UDC 519.6
https://doi.org/10.31891/csit-2023-1-7

Lesia MOCHURAD, Andrii ILKIV
Lviv Polytechnic National University
Oleksandr KRAVCHENKO
National Technical University «Kharkiv Polytechnic Institute»

A NEW INFORMATION SYSTEM FOR ROAD SURFACE CONDITION CLASSIFICATION USING MACHINE LEARNING METHODS AND PARALLEL CALCULATION

Modern information systems are increasingly used in various areas of our life. One of these is the quality control of the condition of the road surface in order to carry out repair work on time if necessary. The machine learning method can facilitate the control process, which was demonstrated in this work.

Analyzing the road surface condition using image classification requires much pre-classified data and decent computing power. As the modern need for proper quality control of the road surface is high, it is possible to analyze using sensor-recorded data in tabular form and machine learning methods, which should show high accuracy of the classification results. Development and research of an information system for classifying the condition of the road surface were described in this paper, including ways for optimizing similar approaches and improving the results obtained through the use of a greater number of features, in particular, taking into account not only the speed indicators at the given time of the car's movement but also the performance indicators of internal combustion engine. As a result, an information system was developed that classifies the road surface condition using features obtained from various types of sensors and recorded in tabular form. Machine learning methods such as Random Forest, Decision Tree, Support Vector Method, and AutoML library were used to compare accuracy results using a large set of artificial intelligence methods. The best results were obtained using the Random Forest ensemble machine learning method. The analysis of the classifier according to various parameters was carried out, and a search for the best hyperparameters was performed. At the same time, achieving a 91.9% accuracy of road surface condition classification was possible. Parallel calculations were used during model training. As a result, training time was decreased by 5 times with the use of the CPU and by 51 times with the help of the GPU.

Keywords: road condition classification, Random Forest, reference vectors method, decision tree, CUDA technology.

Леся МОЧУРАД, Андрій ІЛЬКІВ
Національний університет «Львівська політехніка»
Олександр КРАВЧЕНКО
Національний технічний університет «Харківський політехнічний інститут»

НОВА ІНФОРМАЦІЙНА СИСТЕМА ДЛЯ КЛАСИФІКАЦІЇ СТАНУ ДОРОЖНЬОГО ПОКРИТТЯ ЗА ДОПОМОГОЮ МЕТОДІВ МАШИННОГО НАВЧАННЯ ТА ПАРАЛЕЛЬНИХ ОБЧИСЛЕНЬ

Аналіз стану дорожнього покриття при класичному підході розпізнавання особливостей із використання зображень потребує великої кількості даних з попередньо підготовленим описом та обчислювальних потужностей. Враховуючи сучасні потреби своєчасного контролю якості покриття дорожніх шляхів, аналіз можливо спростити з використанням показників записаних у табличному виглядів та методів машинного навчання, які у сукупності повинні показати задовільну точність результатів. У роботі було обґрунтовано доцільність розробки та дослідження інформаційної системи класифікації стану дорожнього покриття, а також визначено як ключовий напрямок оптимізацію аналогічних підходів та покращення отриманих результатів шляхом використання більшої кількості ознак, зокрема врахування не лише швидкісних показників в момент руху автомобіля, але й показників роботи двигуна внутрішнього згоряння, яке напряму пов'язана з можливістю здійснювати рух. У результаті розроблено інформаційної системи, що класифікує стан дорожнього покриття за ознаками отриманими з різного виду датчиків та записаних у табличному вигляді. Використано методи машинного навчання такі як Випадковий Ліс, Дерево Рішень, Метод опорних векторів та бібліотеку AutoML, яка дозволила провести порівняння точності результатів з використанням великого набору методів штучного інтелекту. Найкращі результати вдалось отримати за допомогою ансамблевого методу машинного навчання Rendom Forest. Проведено аналіз роботи класифікатора при різних параметрах та виконано пошук найкращих гіперпараметрів. При цьому вдалось досягти точності класифікації стану дорожнього покриття рівній 91.9 %. Для можливості прискорення отримання рішення було застосовано паралельні обчислення при тренуванні моделі. В результаті отримано показник прискорення у 5 разів з використанням CPU та у 51 раз з використанням GPU.

Ключові слова: класифікація стану дорожнього покриття, випадковий ліс, метод опорних векторів, дерево рішень, технологія CUDA.

Introduction

When a car passes a surface of different types and conditions, changes in the speed, trajectory, and smoothness of the vehicle's movement occur. Thanks to the use of motion sensors, and sensors of the internal combustion engine, it is possible to classify the type of road surface, driving style, and road traffic [1]. Target variable classification [2] based on data gathered during the vehicle movement allows to combine all of this into effective approach for fast classification to determine the condition of the road surface at a certain moment of the time of operation of the sensors and additionally geolocation of the vehicle at the same moment.

There are two popular approaches to classifying the condition of a road surface:

- Use of tabular data obtained while driving the car;
- Use of overlay images.

Road surface condition classification using images will guarantee more accurate predictions [3] but requires a large amount of image data. Collecting and forming a suitable dataset is much easier in case of classification based on the use of tabular data rather than image-based classification. The approach of data collection using the smartphone gyroscope is popular [4–6]. However, the advantage of the developed information system will be the use of tabular data obtained from the accelerometer and the electronic control unit of the car's internal combustion engine in combination with classifiers, the best of which will be selected using the AutoML library for the high-level Python programming language and optimized using parallel computing algorithms [7]. The use of several existing machine learning algorithms to solve the given task will provide an opportunity to conduct a comparative analysis with the ability to determine the most optimal approach.

Therefore, the purpose of this paper is to create an information system for classifying the condition of the road surface using input data obtained from sensors installed in the car based on machine learning algorithms, improving the results of the system by using parallel computing and conducting a comparative analysis of the results of the system when applying hyperparameter settings classifiers.

The object of the study is the information system of classification or prediction of the condition of the road surface based on tabular data using machine learning algorithms.

The subject of the research will be the use of machine learning methods such as Random Forest, Decision Tree, Support Vector Method, and the AutoML library, which will allow comparison of the accuracy of results using a large set of artificial intelligence methods.

A large number of publications and studies in the field of road pavement condition classification confirm the relevance of the chosen topic [8, 9]. Since pavement maintenance is an important task for maintaining the stable operation of urban infrastructure, automating the identification of areas that require immediate maintenance or are in unsatisfactory condition will facilitate the appropriate analysis.

Related works

During the analysis of related works, modern approaches to solving the problem of road surface condition classification and the optimal application of machine learning methods to solve this problem were considered.

The solution to the problem of road surface condition classification using classifiers are described in [10]. In this work, a study was carried out, the purpose of which was to determine the features from the data set that most affect the accuracy of the obtained results and to carry out classification using the analysis of vehicle vibrations. When analyzing the input data, there is a high probability of anomalies due to the sensitivity of the data collection sensors. Carrying out a sensitivity analysis of the received readings will make it possible to increase the weight of some features compared to others based on their value when obtaining classification results. Using the analysis of the time series of car oscillations, the authors determined the average deviation of the signs. They conducted an analysis based on frequency oscillations from the corresponding sensors. The impact of various features used in classification on the accuracy of the selected classifiers RF, DT, and MLP was also investigated. During the analysis, two sets of data were used: data obtained from sensors and simulated data. The best accuracy of 87% was obtained precisely for the simulation data set, which may indicate a linear relationship between features and affect accuracy.

A similar problem was considered by the paper's authors [4]. The study of road surface anomalies is focused on the data obtained from the smartphone's accelerometer, gyroscope, and GPS data. The authors of this scientific work suggest using a smartphone when analyzing the road surface to determine its condition. Indicators of changes in acceleration and vibration of the smartphone were analyzed, which made it possible to classify each part of the road surface with the corresponding geolocation. The results were obtained using machine learning algorithms and a multilayer neural network. At the same time, the authors tried to combine the results of the smartphone's accelerometer and gyroscope with the obtained road surface images. Since assigning labels to each image in order to further use them in a complex classification requires a large amount of human labor, the authors investigated the possibility of automatic classification and assignment of labels to images for further work with convolutional neural networks. The disadvantage of this work is that with a data set of 1010 rows, it takes 70 seconds to train the model with an accuracy of 88% when using the SVM and RF algorithms.

In paper [11], the authors tried to analyze the state of the road surface and search for its anomalies using a geoinformation system. Only Ox, Oy, and Oz acceleration indicators were used for the analysis. The classification of the condition of the road surface was carried out by an impulse neural network. As an aproach was chosen the ensemble learning of the classification model for the simultaneous determination of the condition of both roads with and without asphalt. The result of the work of the geoinformation system proposed in this article was 99.9% in determining the type of road path and 99.8% during the classification of the state of this path. Since the training dataset used a dataset of 2835 rows to train the model for pavement type detection and a dataset of 4300 rows to train the pavement condition classification model, the model could have been overtrained. If we compare the results with the work [10], then the accuracy of predictions should differ by a factor of two between actual and simulated data.

The optimization of execution time by using algorithms of parallel calculations was considered by the authors of the work [12]. The authors proposed to speed up the training of the RF model by performing parallel training on four threads simultaneously. As a result, it was experimentally proven that execution on four threads and with a given parameter of the number of trees equal to 100 training took 77 seconds, which is twice as fast as training using only two threads. However, work [2] shows that using GPU for parallelization is 83.4 times faster than parallelization using 8 CPU threads.

Improving the running time of model training with GPU was considered in [13]. Even modern equipment can only sometimes cope with the tasks when working with large data sets. The paper proposes an approach that includes the application of a reduction algorithm based on the stochastic distribution of neighbors in the data set in combination with a new approach to calculating the KNN graph using GPU. This approach showed a speedup of 460%.

The advantage of our proposed approach over the methods mentioned above will be taking into account not only the speed indicators at the time of the car's movement but also the performance indicators of the internal combustion engine, which are directly related to the ability to move. Also, using parallel computing algorithms will provide an opportunity to obtain classification results faster without losing the accuracy of these predictions.

Methodology

For achieving aim of this paper next items should be covered:

- preparation for data set classification;
- classification using parallel computing algorithms;
- visualize and compare the obtained results.

Classification of the road surface condition will be done using the data set [14], which consists of 17 features and 24,957 records. The following 17 features are divided into categories depending on the type of sensor from which the data was received:

Data obtained from the accelerometer:

- AltitudeVariation – change in height during movement
- VerticalAcceleration – vertical acceleration of the car
- LongitudinalAcceleration – horizontal acceleration of the car

Data obtained from the car's motion sensors:

- VehicleSpeedAverage – the average speed of the car
- VehicleSpeedVariance – the difference between the speed of the car and speed limit
- VehicleSpeedVariation – average change in vehicle speed
- VehicleSpeedInstantaneous – the speed of the car at a given moment of time

Data obtained from the car's electronic control unit:

- EngineLoad
- EngineCoolantTemperature – engine cooling system temperature
- ManifoldAbsolutePressure – absolute pressure in the exhaust manifold
- EngineRPM – engine revolutions per minute
- MassAirFlow – air flow mass
- IntakeAirTemperature – air temperature at the inlet to the fuel combustion valves
- FuelConsumptionAverage – average fuel consumption by car

Target features:

- roadSurface – condition of the road surface
- traffic – car traffic at a given moment of time
- drivingStyle – the style of driving a car at a given moment in time

As a result of the classification by several target variables, it is possible to achieve a higher accuracy of the classification since the performance indicators of the engine will be used for the analysis, which, depending on traffic jams or aggressive driving style, will increase the load. If these factors are taken into account during classification, accuracy of machine learning algorithms results will be increased [15].

When preparing the data set, possible anomalies that can affect the results' quality must be considered. Such anomalies can be both missing values and values that strongly deviate from the average. To eliminate this problem, we need to normalize the data set. The z-score and min-max methods will be used [16]. RF, SVM, and DT classifiers from the sklearn library for the high-level Python programming language will be the main tools for classification. The accuracy and performance of the trained models will be evaluated using F1-Score, Precision-Recall, MSE, and ROC-AUC metrics. To analyze the obtained results, the matplotlib library will be used to visualize the results and present them as graphs of functions. To optimize the training time of the classification models, training using multiple CPU and GPU threads will be applied to the algorithm that showed the highest accuracy – RF.

Ensemble machine learning methods will be applied to solve the problem of pavement condition classification, such as the RF regression method, which is based on the regression construction of numerous decision

trees during model training, and the SVM method, which performs classification by building models called support vector networks [17].

When training an RF model, the training sample will be divided into random subsamples with replication. In this case, some records may enter the sample several times, and others not. The next step will be to randomly select several features, after which the construction of decision trees will begin, which will classify this subsample and only on the part of the randomly selected features. The most valuable features are selected using the Gini criterion.

In the case of training the RF model on multiple streams simultaneously, the bootstrap aggregation will be applied. Since the sample will be divided during training, the summation of the nearest neighboring classifiers will allow for obtaining a model at the end, which will be a collection of classifiers performed on separate streams[18].

Trees are built until the entire subsample is passed. Thus, not only the accuracy of the classification will depend on the number of built trees, but also the time required for training the model will increase. The optimal number of trees is selected in such a way as to minimize the error of the classifier during the analysis.

An important step in training the RF model is to evaluate the importance of each feature of the training sample. In order to assess the importance of each sample parameter, a model is trained on this set with error estimation and parameter shuffling at the end of each iteration. The importance of the parameter is estimated by calculating the errors before and after the shuffle. Error-values are normalized, taking into account the standard deviation.

To evaluate the performance of classifiers, such metrics as:

- F1 score,
- ROC AUC,
- Precision Recall,
- Accuracy.

Here, accuracy is a percentage value that indicates the number of correctly classified target features. It is calculated according to the following formula:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN},$$

where TP – positive samples that are correctly classified,

FN – negative samples that are incorrectly classified,

FP – positive samples that are incorrectly classified,

TN – negative samples that are correctly classified.

To measure the rate of acceleration and efficiency during the parallel execution of the classification model's training process, the classifier's computational complexity needs to be determined. For RF, it is calculated by the formula: $O(n * \log(n) * d * k,$

Where n – the number of records in the data set;

k – the number of trees during training;

d – the size of the training data set.

In order to calculate the value of the theoretical acceleration indicator, it is necessary to find the ratio of computational complexity when performing training consecutively and in parallel according to the formula:

$$S_p(n) = \frac{T_1(n)}{T_p(n)},$$

where p – number of threads;

$T_1(n)$ – time complexity of sequential execution of the algorithm;

$T_p(n)$ – time complexity of parallel execution of the algorithm for p threads.

The efficiency indicator is calculated according to the formula: $E_p(n) = \frac{S_p(n)}{p} = \frac{O(n*\log(n)*d*k}{p*O(\frac{n}{p}*\log(\frac{n}{p})*d*k)}$.

The performance indicator cannot be calculated for training performed on the GPU because the GPU has a complex structure of cores called pipelines.

Experiments

The training of classification models was carried out on a machine with an Intel core I7-6700 CPU with 4 cores and 8 threads. NVIDIA GT970M video card with 2 gigabytes of video memory was used to train the model using the GPU.

The pavement condition classification study was conducted on the previously described data set split 80 : 20 into training and test data sets.

The definition of the classifier with the highest accuracy was carried out using the AutoML software library. As an input, it receives a training data set, after which it analyzes features, their interdependence, and the type of the target variable. The result of the work of the AutoML library is a rating table of the accuracy of the obtained classifiers,

which were determined to be the most suitable for classification. Each classifier trained the model with standard parameters. Fig. 1 shows the rating table of classifiers.

model_id	rank	ensemble_weight	type	cost	duration
2	1	0.52	random_forest	0.066227	44.534895
3	2	0.08	extra_trees	0.122478	34.161803
8	3	0.08	mlp	0.148524	51.222872
11	4	0.06	mlp	0.277987	31.608544
10	5	0.16	multinomial_nb	0.409596	2.396460
7	6	0.10	bernoulli_nb	0.545404	2.334953

Fig. 1 Rating table of classification quality of trained models

The RF classifier showed the best accuracy, with an accuracy of 85%, training the model in 44.5 seconds. Training and searching for the best model among the available ones took 6 minutes. The following metric results were obtained (see Fig. 2):

	precision	recall	f1-score	support
FullOfHolesCondition	0.98	0.96	0.97	636
SmoothCondition	0.99	0.98	0.99	3012
UnevenCondition	0.99	0.94	0.97	1344
HighCongestionCondition	1.00	0.89	0.94	647
LowCongestionCondition	0.99	0.99	0.99	3701
NormalCongestionCondition	0.98	0.91	0.95	644
AggressiveStyle	0.87	0.48	0.61	555
EvenPaceStyle	0.95	0.98	0.97	4437
micro avg	0.98	0.95	0.96	14976
macro avg	0.97	0.89	0.92	14976
weighted avg	0.97	0.95	0.96	14976
samples avg	0.98	0.95	0.96	14976

Fig. 2 Performance metrics of the best RF classifier model

Since the model was trained using the default value of the number of trees, depth, and the minimum number of tree leaves, a high accuracy value could not be achieved. To improve the results, it is necessary to select hyperparameters that could increase the accuracy of the classifier. The RandomizedSearchCV function from the sklearn library will be used for this.

```
[Parallel(n_jobs=-1)]: Done 150 out of 150 | elapsed: 56.2min finished
{'n_estimators': 307, 'min_samples_split': 20, 'min_samples_leaf': 4,
 'max_features': 'sqrt', 'max_depth': 90, 'criterion': 'gini', 'bootstrap': True}
Time to get best hyperparameters = 3394.156 seconds
```

Fig. 3 The result of the hyperparameter selection function

As we can see from Fig 3, the applied function selected the most suitable among many possible parameters, so let us build a new tree using these hyperparameters.

	precision	recall	f1-score	support
FullOfHolesCondition	0.99	0.97	0.98	632
SmoothCondition	0.99	0.99	0.99	3062
UnevenCondition	0.99	0.98	0.98	1298
HighCongestionCondition	0.99	0.96	0.97	588
LowCongestionCondition	0.99	1.00	0.99	3781
NormalCongestionCondition	1.00	0.94	0.97	623
AggressiveStyle	0.92	0.58	0.71	589
EvenPaceStyle	0.95	0.99	0.97	4403
micro avg	0.98	0.97	0.97	14976
macro avg	0.98	0.93	0.95	14976
weighted avg	0.98	0.97	0.97	14976
samples avg	0.98	0.97	0.97	14976

```
Model training time is  252.8594033718109
Model Classification Accuracy =  0.9186698717948718
```

Fig. 4 Results of RF classifier training using the obtained parameters

The training result is a model with a classification accuracy of 91.8%, which is a satisfactory accuracy when using the classifier. However, the time required to train the RF model is 252 seconds (see Fig. 4).

When using this classifier to train a model using even more data, the time will increase, so to optimize this approach. It is necessary to apply parallelization and compare the obtained results. In order to solve this problem and reduce the training time of the classification model, an approach using parallel training using the different number of threads and computing technology using GPU - CUDA will be applied.

When training the RF model for parallel computation, the training data set is divided into number particles, and a classification tree is built for each. The simultaneous training of several trees and combining their results makes obtaining a trained model faster [18].

FullOfHolesCondition	0.98	0.98	0.98	627
SmoothCondition	1.00	0.99	0.99	3025
UnevenCondition	1.00	0.98	0.99	1340
HighCongestionCondition	1.00	0.95	0.97	637
LowCongestionCondition	0.99	1.00	0.99	3716
NormalCongestionCondition	1.00	0.94	0.97	639
AggressiveStyle	0.89	0.58	0.71	540
EvenPaceStyle	0.95	0.99	0.97	4452
micro avg	0.98	0.97	0.97	14976
macro avg	0.98	0.92	0.95	14976
weighted avg	0.98	0.97	0.97	14976
samples avg	0.98	0.97	0.97	14976

```
Model training time is  121.47707033157349
Model Classification Accuracy =  0.9192708333333334
```

Fig. 5 RF model training on 2 threads

After training the model using two streams (see Fig. 5), it was possible to reduce the time of the training process by 2.1 times compared to sequential training.

	precision	recall	f1-score	support
FullOfHolesCondition	0.98	0.98	0.98	627
SmoothCondition	1.00	0.99	0.99	3025
UnevenCondition	1.00	0.98	0.99	1340
HighCongestionCondition	1.00	0.95	0.97	637
LowCongestionCondition	0.99	1.00	0.99	3716
NormalCongestionCondition	1.00	0.94	0.97	639
AggressiveStyle	0.89	0.58	0.71	540
EvenPaceStyle	0.95	0.99	0.97	4452
micro avg	0.98	0.97	0.97	14976
macro avg	0.98	0.92	0.95	14976
weighted avg	0.98	0.97	0.97	14976
samples avg	0.98	0.97	0.97	14976

```
Model training time is  77.025221824646
Model Classification Accuracy =  0.9192708333333334
```

Fig. 6 RF model training on 4 threads

When training the classification model using 4 streams (see Fig. 6), it took 3.2 times less time compared to sequential execution.

	precision	recall	f1-score	support
FullOfHolesCondition	0.98	0.98	0.98	627
SmoothCondition	1.00	0.99	0.99	3025
UnevenCondition	1.00	0.98	0.99	1340
HighCongestionCondition	1.00	0.95	0.97	637
LowCongestionCondition	0.99	1.00	0.99	3716
NormalCongestionCondition	1.00	0.94	0.97	639
AggressiveStyle	0.89	0.58	0.71	540
EvenPaceStyle	0.95	0.99	0.97	4452
micro avg	0.98	0.97	0.97	14976
macro avg	0.98	0.92	0.95	14976
weighted avg	0.98	0.97	0.97	14976
samples avg	0.98	0.97	0.97	14976

```
Model training time is  55.0640594959259
Model Classification Accuracy =  0.9192708333333334
```

Fig. 7 RF model training on 8 threads

When using the maximum possible number of streams (8 threads, see Fig. 7) of the machine on which the classifier's training was performed, it was possible to reduce the time by 5 times. Notably, the accuracy of such a model was the same regardless of the number of threads.

Cuml software library was used to train the classifier model using GPU and CUDA technology. Although training a classification model using a GPU requires preprocessing parts of the training data set and loading them directly onto the GPU, as seen in Fig. 8, the training process is much faster.

```
                    FullOfHolesCondition       0.99      0.98      0.98       627
                        SmoothCondition       1.00      0.99      0.99      3025
                       UnevenCondition        1.00      0.98      0.99      1340
                  HighCongestionCondition     1.00      0.95      0.97       637
                   LowCongestionCondition     0.99      1.00      0.99      3716
                NormalCongestionCondition     0.99      0.93      0.96       639
                      AggressiveStyle         0.90      0.58      0.70       540
                       EvenPaceStyle          0.95      0.99      0.97      4452

                           micro avg          0.98      0.97      0.97     14976
                           macro avg          0.98      0.92      0.95     14976
                        weighted avg          0.98      0.97      0.97     14976
                         samples avg          0.98      0.97      0.97     14976

Model training time is 5.8234123
Model Classification Accuracy =  0.9184695512820513
```

Fig. 8 Training the RF model using the GPU

The time required for training was reduced by 51 times, indicating CUDA technology's effectiveness in parallel training of the classifier model.

We will consider the results of the training, namely the execution time and accuracy of the trained classification model, using the parallel implementation of the program on CPU and GPU, and conduct a comparative analysis.

Table 1

Program execution time with sequential and parallel training, s

Sequential execution	Parallel execution, number of threads			
	2	4	8	GPU
252.85	121.47	77.02	55.06	5.82

As we can see from Table 1, although the training time of the model with the maximum possible number of parallel threads for the architecture we use is 5 times less than the training time with sequential processing, it is pretty slow compared to training on the GPU. This indicates the incredible computing power of the video card and the advantages of performing parallel calculations on it relative to calculations on the processor.

Table 2

Accuracy of the trained model, %

Sequential execution	Parallel execution, number of threads			
	2	4	8	GPU
91.86	91.92	91.92	91.92	91.84

From the results presented in Table 2, the accuracy experienced slight deviations depending on the number of threads or the computational unit on which the training was performed. However, the quality of the performance was not affected.

Now let us calculate experimental indicators of acceleration and efficiency of parallel algorithms at different numbers of threads if parallel calculations are carried out on the processor, as well as indicators of acceleration of parallel algorithms for GPU.

First, let us perform a theoretical speedup evaluation for different numbers of parallel threads that we will use to train our tree. It should be emphasized that it is analytically impossible to calculate a theoretical estimate of the acceleration of model training on GPU since the number of graphics cores used during training is still being determined.

$$S_2(24957) = \frac{O(24957 * log(24957) * 17 * 800}{O(\frac{24957}{2} * log\left(\frac{24957}{2}\right) * 17 * 800)} \approx 2.1;$$

$$S_4(24957) = \frac{O(24957 * log(24957) * 17 * 800}{O(\frac{24957}{4} * log\left(\frac{24957}{4}\right) * 17 * 800)} \approx 3.6;$$

$$S_8(24957) = \frac{O(24957 * log(24957) * 17 * 800}{O(\frac{24957}{8} * log\left(\frac{24957}{8}\right) * 17 * 800)} \approx 5.8.$$

Table 3

The acceleration indicators of the parallel algorithm are obtained based on numerical experiments

Number of threads			GPU
2	**4**	**8**	
2.1	3.2	5	51

Table 3 shows the acceleration indicators obtained with the help of parallel execution of the algorithm for the different number of threads when working on the CPU and with the help of execution on the GPU.

The following conclusion can be drawn by comparing the theoretical estimation of acceleration and the results obtained in Table 3. The training was conducted on a laptop with a 4-core processor that supports Hyper-Threading technology, which provides an opportunity to obtain additional 4 computing threads (virtual threads). Although we can use 8 parallel threads for training, the load on the processor increases significantly, which leads to a loss of efficiency in the calculation of operations. Therefore, when increasing the number of parallel threads we use to train the model, we can see that the actual speedup estimate is significantly different from the theoretical one. If we talk about the parallel execution of training on the processor, then when the number of threads increases, the acceleration value increases. However, as we can see, compared to the execution of parallel calculations on the video card, the acceleration obtained by execution on the processor is much smaller, which once again indicates the incredible power of the execution of calculations with the help of the video card.

Since a theoretical acceleration estimate was made, which is the maximum we can achieve, we assume that the theoretical efficiency estimate is also the maximum possible and being equal to 1.

Table 4

Actual performance indicators of the parallel algorithm with different number of threads of the processor

Number of threads		
2	4	8
1	0.8	0.625

Analyzing Table 4, we can see that the efficiency decreases as the number of parallel threads we use to train the model increases. This can be explained by the fact that when the number of parallel threads we use for calculation increases, the load on the processor increases, which in turn, causes the calculation to be less efficient. Therefore, it will be advisable to use the GPU as a more efficient unit when performing calculations.

Conclusions

Modern information systems are increasingly used in various areas of our life. One of these is the quality control of the condition of the road surface in order to carry out repair work on time if necessary. The machine learning method can facilitate the control process, which was demonstrated in this work.

Data analysis methods for classifying road surface conditions were considered in detail. The research used a data set that contains approximately 25,000 records with 17 parameters about the car's movement, engine operation, road traffic conditions, driver behavior, and road surface conditions.

During the research, several models were successfully trained for the classification of the road surface condition, and a comparative analysis of their work was carried out using the AutoML software library.

The results, which were the best obtained using the RF ensemble method of machine learning, were discussed in more detail. After analyzing the performance of the classifier with different parameters and searching for the best hyperparameters, it was possible to achieve an accuracy of 91.9% classification of the road surface condition.

To improve the performance of the model, parallel computing was applied when training the model, which made it possible to speed up the training by 5 times using the CPU and 51 times using the GPU.

In this way, information technology was developed to train a classification model based on an input data set in the shortest possible time. This model can further be used to analyze and classify the data set for automated determination of the condition of the road surface.

References

1. Menegazzo Jeferson, Von Wangenheim Aldo. Road Surface Type Classification Based on Inertial Sensors and Machine Learning: A Comparison Between Classical and Deep Machine Learning Approaches for Multi-Contextual Real-World Scenarios. *Computing* 103(4), 2021. doi:10.1007/s00607-021-00914-0.
2. Mochurad L., Ilkiv A. A novel method of medical classification using parallelization algorithms. *Comput. Syst. Inf. Technol.*, № 1, pp. 23–31, Apr. 2022, doi: 10.31891/CSIT-2022-1-3.
3. Li J., Liu T., Wang X., and Yu J. Automated asphalt pavement damage rate detection based on optimized GA-CNN. *Autom. Constr.*, Vol. 136, p. 104180, Apr. 2022, doi: 10.1016/j.autcon.2022.104180.
4. Basavaraju A., Du J., Zhou F., and Ji J. A Machine Learning Approach to Road Surface Anomaly Assessment Using Smartphone Sensors. *IEEE Sens. J.*, 20(5), pp. 2635–2647, 2020, doi: 10.1109/JSEN.2019.2952857.
5. Setiawan B.D., Serdult U.I. and Kryssanov V. Smartphone Sensor Data Augmentation for Automatic Road Surface Assessment Using a Small Training Dataset, in *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Jeju Island, Korea (South), Jan. 2021, pp. 239–245. doi:10.1109/BigComp51126.2021.00052.
6. Wu C. *et al.* An Automated Machine-Learning Approach for Road Pothole Detection Using Smartphone Sensor Data, *Sensors*, 20(19), p. 5564, Sep. 2020, doi: 10.3390/s20195564.

7. Mochurad L., Boyko N. Solving Systems of Nonlinear Equations on Multi-core Processors. *Advances in Intelligent Systems and Computing IV*, Vol. 1080, N. Shakhovska and M. O. Medykovskyy, Eds. Cham: Springer International Publishing, 2020, pp. 90–106. doi: 10.1007/978-3-030-33695-0_8.
8. Martinez-Ríos E.A., Bustamante-Bello M.R., Arce-Sáenz L.A. A Review of Road Surface Anomaly Detection and Classification Systems Based on Vibration-Based Techniques. *Applied Sciences*. 2022; 12(19):9413. doi:10.3390/app12199413.
9. Nausheen Saeed, Roger G. Nyberg & Moudud Alam. Gravel road classification based on loose gravel using transfer learning, *International Journal of Pavement Engineering* 2022*,* doi: 10.1080/10298436.2022.2138879.
10. Ferjani I., Ali Alsaif S. How to get best predictions for road monitoring using machine learning techniques. *PeerJ Computer Science* 8:e941, 2022, doi:10.7717/peerj-cs.941.
11. Agebure M.A., Oyetunji E.O., Baagyere E.Y. A three-tier road condition classification system using a spiking neural network model. *J. King Saud Univ. - Comput. Inf. Sci.*, 34(5), pp. 1718–1729, May 2022, doi: 10.1016/j.jksuci.2020.08.012.
12. Azizah N., Riza L.S., Wihardi Y. Implementation of random forest algorithm with parallel computing in R. *J. Phys. Conf. Ser.*, Vol. 1280, p. 022028, Nov. 2019, doi: 10.1088/1742-6596/1280/2/022028.
13. Meyer B.H., Pozo A.T.R., Zola W.M.N. Improving Barnes-Hut t-SNE Algorithm in Modern GPU Architectures with Random Forest KNN and Simulated Wide-Warp. *ACM J. Emerg. Technol. Comput. Syst.*, 17(4), pp. 1–26, Jun. 2021, doi: 10.1145/3447779.
14. A set of data about car movement information, the operation of its internal combustion engine. URL: https://www.kaggle.com/code/absolutegaming/road-prediction/data.
15. Silva N., Soares J., Shah V., Santos M.Y., Rodrigues H. Anomaly Detection in Roads with a Data Mining Approach. *Procedia Comput. Sci.*, Vol. 121, pp. 415–422, 2017, doi: 10.1016/j.procs.2017.11.056.
16. Kappal S. Data normalization using median median absolute deviation MMAD based Z-score for robust predictions vs. min–max normalization. *Lond. J. Res. Sci. Nat. Form.*, 19(4): 39-44, 2019.
17. Christine Dewi. Random Forest and Support Vector Machine on Features Selection for Regression Analysis. *International journal of innovative computing, information & control: IJICIC* 15(6):2027–2037, 2019, doi: 10.24507/ijicic.15.06.2027.
18. Bruce P.C. and Bruce A. Practical statistics for data scientists: 50 essential concepts, *First edition. Sebastopol, CA: O'Reilly*, 2017.
19. Lindroth L. Parallelization of Online Random Forest. 2021. Accessed: Jun. 05, 2022. [Online]. Available: http://urn.kb.se/resolve?urn=urn:nbn:se:bth-21098.

Lesia Mochurad **Леся Мочурад**	PhD, Associate Professor of Department of Artificial Intelligence, Lviv Polytechnic National University, Lviv, Ukraine e-mail: lesia.i.mochurad@lpnu.ua. https://orcid.org/0000-0002-4957-1512	кандидат технічних наук, доцент, доцент кафедри систем штучного інтелекту національного університету "Львівська політехніка", Львів, Україна
Andrii Ilkiv **Андрій Ільків**	student of Department of Artificial Intelligence, Lviv Polytechnic National University, Lviv, Ukraine e-mail: andrii.ilkiv.knm.2018@lpnu.ua https://orcid.org/0000-0001-6438-0784	студент кафедри систем штучного інтелекту національного університету "Львівська політехніка", Львів, Україна
Oleksandr Kravchenko **Олександр Кравченко**	PhD student of National Technical University "Kharkiv Polytechnic Institute", Kharkiv, Ukraine e-mail: askraff@gmail.com https://orcid.org/0000-0002-6169-1250	аспірант 2-го курсу (Комп'ютерні науки), Національний технічний університет "Харківський політехнічний інститут", Харків, Україна