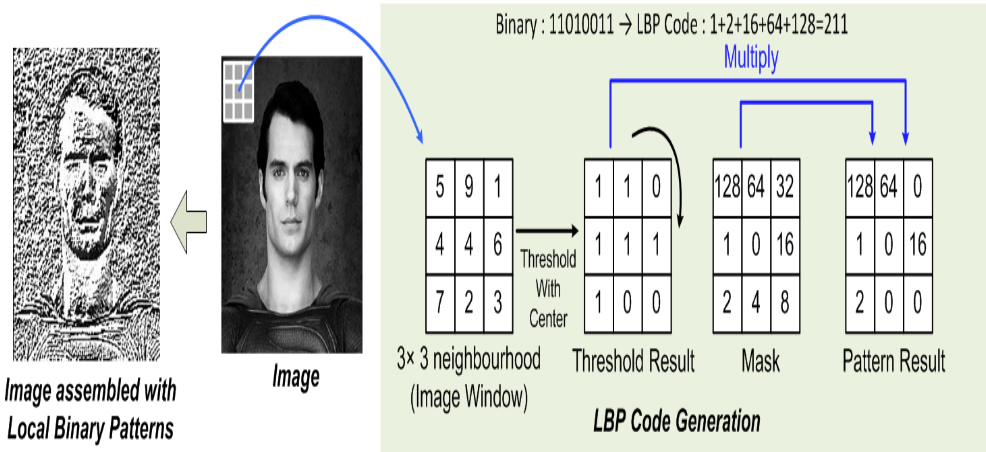


Figure 7: Figure of Local Binary Patterns

**Gabor**

In image processing, a Gabor filter is a linear filter used for texture analysis which basically means that it analyzes whether there is any particular frequency material in the image in specific directions around the point or area of the analysis in a localized region. Many contemporary vision scientists believe that the frequency and orientation representations of Gabor filters are very similar to those of the sign recognition system. For texture representation and discrimination, they have been found to be especially suitable. A 2D Gabor filter in the spatial domain is a Gaussian kernel function modulated by a sinusoidal plane wave.
Gabor filter takes too much time for performing features due to its dimension of the feature vector is very long and it has high redundancy of features. These are the reasons for us to not use it.

Simple cells in the visual cortex of mammalian brains can be modeled by Gabor functions, some authors say. Thus, some consider image processing with Gabor filters to be close to perception in the system [36] .

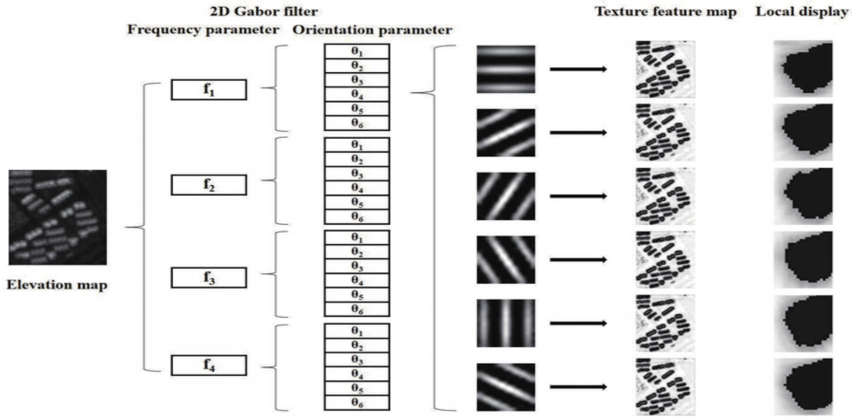


Figure 8: Figure of Gabor

**Brisk**

The Binary Robust Invariant Scalable Keypoints (BRISK) algorithm is a feature point detection and description algorithm with scale invariance and rotation invariance. It constructs the feature descriptor of the local image through the grayscale relationship of random point pairs in the neighborhood of the local image, and obtains the binary feature descriptor.
Compared with other algorithms, the matching speed of BRISK is faster and the storage memory is lower, but the robustness of BRISK is reduced. We need high robustness and this is the reason why we are not using it [37].

## 4.2 Data Augmentation

In the newly introduced **TRaffic Sign** database 10 new classes for Turkish signs are added and a total of 2000 new pictures have been added to the data-set, 200 images per class. The German Traffic Sign Recognition Data-Set (GTSRB) [31] database was released in 2011, it is made up of 43 signs, for a total of 50,000 images.
Data augmentation technique is necessary to increase the total number of images before attempt to classify them. It can reduce overfitting and improve the generalization of our model because it increases the diversity of our training set.
In data augmentation, we use ImageDataGenerator with the width of the value shift range, height shift range, zoom range, sheer range, and rotation range. With these properties, we get a more generic data set. Also Haar-like feature and Hue-Histogram feature have been employed.

## 4.3 Classification

In this part we examined popular classification methods.

**Decision Tree**

Tree-based learning algorithms are one of the widely used supervised learning algorithms. In general, they can be adapted to the classification and regression solution of all the problems discussed. Decision tree algorithm is one of the data mining classification algorithms so A decision tree is a structure used to divide a data set containing a large number of records into smaller sets by applying a set of decision rules. In other words, it is a structure that is used by applying simple decision-making steps, dividing large amounts of records into very small groups of records but not useful for this project. This is because, Overly complex trees can be produced that do not explain the data well. In this case, tree branching may not be followed and over-fitting is can be a problem [38].

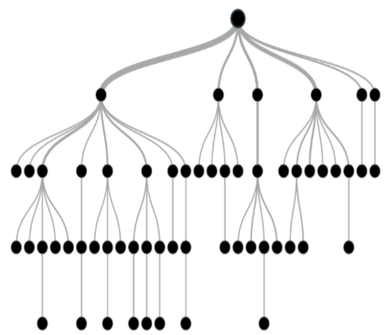

Figure 9: Decision Tree

**Genetic Algorithm**

Genetic Algorithm is a search-based optimization technique based on the principles of Genetics and Natural Selection. It is frequently used to find optimal or near-optimal solutions to difficult problems which otherwise would take a lifetime to solve. It is frequently used to solve optimization problems in machine learning but a Genetic algorithm is not useful for this project too. This is because, Being stochastic, there are no guarantees on the optimality or the quality of the solution [39].


Figure 10: Genetic Algorithm

**AdaBoost**

We can say the first boosting algorithm for AdaBoost.This method helps you combine multiple weak classifiers into a single strong classifier, AdaBoost works by putting more weight on difficult to classify instances and less on those already handled well. AdaBoost also has few disadvantages such as it is from empirical evidence and particularly vulnerable to uniform noise, weak classifiers being too weak can lead to low margins and over-fitting [40] .

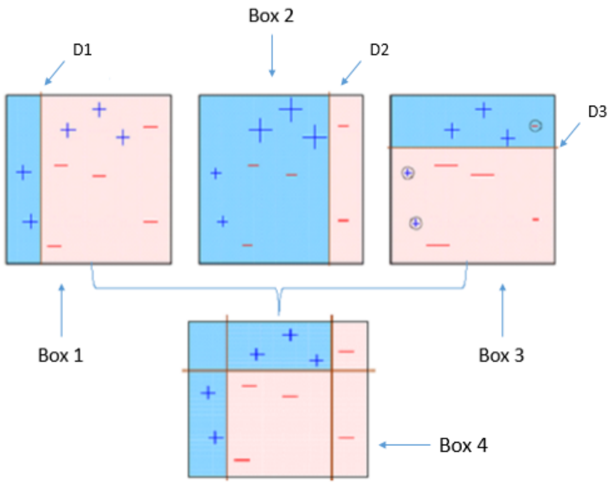


Figure 11: AdaBoost

**Support Vector Machine**

We can think about examples of two classes that are distributed linearly. In this case, it is aimed to distinguish these two classes with the help of a decision function obtained using training data. It is called the correct decision line that divides the data set into two. Although it is possible to draw infinite decision lines, the important thing is to determine the optimal decision line. In order for the decision line to be resistant to the newly added data, the borderline must be at the closest distance to the borderlines of the two classes. The points closest to this borderline are called support points. These machine learning algorithms are useful but not as good as the CNN algorithm. This is because this algorithm can not generate probabilistic predictions [41].

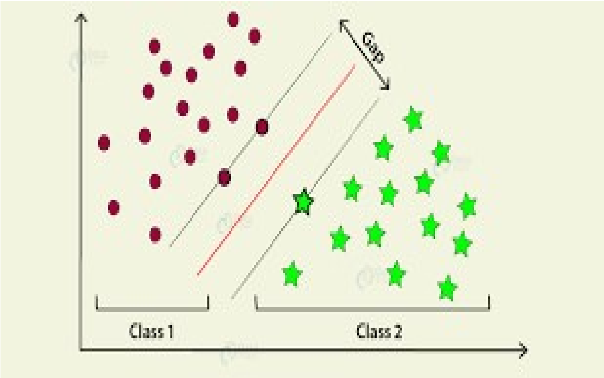


Figure 12: Support Vector Machine

**Convolution Neural Network (CNN)**
In deep learning, a convolutional neural network (CNN) is a class of deep neural networks, most commonly applied to analyzing visual imagery.
CNNs are regularized versions of multilayer perceptrons. Multilayer perceptrons usually mean fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. These neurons are made of learnable weights and biases. Each specific neuron receives numerous inputs and then takes a weighted sum over them, where it passes it through an activation function and responds back with an output.
CNNs are primarily used to classify images, cluster them by similarities, and then perform object recognition. Many algorithms using CNNs can identify faces, street signs, animals, etc.

**How does it work**
They are prompt by volume and utilize multi-channeled images as opposed to flat images that humans can see that only have width and heigh. Due to digital color images having red-blue-green (RGB) encoding, CNNs mix those three colors to produce the color spectrum humans perceive. A convolutional network ingests such images as three separate strata of color stacked one on top of the other. The depth layers in the three layers of colors (RGB) interpreted by CNNs are referred to as channels.

**The first layer is the CONVOLUTIONAL LAYER**, which is the core building block and does most of the computational heavy lifting. Data or imaged is convolved using filters or kernels. Filters are small units that we apply across the data through a sliding window. The depth of the image is the same as the input, for a color image that RGB value of depth is 4, a filter of depth 4 would also be applied to it. This process involves taking the element-wise product of filters in the image and then summing those specific values for every sliding action. The output of a convolution that has a 3d filter with color would be a 2d matrix.

**Second is the ACTIVATION LAYER** which applies the ReLu (Rectified Linear Unit), in this step we apply the rectifier function to increase non-linearity in the CNN. Images are made of different objects that are not linear to each other.

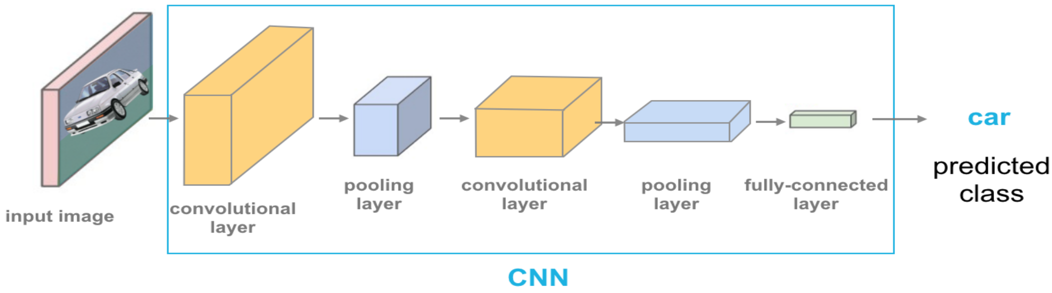


Figure 13: CNN

**Third, is the POOLING LAYER**, which involves the downsampling of features. It is applied through every layer in the 3d volume. Typically there are hyperparameters within this layer:
1-**The dimension of spatial extent:** which is the value of n which we can take N cross and feature representation and map to a single value.
2-**Stride:** which is how many features the sliding window skips along the width and height.

A common POOLING LAYER uses a 2 cross 2 max filter with a stride of 2, this is a non-overlapping filter. A max filter would return the max value in the features within the region. Example of max pooling would be when there is 26 across 26 across 32 volume, now by using a max pool layer that has 2 cross 2 filters and astride of 2, the volume would then be reduced to 13 crosses, 13 crosses 32 feature map.

**Last one is the FULLY CONNECTED LAYER**, which involves Flattening. This involves transforming the entire pooled feature map matrix into a single column which is then fed to the neural network for processing. With the fully connected layers, we combined these features together to create a model. Finally, we have an activation function such as softmax or sigmoid to classify the output.
[42]

# 5 Experiments

Different competitions use different subsets of images of the German database. In our challenge, there is a total of 34799 images belonging to 43 classes. In the classical experiment, 17051 pictures are used for training and 10440 for testing, and the remaining part for validations.
In our experiments, we used the HOG, Hue, and Haar-like feature extractions methods and CNN for classification. More in detail, our model is made up of four convolutional layers (conv2D), a max-pooling layer following every couple of conv2D and dropout. The detailed description of the model is given in Figure 14

```python
def myModel():
    no_Of_Filters = 60
    size_of_Filter = (5, 5)
    size_of_Filter2 = (3, 3)
    size_of_pool = (2, 2)
    no_Of_Nodes = 500
    model = Sequential()
    model.add((Conv2D(no_Of_Filters, size_of_Filter, input_shape=(imageDimesions[0], imageDimesions[1], 1),
                      activation='relu')))
    model.add((Conv2D(no_Of_Filters, size_of_Filter, activation='relu')))
    model.add(MaxPooling2D(pool_size=size_of_pool))

    model.add((Conv2D(no_Of_Filters // 2, size_of_Filter2, activation='relu')))
    model.add((Conv2D(no_Of_Filters // 2, size_of_Filter2, activation='relu')))
    model.add(MaxPooling2D(pool_size=size_of_pool))
    model.add(Dropout(0.5))

    model.add(Flatten())
    model.add(Dense(no_Of_Nodes, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(noOfClasses, activation='softmax'))

    model.compile(Adam(lr=0.001), loss='categorical_crossentropy', metrics=['accuracy'])
    return model
```

Figure 14: CNN Model

The performance of our code in the German database is 99.66%. In the newly introduced **TRaffic Sign** database 10 new classes for Turkish signs are added and a total of 2000 new pictures have been added to the data-set, 200 images per class. The best performance of our code in the TRaffic Sign database is 97.23%

The performances of our code with changing parameters in the laptop that has Intel Coffee Lake Core i7-9750H, 4GB GDDR5 nVIDIA GeForce GTX1650 128-Bit and 16GB DDR4L 1.2V 2666MHz SODIMM features can be shown as;

| Experiments | Batch Size | Steps Per Epoch | Epoch | Accuracy(%) |
|---|---|---|---|---|
| Experiment1 | 50 | 2000 | 30 | **97.23** |
| Experiment2 | 50 | 2000 | 20 | 96.67 |
| Experiment3 | 50 | 2000 | 10 | 96.15 |
| Experiment4 | 50 | 1500 | 30 | 97.08 |
| Experiment5 | 50 | 1500 | 20 | 96.90 |
| Experiment6 | 50 | 1500 | 10 | 95.10 |
| Experiment7 | 50 | 1000 | 30 | 95.84 |
| Experiment8 | 50 | 1000 | 20 | 96.23 |
| Experiment9 | 50 | 1000 | 10 | 94.19 |
| Experiment10 | 35 | 2000 | 30 | 96.16 |
| Experiment11 | 35 | 2000 | 20 | 96.37 |
| Experiment12 | 35 | 2000 | 10 | 94.61 |
| Experiment13 | 35 | 1500 | 30 | 96.33 |
| Experiment14 | 35 | 1500 | 20 | 95.65 |
| Experiment15 | 35 | 1500 | 10 | 94.16 |
| Experiment16 | 35 | 1000 | 30 | 95.76 |
| Experiment17 | 35 | 1000 | 20 | 94.63 |
| Experiment18 | 35 | 1000 | 10 | 93.29 |
| Experiment19 | 20 | 2000 | 30 | 95.10 |
| Experiment20 | 20 | 2000 | 20 | 94.95 |
| Experiment21 | 20 | 2000 | 10 | 91.58 |
| Experiment22 | 20 | 1500 | 30 | 94.34 |
| Experiment23 | 20 | 1500 | 20 | 93.88 |
| Experiment24 | 20 | 1500 | 10 | 92.00 |
| Experiment25 | 20 | 1000 | 30 | 92.95 |
| Experiment26 | 20 | 1000 | 20 | 92.05 |
| Experiment27 | 20 | 1000 | 10 | 89.58 |

Table 3: Experiments on the TRaffic Sign Classifier Dataset. Best performance is in black.

# 6 Design Methodology

There is an agile methodology in the progress of this project because of the cyclical and rapid progress of planning, working on improvement continuously, and keep going for each step. Also giving more weight to customer satisfaction to increase the efficiency of this project more

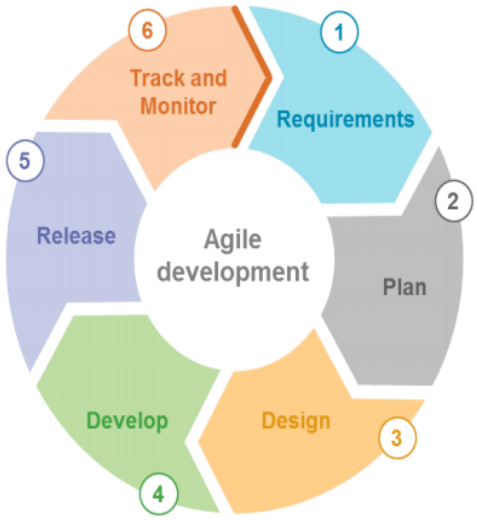


Figure 15: Figure of the methodology

At the request stage, the Team must collect information such as who will use the product and how to use it like features and susceptibility to the autonomous vehicle of the drivers in turkey. In the planning phase, it is aimed to divide the thought into smaller parts and then prioritize each feature and assign it to an iteration like firstly creating a database to feed the algorithm in the project. During the design phase, the Team should consider what the product or solution will look like. There is a camera on the vehicle and the scanned image is reflected on the vehicle's information screen with the meaning of the sign. In the development phase, the software required for the product is researched and compared with different algorithms, created, and then developed and applied. So the algorithm created in the project takes its final form after certain operations. At the release stage, the final version of the project will be tested by all the signs in the database enough to ensure that it is stage-ready to be presented to the customer. In the monitoring phase, it is monitored whether the project is working properly in the field, intervened if necessary, and information is collected from users for improvement like whether the reaction speed of the program is satisfied or not.

# 7 Risks

There are some problems rather than risks we can face while using this project like every other project.

## 7.1 Potential Risks

Most of the problems are caused by signs. There can be signs which are incomplete or cropped. Aging signs can cause problems because reading the writing on the signs would be harder. Also, poor image quality and humans making drawings on the signs are some problems that could cause misunderstanding while reading the signs.
Another problem we can face is the implementation of the program to the system of the cars. System failures and adaptation problems can be faced.

## 7.2 Risk and Change Management

The risks(in our case problems) can be minimized by having a large and various data-set which includes images for signs in different situations. Managing changes will be done by following the information about signs. For example, if there is a new sign created then it will be added to the data-set for our program to recognize it correctly.

# 8 Contributions

The main contribution of this work is to help drivers and self-driving cars to recognize the traffic signs and help them drive safely by following the traffic rules correctly. The correct recognition of traffic signs will result in more ordered traffic which results in fewer accidents happening. This paper introduces to the research community the first traffic sign database with Turkish sign also. So we can say that it creates a safer, healthier environment and the serenity of the people living in Turkey.

# 9 Conclusion

Traffic sign recognition is an important issue, which needs to be investigated more and more due to the recent improvement in technology. Currently, there is not any Turkish traffic sign database. This study introduces the first Turkish traffic sign database by extending the German Traffic Sign Recognition Database (GTSRB) with local Turkish traffic signs.
We used several feature extraction methods to extract discriminative info from images and a deep learning algorithm for classification. Our code reached the top performance of 99.66% on the German database and 97.23% on the newly introduced TRraffic Sign database.

**Realistic Constrains**
Realistic Constrains are the things we need to pay attention to while making a design.The designers should be able to identify, formulate and solve engineering problems with relevant standards within a certain budget and timeline.

**Social,Environmental and Economic Impact**
Needless to say, when it comes to driving, understanding of the environment at a precise level is very important. As a result, faster recognition of the objects in the environment of which traffic signs are an inevitable part determines how accurately a self-driving car or a human driver for that matter would actuate in the environment. These points show us some benefits in social, environmental, and economic ways. Not everybody knows what each sign means and this causes many problems such as traffic, accidents, and damage to the environment. For example correct and fast understanding of signs would result in more organized traffic, healthier environment and a result of this probability of paying money after accidents would decrease a lot and this would be beneficial for people in an economical way.
To summarize we can say that this project is very beneficial in social, environmental, and economic areas.

**Cost Analysis**

During the research and development of our project, we made no expenses so there is not really a design cost and there is no cost for the product because it is a program downloaded into the system of the self-driving car but there is the labor cost for our work in this project.
The average engineer salary is 5500 Turkish liras and the average work time for an engineer in Turkey is 40 hours a week. So the salary for one hour work of an engineer is 35 Turkish liras. For a time period of 6 months, each of us worked on both the research and development part of the project about 10 hours a week. So the labor cost of this project is 8400 Turkish liras for each of the members and 25.200 Turkish liras in total.

**Standards**

If we look at the engineering standards. We can see that there was an effort to make quality and efficient work on this project. The main advantage of software standards is efficiency. Because the entire project team will adhere to the same standards, any programmer will be able to work anywhere within the source code because the code is required to follow a predefined structure, which will make updating the code easier. The standards are made up of documents that define the development process, which begins with planning and analyzing project requirements, then moving on to implementing the design needed to build the model, and finally testing and integration, which is followed by necessary maintenance.

# References


[1] Y. Saadna and A. Behloul, "An overview of traffic sign detection and classification methods," *International Journal of Multimedia Information Retrieval*, vol. 6, pp. 1–18, 09 2017.

[2] M. Bazire and C. Tijus, "Understanding road signs," *Safety Science - SAF SCI*, vol. 47, pp. 1232–1240, 11 2009.

[3] J. Singh, M. Sahni, S. Bilquees, s. M. S. Khan, and I. Haq, "Reasons for road traffic accidents victims' perspective," *International Journal of Medical Science and Public Health*, vol. 5, p. 1, 01 2016.

[4] Y. Wang, S. Liu, X. Wu, and W. Shi, "Cavbench: A benchmark suite for connected and autonomous vehicles," in *2018 IEEE/ACM Symposium on Edge Computing (SEC)*, pp. 30–42, 2018.

[5] F. Pirotti, F. Sunar, and M. Piragnolo, "Benchmark of machine learning methods for classification of a sentinel-2 image," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLI-B7, pp. 335–340, 06 2016.

[6] "Cnn design for real-time traffic sign recognition," *Procedia Engineering*, vol. 201, pp. 718 – 725, 2017.

[7] . Arcos Garcia, J. Alvarez Garcia, and L. Soria Morillo, "Deep neural network for traffic sign recognition systems: An analysis of spatial transformers and stochastic optimisation methods," *Neural Networks*, vol. 99, 01 2018.

[8] R. Timofte, K. Zimmermann, and L. Van Gool, "Multi-view traffic sign detection, recognition, and 3d localisation," vol. 25, pp. 1–8, 12 2009.

[9] J. Zhang, W. Wang, C. Lu, J. Wang, and A. Kumar, "Lightweight deep network for traffic sign classification," *Annals of Telecommunications*, vol. 75, 07 2019.

[10] S. Aziz, M. Aroussi, and Y. FAKHRI, "Traffic sign recognition based on multi-feature fusion and elm classifier," *Procedia Computer Science*, vol. 127, pp. 146–153, 01 2018.

[11] F. Franzen, C. Yuan, and Z. Li, "Traffic sign recognition with neural networks in the frequency domain," *Journal of Physics: Conference Series*, vol. 1576, p. 012015, 06 2020.

[12] F. Zaklouta, B. Stanciulescu, and O. Hamdoun, "Traffic sign classification using k-d trees and random forests," in *The 2011 International Joint Conference on Neural Networks*, pp. 2151–2155, 2011.

[13] A. Ellahyani, M. Ansari, and I. El jaafari, "Traffic sign detection and recognition based on random forests," *Applied Soft Computing*, vol. 46, 02 2016.

[14] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in *The 2011 International Joint Conference on Neural Networks*, pp. 2809–2813, 2011.

[15] H. Habibi Aghdam., E. Jahani Heravi., and D. Puig., "A unified framework for coarse-to-fine recognition of traffic signs using bayesian network and visual attributes," in *Proceedings of the 10th International Conference on Computer Vision Theory and Applications - Volume 3: VISAPP, (VISIGRAPP 2015)*, pp. 87–96, INSTICC, SciTePress, 2015.

[16] S. K. Berkaya, H. Gunduz, O. Ozsen, C. Akinlar, and S. Gunal, "On circular traffic sign detection and recognition," *Expert Systems with Applications*, vol. 48, pp. 67–75, 2016.

[17] A. Ellahyani, M. Ansari, and I. El jaafari, "Traffic sign detection and recognition based on random forests, applied soft computing," *Applied Soft Computing*, vol. 46, 02 2016.

[18] X. He and B. Dai, "A new traffic signs classification approach based on local and global features extraction," *2016 6th International Conference on Information Communication and Management (ICICM)*, pp. 121–125, 2016.

[19] C. Li and C. Yang, "The research on traffic sign recognition based on deep learning," *2016 16th International Symposium on Communications and Information Technologies (ISCIT)*, pp. 156–161, 2016.

[20] "Hierarchical traffic sign recognition. in: Pacific rim conference on multimedia, springer, pp," pp. 200–209, 2016.

[21] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Networks*, vol. 32, pp. 323–332, 2012. Selected Papers from IJCNN 2011.

[22] H. Aghdam, E. J. Heravi, and D. Puig, "A practical approach for detection and classification of traffic signs using convolutional neural networks," *Robotics Auton. Syst.*, vol. 84, pp. 97–112, 2016.

[23] "A committee of neural networks for traffic sign classification. in: The 2011 international joint conference on neural networks (ijcnn), ieee," p. 19181921.

[24] S. Eickeler, M. Valdenegro, T. Werner, and M. Kieninger, "Future computer vision algorithms for traffic sign recognition systems," 2015.

[25] R. Qian, Y. Yue, F. Coenen, and B. Zhang, "Traffic sign recognition with convolutional neural network based on max pooling positions," *2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pp. 578–582, 2016.

[26] K. Xie, S. Ge, Q. Ye, and Z. Luo, "Traffic sign recognition based on attribute-refinement cascaded convolutional neural networks," in *17th Pacific-Rim Conference on Advances in Multimedia Information Processing - Volume 9916*, PCM 2016, (Berlin, Heidelberg), p. 201210, Springer-Verlag, 2016.

[27] "Fast traffic sign recognition using color segmentation and deep convolutional networks. in: International conference on advanced concepts for intelligent vision systems," p. 205216.

[28] Y. Zeng, X. Xu, Y. Fang, and K. Zhao, "Traffic sign recognition using extreme learning classifier with deep convolutional features," 2015.

[29] S. Tang and L.-L. Huang, "Traffic sign recognition using complementary features," in *Proceedings of the 2013 2nd IAPR Asian Conference on Pattern Recognition*, ACPR '13, (USA), p. 210214, IEEE Computer Society, 2013.

[30] H. Aghdam, E. J. Heravi, and D. Puig, "A practical approach for detection and classification of traffic signs using convolutional neural networks," *Robotics Auton. Syst.*, vol. 84, pp. 97–112, 2016.

[31] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Networks*, vol. 32, pp. 323 – 332, 2012. Selected Papers from IJCNN 2011.

[32] Z. Huang, Y. Yu, J. Gu, and H. Liu, "An efficient method for traffic sign recognition based on extreme learning machine," *IEEE Transactions on Cybernetics*, vol. 47, no. 4, pp. 920–933, 2017.

[33] A. Youssef, D. Albani, D. Nardi, and D. Bloisi, "Fast traffic sign recognition using color segmentation and deep convolutional networks," vol. 10016, 10 2016.

[34] M. J. A. Patwary, S. Parvin, and S. Akter, "Significant hog-histogram of oriented gradient feature selection for human detection," *International Journal of Computer Applications*, vol. 132, pp. 20–24, 12 2015.

[35] X. Fu and W. Wei, "Centralized binary patterns embedded with image euclidean distance for facial expression recognition," (USA), IEEE Computer Society, 2008.

[36] R. Samad and H. Sawada, "Edge-based facial feature extraction using gabor wavelet and convolution filters," 06 2011.

[37] Y. Liu, H. Zhang, H. Guo, and N. Xiong, "A fast-brisk feature detector with depth information," *Sensors*, vol. 18, p. 3908, 11 2018.

[38] M. Gk, "Makne renmes yntemler le akademk baarinin tahmn edlmes," *Gazi niversitesi Fen Bilimleri Dergisi Part C: Tasarm ve Teknoloji*, vol. 5, pp. 139 – 148, 2017.

[39] . Takn and G. Emel, "Genetik algoritmalar ve uygulama alanlar," *Uluda niversitesi BF Dergisi*, vol. 21, pp. 129–152, 02 2002.

[40] T. Chengsheng, L. Huacheng, and X. Bing, "Adaboost typical algorithm and its application research," *MATEC Web of Conferences*, vol. 139, p. 00222, 01 2017.

[41] M. Awad and R. Khanna, *Support Vector Machines for Classification*, pp. 39–66. 01 2015.

[42] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *ArXiv e-prints*, 11 2015.