# Pothole detection - Machine Learning Binary Classification

1st Jacek Piszczek
*Computer Science*
*Wroclaw University of Science and Technology*
Wroclaw, Poland

2nd Damodar Łepski
*Computer Science*
*Wroclaw University of Science and Technology*
Wroclaw, Poland

***Abstract*—Good condition of the roads are important thing in this times. There is a lot of people which travel by car or bus. Roads always are intensive used by heavy vehicles which causes the road damage. The potholes are main problem of the roads which can cause also car damage and faster utilisation, which can cost a lot of money. One of the solution is to repair the roads as fast as possible. But as known, in practice this is impossible to keep all roads in well condition. This paper provides the study of machine learning pothole detection, which can be treated as basic mechanism for implementation pothole avoiding car system, which for example will detect potholes from a long distance by advanced optical sensors and cameras. In this study there is built the system that can detect if road image has a potholes or not.**

***Index Terms*—machine learning, pothole, detection, roads, images, computer vision**

## I. Introduction

Roads play a crucial role in supporting an human society and economy. Road conditions are important, because people always traveling, moving goods and do a lot of other activities. Modern solutions in communication like smart and intelligent cars have to be equipped with special systems which measures road quality. It is important thing which should be considered during production safe autonomous transport and intelligent cars. Best solution is to manual check the pothole exists, but it is impossible during the travel by car. People can drive fast and it can be dangerous for them and passengers. Intelligent pothole detector implemented in modern car can prevent accidents connected with road damages by alerting the driver about upcoming pothole and give the driver time for reaction. One technology which can handle with pothole and road damage detection is Computer Vision. There is a lot of research about pothole detection using image processing to detect and estimate the surface of the pothole, using cameras or laser sensors [2].

In this research there are provided machine learning binary classification of potholes. Detection will be done using sets of images which will be specially prepared to learn and predict classifiers to provide best results.

To optimise work, Python technology and Jupiter Lab framework was used. JupyterLab is a web-based interactive development environment for Jupyter notebooks, code, and data. JupyterLab is flexible: configure and arrange the user interface to support a wide range of workflows in data science, scientific computing, and machine learning. JupyterLab is extensible and modular: write plugins that add new components and integrate with existing ones [3].

To examine images and achieve best performance, the Histogram of Oriented gradients was used. It is a feature descriptor of image objects.

Compute a Histogram of Oriented Gradients [9] (HOG) by

- (optional) global image normalisation
- computing the gradient image in x and y
- computing gradient histograms
- normalising across blocks
- flattening into a feature vector

Sample HOG image:

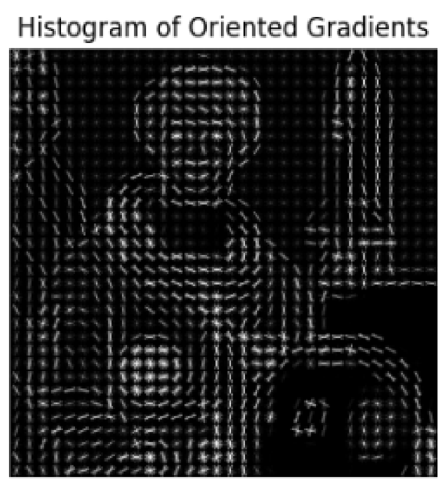


Fig. 1. Example image processed by HOG image algorithm [9]

In project it was implemented and used from *skimage* python library.

## II. Related work

Authors in [2] prepared the model of pothole recognition system of car using raspberry pi. They mounted wireless portable camera at the front of the car, computer inside. The classification algorhitm consists of the dataset which used 1000 images separated for train and test by the ratio of 70:30, so 700 image been used for training data and the 300 image left been used for testing data. This system is sending the data to the global SEMAR system IoT platform with attitude and longitude of the pothole. Then it can be visualised on the map where pothole is. The deep learning framework that being used in this research is Tensorflow [4].

Another interesting work at el. [5] provides the study of detecting the potholes and cracks in the road using Support Vector Machine, Artifictial Neural Netwok and Random Forest algorithms. Dataset was prepared from 500m asphalt road located in Shirihezi City in China. The road is in bad condition because of transportation heavy trucks and tractors. Data for training and validating were extracted, and total 1760 pieces of sample image objects containing 538 potholes, 753 cracks and 469 nondistressed pavements (305 damage-free pavement and 164 yellow traffic lines) [5]. The overall accuracy of the classification of cracks, potholes, and nondistressed pavements is 98.3%.

Interesting study was prepared in [6], where authors used Neural Network (CNN) to detect potholes. they prepared 500 testing images from serveral different places with various variations like in wet, dry and shady conditions. The focused on deep learning and their model consist of 4 convolutional layers. At the end of convolutional and pooling layers, networks generally use fully-connected layers in which each pixel is considered as a separate neuron just like a regular neural network. The last fully connected layer will contain as many neuron just like a regular neural network. Proposal solution are compared with Support Vector Machine algorithm and outperform it. SVM achieves 88% accuracy and 86% precision, where proposed neural network acieves 99.8% accuracy and 100% precision.

In this paper is prepared the study of image recognition of potholes gathered from the internet [7]. It provides binary classification of pothole images to decide is pothole detected on image. To carry out the study, tools from sklearn [8] was used. Specially prepared images with histogram of oriented gradients processing and features extracted are used for learning and predict the results.

## III. Methodology

A lot of studies with computer vision have been proposed above, but in this one images with real potholes will be examined. Images are resized for equality of feature extraction.

To evaluate the study, images are gathered and stored into two folders: Image with potholes and just roads without potholes. Each of them are scaled and preprocessed to gather Histogram of oriented gradients representation. From HOG image the desirable features are extracted and used in machine learning fitting three classifiers. Results are compared with different metrics.

Plan of carry out the studies:

- Collect two types of road: with potholes, and without.
- Solve problem of unbalanced data to increase three times more data:
  - flip images horizontally,
  - rotate images by 180 degree,
- Calculate Histogram of Oriented Gradients,
- Label the data,
- Store extracted HOG feature vectors in DataFrame,
- Divide prepared dataset to train and test using Stratified K-Fold validation,
- Train and test five classifiers:
  - Support Vector Machine,
  - Gaussian Naive Bayes,
  - Stochastic Gradient Tree,
  - K Nearest Neighbours (not suitable, but tested for comparison)
  - Decision tree (not suitable, but tested for comparison)
- Compare results using metrics:
  - Accuracy,
  - Precision.

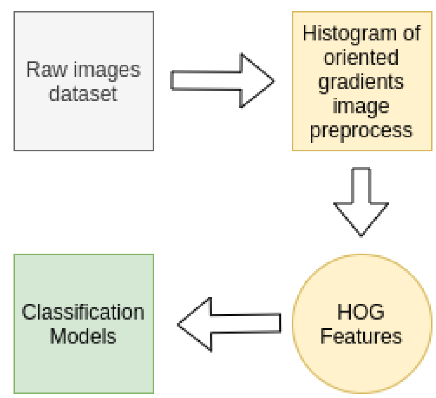


Fig. 2. Diagram representing steps to examine classification [9]

The given result presents how much images are detected as a roads with potholes.

## IV. Algorithms

In this section are presented used algorithms needed to evaluate the study.

### A. *Histogram of Oriented Gradients*

In detail, this algorithm in the first stage applies an optional global image normalisation equalisation that is designed to reduce the influence of illumination effects. The second stage computes first order image gradients. These capture contour, silhouette and some texture information, while providing further resistance to illumination variations. The locally dominant color channel is used, which provides color invariance to a large extent. The third stage aims to produce an encoding that is sensitive to local image content while remaining resistant to small changes in pose or appearance. The adopted method pools gradient orientation information locally in the same way as the SIFT 2 feature.

The image window is divided into small spatial regions, called "cells". For each cell we accumulate a local 1-D histogram of gradient or edge orientations over all the pixels in the cell. This combined cell-level 1-D histogram forms the

Fig. 3. Pothole image.

Fig. 4. Histogram of oriented gradients of pothole image.

basic “orientation histogram” representation. The fourth stage computes normalisation, which takes local groups of cells and contrast normalises their overall responses before passing to next stage. The final step collects the HOG descriptors from all blocks of a dense overlapping grid of blocks covering the detection window into a combined feature vector for use in the window classifier [9]. Example pothole image at figure 3 after applying HOG algorithm results as on figure 4.

### *B. Support Vector Machine*

Support Vector Machine offers very high accuracy compared to other classifiers. It is used in a lot of applications, such as face recognition, movement detection, spam, emails, genres classification or even handwriting recognition. SVM constructs a hyperplane in multidimensional space to separate different classes. Support vectors are the data points, which are closest to the hyperplane. A margin is a gap between the two lines on the closest class points.

The main objective is to segregate the given dataset in the best possible way. The distance between the either nearest points is known as the margin [10].

### *C. Gaussian Naive Bayes*

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem.

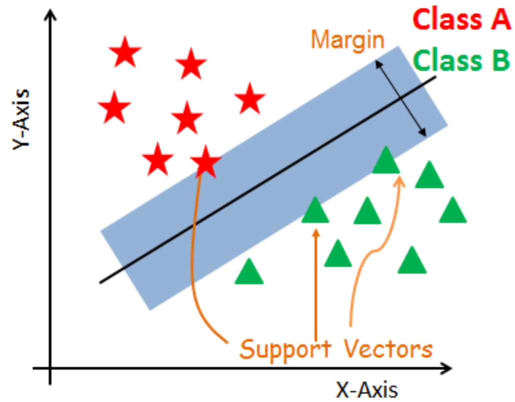


Fig. 5. Support vector machine [10]

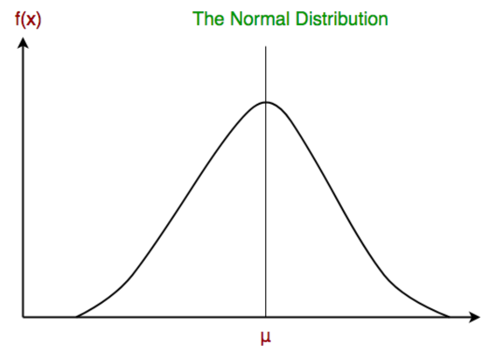


Fig. 6. The normal distribution [11]

In Gaussian Naive Bayes, continuous values associated with each feature are assumed to be distributed according to a Gaussian distribution [11]. A Gaussian distribution is also called Normal distribution. When plotted, it gives a bell shaped curve which is symmetric about the mean of the feature values as shown at figure 7.

### *D. Stochastic Gradient Descent*

Stochastic Gradient Descent is a very popular algorithm used in Machine learning techniques, e.g in Neural Networks. SGD has been successfully applied to large-scale and sparse machine learning problems often encountered in text classification and natural language processing. Given that the data is sparse, the classifiers in this module easily scale to problems with more than $10\hat{5}$ training examples and more than $10\hat{5}$ features [8].

### *E. k Nearest Neighbours*

This classifier are not good choice in computer vision recognition tasks, but it will be taken into consideration and comparison as well as Decision tree to see how big is the difference with potentially better classifiers mentioned above.

k-NN algorithm implements the k-nearest neighbors vote. KNN algorithm is used for both classification and regression

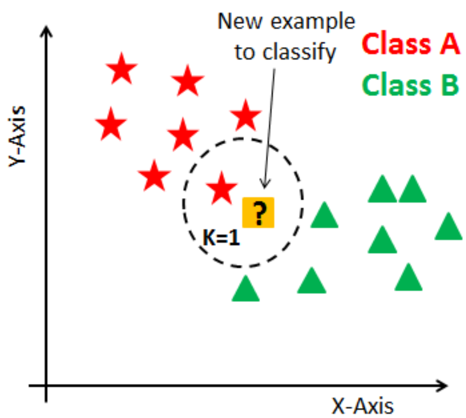


Fig. 7. How kNN classifier works [10]

problems and is based on feature similarity approach. In KNN, K is the number of nearest neighbors. The number of neighbors is the core deciding factor. K is generally an odd number if the number of classes is 2. When K=1, then the algorithm is known as the nearest neighbor algorithm [10].

### F. Decision Tree

Decision Tree algorithm is capable of performing multi-class classification on a dataset. A decision tree is a flowchart-like tree structure where an internal node represents feature(or attribute), the branch represents a decision rule, and each leaf node represents the outcome [10].

### G. Accuracy

It is the number of correctly predicted samples over all predicted samples.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

### H. Precision

Precision is defined as the number of positively predicted samples divided by the total number of positively predicted sample. It is also called the Positive Predictive Value (PPV).

$$precision = \frac{TP}{TP + FP} \tag{2}$$

Precision could be the determinant of a classifiers exactness - a large number of False Positives is detected by a low precision. [12]

## V. Dataset preparation

In this section data processing are described. Final result of process images is fulfilled *DataFrame* object by labeled image features.

### A. Image collection

In order to start computer vision with processing images, they are stored in two folders. First folder consists of images from public repository build from images scrapped from Google [7]. Second folder consists of images of nice roads without potholes downloaded manually from Google. All images are different size. Dataset consists of 148 images with potholes and 50 nice roads.

### B. Scale transformations

In this step each image is scaled and resized to 640x480 resolution to extract the same amount of features. Next in order to avoid unbalanced data, the smaller nice roads image set are transformed twice:

- rotated by 180 degree,
- flipped horizontally.

Each image was transformed twice as above in order to achieve three times more elements in nice roads dataset. Finally there is 148 images with potholes and 150 images of roads without potholes.

### C. Historgam of oriented gradients

In this step each image are processed by algorithm which compute Histogram of Oriented Gradients. This algorithm returns extracted features of an image. This features can be used for feed classifiers. Feature vector of an certain image has 38400 values. This amount depends of image resolution.

### D. DataFrame object

Collected feature vectors from each image are connected in one dataset. Pothole vectors are labeled as (1), nice road vectors as (0). Data with labels are moved do DataFrame object and shuffled.

## VI. Experiment Result

Testing was carry out with Stratified K-Fold validation in order to increase randomness of the train and test sets. Train - test ratio is 2:1. Whole dataset has 298 elements, which leads to train subset have 199 elements and test subset have 99 elements.

Three classifiers are trained: *Support Vector Machine*, *Gaussian Naive Bayes* and *Stochastic Gradient Descent*. Each of them results with pretty good accuracy and precision. Results are presented in table I.

TABLE I
Results of pothole classification

| Classifier | Accuracy | Prediction |
|---|---|---|
| Support Vector Machine | 78% | 79% |
| Gaussian Naive Bayes | 79% | 78% |
| Stochastic Gradient Descent | 75% | 74% |
| K Nearest Neighbours | 51% | 50% |
| Decision Tree | 62% | 61% |

## VII. Conclusions and future work

This paper examined the study of computer vision, machine learning pothole detection using real roads image data. Prediction results are at acceptable level and it can be concluded that the system of detecting potholes working pretty well. After rescaling the images, some of them lost small amount of intensity, but still was pretty clear and visible. Dataset was preprocessed and divided by K-Fold validation with k=3, to keep 2:1 train:test ratio. There was three folds, which allows to perform study with better randomization of data.

Best results achieves SVM and GNB, which was almost the same level above 80%. SGD Classifier fall a little bit down, but still acceptable with accuracy with 75% and 74% precision. As we can see *KNN* and *Decision Tree* achieve poor score, kNN is around 50% which means it is not suitable to this study at all. Decision Tree got little more than 60% which is also poor result. It proves the SVM and GNB are the most suitable algorithms in Computer Vision machine learning image recognition.

Future work can be based on analyse real time video captured from camera and looking for potholes. That systems can be integrated with modern intelligent cars, which can warn driver about the danger and possible damaged road in front of car.

## References


[1] Land Transportation Statistic 2016. Badan Pusat Statistik. 2016.

[2] A. Rasyid et al., "Pothole Visual Detection using Machine Learning Method integrated with Internet of Thing Video Streaming Platform, 2019 International Electronics Symposium (IES), Surabaya, Indonesia, 2019."

[3] Project Jupyter, JupyterLab 1.0: Jupyter's Next-Generation Notebook Interface *https://jupyter.org/*, 2020

[4] Tensoflow, *https://www.tensorflow.org/*, 2020

[5] Y. Pan, X. Zhang, G. Cervone, L. Yang, Detection of Asphalt Pavement Potholes and Cracks, October 2018

[6] V. Pereira, S. Tamura, S. Hayamizu and H. Fukai, "A Deep Learning-Based Approach for Road Pothole Detection in Timor Leste," 2018 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), Singapore, 2018, pp. 279-284, doi: 10.1109/SOLI.2018.8476795.

[7] S. Patel, Pothole image dataset, Kaggle, *https://www.kaggle.com/sachinpatel21/pothole-image-dataset*, 2019

[8] scikit-learn, Machine learning in python, *https://scikit-learn.org/*, 2020

[9] Histogram of Oriented Gradients, *https://scikit-image.org/*, 2020

[10] Support Vector Machines with Scikit-learn, *https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python*, 2019

[11] Naive Bayes Classifiers, *https://www.geeksforgeeks.org/naive-bayes-classifiers/*, 2020

[12] A. Mishra, "Metrics to Evaluate your Machine Learning Algorithm" https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234 Feb 24, 2018