FACTA UNIVERSITATIS
Series: **Automatic Control and Robotics** Vol. 22, Nº 1, 2023, pp. 1 - 13
https://doi.org/10.22190/FUACR230123001P

Regular Paper

APPLICATION OF CONVOLUTIONAL NEURAL NETWORKS FOR ROAD TYPE CLASSIFICATION

UDC (004.85:004.032.26)

Staniša Perić, Nina Vukić, Dragan Antić, Marko Milojković, Anđela Đorđević

University of Niš, Faculty of Electronic Engineering, Department of Control Systems, Republic of Serbia

Abstract. *This paper presents an application of convolutional neural networks (CNN) in autonomous driving system, which represents the capacity of a vehicle to operate mostly or entirely autonomously, with or without human assistance. After basic introduction of CNN, we design and apply CNN-based algorithm in road type classification problem. The performed simulations show that satisfactory results can be achieved even with little data available.*

Key words: *Convolutional neural network, autonomous vehicles, road surface*

1. INTRODUCTION

Road surface classification is an important task that enables autonomous vehicles and driving assistance systems to adapt to the driving scene and the current environment. In fact, it is necessary to take note of the influence of the environment, and especially the road surface, on the driving scene, and therefore, the driving policy and style of the autonomous vehicle. The literature doesn't offer a general solution to this problem. There needs to be more research done in order to get to a system that could work well in any driving environment i.e., on any possible road type as it is a difficult task to create an algorithm that would be so general that it recognizes all road types [1].

There are many tasks that an autonomous driving system needs to perform. In recent years, a lot of progress has been made in the field of autonomous vehicles as there are already

Received January 23, 2023 / Accepted March 14, 2023
Corresponding author: Staniša Perić
University of Niš, Faculty of Electronic Engineering, Department of Control Systems, Aleksandra Medvedeva 14, 18000 Niš, Republic of Serbia
E-mail: stanisa.peric@elfak.ni.ac.rs

vehicles with some autonomous functions on the streets. Many researchers focus on finding out the best algorithms for creating an autonomous driving system and a lot of successful studies have already been done in this field. A lot of studies focus on the usage of convolutional neural networks (CNN) for autonomous driving tasks.

Convolutional neural networks are widely used in many modern applications, especially those that include computer vision tasks, such as object recognition. Theoretical aspects of CNNs and their usage, as well as their advantages in comparison to traditional neural networks and other methods have been widely covered in literature. These networks can be applied for a variety of driving tasks, such as object detection, traffic signs recognition, lane detection, pedestrian detection etc. [2]-[6]. One of the tasks in which convolutional neural networks may also be applied is the road type classification.

Road type classification can be done by using different kinds of data. For example, the study [1] used images obtained from *Google Street View* and a CNN model to recognize one of the six types of roads: divide roads, express roads, paved street roads, unpaved street roads, parquet streets, and distorted roads. The study included both road type classification and road quality detection, and the experiments were done on both raw and pre-processed data. The accuracy achieved for road type classification on raw data was 82.03%, and after using pre-processed data, the accuracy was 91.41%.

On the other hand, the paper [7] focuses on discovering the road friction coefficient. More precisely, the authors use the information acquired from tires as they are in direct contact with the driving surface and they can give an understanding of the road friction which depends on both the road type and the current weather conditions. Therefore, the data used for classification are the acoustic signals that are generated while driving, when the tire gets in contact with the road surface. Then, the friction is estimated indirectly, by using a CNN model to classify the road surface in real time. The paper emphasizes the importance of road surface classification as it enables more robust decision-making. It is pointed out that the friction must be estimated in real time and that it is of upmost importance for vehicle actions such as braking. The conducted experiment used two types of tires and two types of road surfaces and CNN was successfully implemented for the classification task.

The importance of knowing the road-tire friction coefficient in order to control the vehicle and achieve stability is also emphasized in [8]. As the friction coefficient is mainly influenced by the road type and road surface condition, even though there are other factors that influence it as well, road surface classification is an important part of estimating the friction coefficient. The paper used two different CNN models for classifying road surfaces. However, the authors point to the necessity of a good dataset in order to achieve good classification results. They mention that there are no available datasets made especially for road type classification that cover enough weather conditions and road types, as well as that there are always minority classes in each dataset for which not enough images are provided, making the dataset unbalanced. In order to overcome such problems, the authors attempt to create a new dataset by mixing already available ones in order to classify roads into six different types. They also used *Google Images* to add more images to classes which are underrepresented. When the authors tested the models on whole images, which enables the model to have additional information on the environment, the performances of the models were not satisfactory since they faced overfitting issues. However, when tested on cropped images which contained only the region of interest, the results were better, with accuracy over 90%.

In the paper [9], the authors also mention the importance of knowing the surface type for interpreting the environment while driving. However, the paper provides a different approach to road classification as it uses occupancy grids as data for the classification. Occupancy grid mapping method divides the environment into cells which all have an attached probability of being occupied. The authors state that this method's advantage is the invariance to the illumination, as well as the possibility to use any sensor without being limited to using a camera, for example. The roads were classified into four different classes: freeway, highway, parking area and urban environment. Both CNNs and Support Vector Machine classifier were used and the results were compared. In order to use these classifiers, grids are treated like images and fed to the model. Both types of classifiers were able to classify data correctly with the achieved accuracy of 94%.

The remainder of this paper is organized as follows. In Section 2, a basic understanding of the convolutional neural networks is given. In Section 3, the basics of autonomous vehicles are presented, which is followed by Section 4 in which road classification task is explained in more detail. Lastly, in Section 5, an explanation of the experimental testing of CNN architectures for road type classification and acquired results are presented.

2. Convolutional Neural Networks

Convolutional Neural Network (CNN) is a type of neural networks that is specialized for working with unstructured data, such as images, videos, text, speech etc. [10]-[12]. The main difference between this type of networks and the traditional neural networks is their architecture. They always consist of one or more convolutional layers, but they also include some other special layers. Most often, convolutional neural networks are made up of three types of layers: convolutional layers, pooling layers and fully-connected layers [13]. Unlike a traditional fully-connected network in which each neuron has a connection to all the other neurons in the previous and the following layer, neurons in a convolutional neural network are connected to a small region of a previous layer called a *receptive field* [14, 15]. Input data is presented in a form of a multidimensional matrix or a *tensor*. For example, an RGB image can be presented as a tensor of dimensions $H \times W \times 3$, where H is the image height, W is the image width and 3 refers to the number of color channels – red, green and blue for an RGB image [16]. This kind of a representation is useful as it allows a convolutional neural network to search through the data and extract its features, which is done in the convolutional layers, thus forming the output feature maps, which present an input for pooling layers. Pooling layers help reduce the dimensionality of the feature maps, and the fully connected layers allow mapping of the extracted features to the output of the network [13].

Convolutional layers perform the operation called *convolution*, which can be described as the application of filters or *kernels* to the input data in order to extract the most important characteristics of the data [17]. Kernel is essentially a matrix of numeral values or *weights* and there is a variety of filters that can be used in order to search through data [13,17]. Filters are applied to data starting from the top left corner, and moving to the right and then downwards. In each step, elements in corresponding positions in the matched input data and filter region are multiplied, the scores are summed up and an element of the output feature map in the same corresponding position is created. The process is repeated multiple times, until the output feature map is formed [13]. This process can be visualized as presented in Figure 1. By applying different filters, different feature maps are created [13]. For example, filters such as Gaussian filter, vertical and horizontal Sobel filters, mean filter etc. can be used for images in order to blur them, extract vertical and horizontal edges, reduce noises etc. [10, 18].

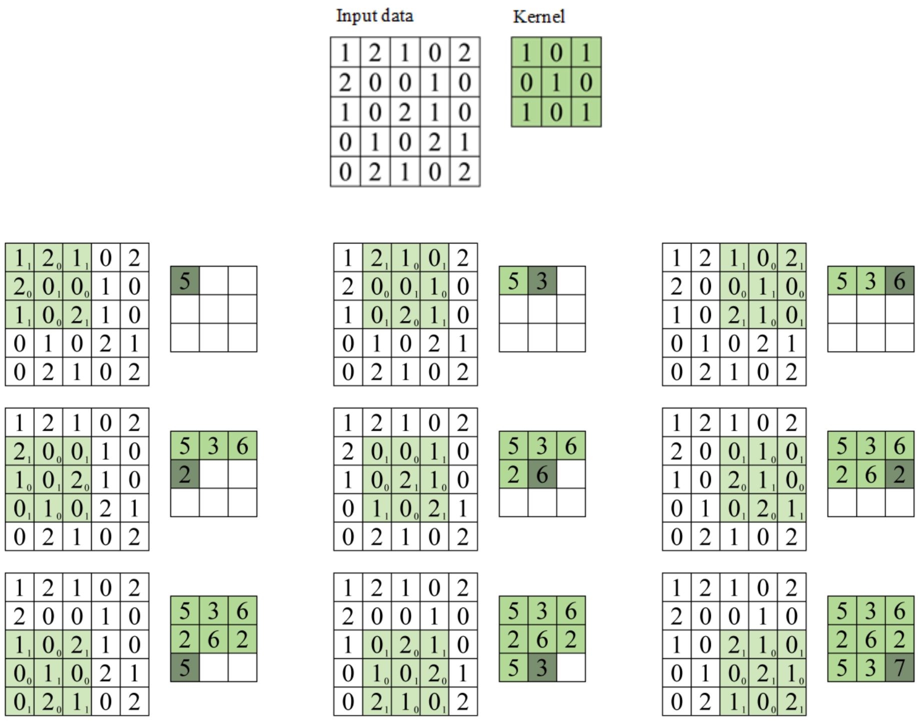

Fig. 1 Convolution process [19].

There are some parameters that need to be considered in the process of convolution. One of them is *stride*, which represents the number of matrix cells (pixels, if considering images) that are skipped while moving the filter to the right and downwards in each step of the convolution [11, 20]. Larger stride means a smaller output map as some of the cells are skipped over [20]. Figure 1 uses stride 1, i.e., no pixels are skipped in the process. Another important parameter is the filter size. Usually, size 3×3 filters are used, even though it is possible to use some other dimensions as well [13]. It should be noted that these dimensions present the width and the height of the filter, but that filters can also have a third dimension – the depth which is the same as the third dimension of the data [17].

It can be noticed that input data dimensions will be reduced with each convolution. However, in some applications, it is important to keep the original dimensions of the data. In such cases, padding is used. Padding refers to the addition of outer matrix cells to the input data. Usually, it is done by adding zeros, which is called *zero padding*. By using padding, elements in the corner cells of the data get a bigger role in the convolution, which can sometimes be important [10, 11, 13, 20].

Neural networks use activation functions to decide which information should be passed on to the following neurons. Considering the linearity of convolution, a non-linearity must be introduced in convolutional neural networks since most of the real-world data is non-linear. Therefore, non-linear activation functions are used [20]. Most often, the *Rectified Linear Unit* or *ReLU*, which can be described as $f(x) = \max(0,x)$ is used [15].

Essentially, convolutional neural network recognizes features of the data, i.e., it recognizes some patterns that are present in the data. For example, if a CNN is used to recognize an object in an image, it can recognize the object no matter its location in the image because filters are applied in different regions of the image and features are detected in any part of the image. The focus is on discovering whether the image possesses a certain feature, not to detect the location of that feature. This is called *translation invariance*, and it is one of the reasons why convolutional neural networks are so successful when working with images [17].

Another important characteristic of convolutional neural networks is the *hierarchical decomposition* of the input, which refers to the hierarchy of the features that the network recognizes through its layers. CNN works in such a way that the more convolutional layers are applied, the more complex the features that can be recognized. First layers recognize the simplest features such as lines and simple shapes. Then, the following layers are able to recognize the combinations of the simple features. Going deeper into the networks, more and more complex features are detected, and the network becomes able to recognize objects such as people, animals, cars, buildings etc. Therefore, it can be said that the features are becoming more abstract with the depth of the network [16, 17].

Feature maps can have some redundant values. For example, pixels in an image that are next to each other are usually very similar, which means that not all pixels contain important information. Therefore, when extracting features from such data, some values are not necessary and the feature maps can be reduced in order to keep the most important information only [10, 20]. This reduction refers to the dimensions of the feature maps and it is done in pooling layers of a convolutional neural networks [18]. Pooling layers are important as they allow the usage of less computing resources [21, 22] and transform the input data so that working with them is simpler [20]. Also, the possibility of overfitting happening is reduced because feature representation is more generalized [10], [20]-[23]. Pooling can be explained similarly to the convolution. Filters are applied to the input feature maps starting from the top left corner, and going to the right and downwards. Filter size, stride and padding need to be set before pooling [13]. However, these filters are used differently in the process of pooling. Most often, maximum pooling is used, but some other types, such as minimum pooling and average pooling are also present. Maximum pooling matches the filter to the input feature map, and takes only the maximum value of the matched region as the output value in the corresponding cell [12, 14, 20], as can be visualized in Figure 2.

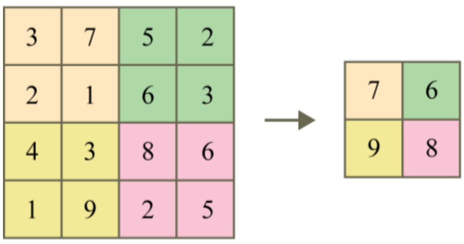

Fig. 2 Maximum pooling [11].

Similarly, average pooling takes the average value of the cells in the matched region of the input feature map [12, 18], which can be visualized as in Figure 3. Pooling usually uses 2×2 filter size with stride 2, which results in a two times smaller feature map [13, 14]. However, the depth of the data does not change with pooling i.e., the reduction of the dimensions refers to the width and the height. Pooling is applied to each feature map, so their number remains unchanged [20].

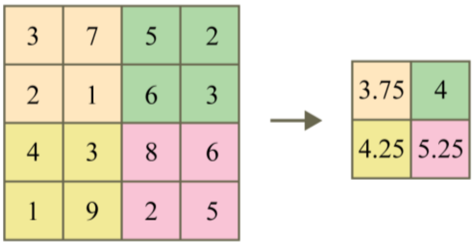

Fig. 3 Average pooling [11].

Fully connected layers are located at the end of the convolutional neural network and they produce the final output of the network. These layers can be seen as traditional fully connected neural network layers, in which each layer consists of neurons that are connected with all neurons of the previous and the following layer [20]. The task of the fully connected layers in a CNN is prediction [13, 20]. In classification tasks, prediction refers to the possibility that the input data belongs in a certain class [23]. On the other hand, in regression problems, the output is a numerical value that the network predicted [13]. Depending on the task of the network, there are different activation functions that are applied to the last layer of the network, which produces the output [23]. Either way, the activation function of the last layer of the network is different than those in all the other layers [13]. For example, for classification problems, *softmax* function is often used. The result of this function is a vector which consists of values between 0 and 1 which all sum up to 1 [13, 20, 23].

3. Autonomous Vehicles

Autonomous vehicle is a vehicle that can move without direct human control. It can make decisions and do certain actions on its own, based on the applied algorithm. The vehicle collects information on its environment through many sensors and makes decisions according to that information. In recent years, a lot of progress has been made in the field of autonomous vehicles with the usage of machine learning and artificial intelligence [22]. It can be argued that the usage of autonomous vehicles can lead to safer road traffic, less traffic accidents, more mobility for older people or people who cannot drive, better usage of time, less driving induced stress, less emissions etc. [24]. However, despite having cars with some autonomous functions, it is still a very difficult task to develop a vehicle that could drive in all possible scenarios considering the complexity of the road traffic and unpredictability of other drivers [25].

In order to understand the complexity of this problem, many tasks that are put ahead of autonomous vehicles can be considered: they need to drive inside certain road boundaries, inside the correct lane, they have to follow speed regulations, traffic lights, traffic signs, crosswalks, they have to keep a certain distance from other vehicles, they have to understand rules when it comes to changing lanes, overtaking other vehicles, intersections, they have to find the best possible route to their destination, avoid obstacles, park etc. [26]. In fact, there are many more tasks and situations that a vehicle could encounter on the road, making the development of an autonomous vehicle an extremely difficult task.

SAE J3016 is a standard given by *The Society of Automotive Engineers* (*SAE*) that defines different levels of automation of autonomous driving systems on a scale from 0 to 5 [22, 25]. Level 0 means that there is no automation at all i.e., all driving tasks are a human responsibility. Levels 1 and 2 refer to some driving assistance systems which include some type of automation, such as emergency braking system, for example [24, 25]. Level 3 includes autonomous vehicles that are present today. These vehicles include the autonomous driving option, i.e. they can either be driven by the driver or they can drive autonomously. However, the autonomous option can be used in some scenarios, and it is still expected that the driver takes control of the vehicle in emergency situations, if necessary. Levels 4 and 5 have not yet been achieved by any available vehicles today. They refer to vehicles that do not need any kind of human intervention [25].

Architecture-vise, autonomous vehicles can be ego-only systems or connected systems. Ego-only systems consist of only one vehicle, while connected systems have more sub-systems of vehicles and infrastructure working together [25]. Both types of systems can be realized as modular or end-to-end systems. Modular systems consist of several components working together in order to produce an output, that is, the movement of the vehicle, based on the inputs coming from various sensors. On the other hand, end-to-end systems directly map inputs to the output. Either way, a supervision monitor is used in order to ensure the safety of the whole system [22, 25]. Block diagram of a modular autonomous vehicle is presented in Figure 4. Modular autonomous driving system consists of a few basic components, i.e. tasks, which are: perception and localization; global route planning; local route planning; and motion control [22]. An advantage of the modular system is that it allows the development of more components that are simpler in comparison to the development of one complex system [25].

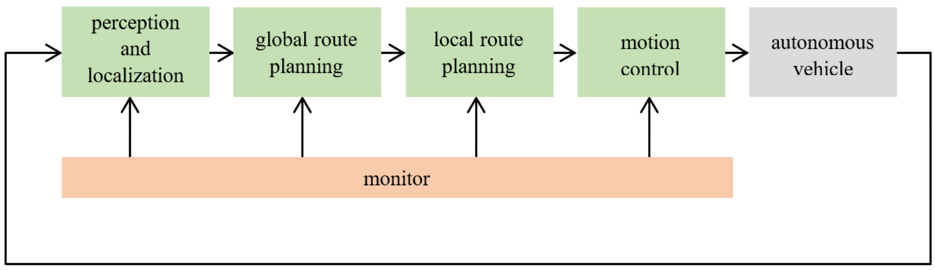

Fig. 4 Block diagram of a modular autonomous vehicle [22].

Perception has the goal of creating a representation of the current environment that can include, for example, the position of road lanes, other vehicles, pedestrians, traffic lights, etc. It gives the vehicle an awareness of its surroundings [27]. In other words, perception represents a task to recognize and collect all information on vehicle’s environment that could be helpful for safe vehicle control [25]. It is very important for the vehicle to have the best possible perception of its environment in order to avoid possible mistakes. Therefore, autonomous vehicle usually includes many sensors that would collect information and form an abstract representation of the current driving scene that would then be proceeded to the following modules in order to make decisions [27]. Convolutional neural networks are often used for perception, in order to detect important information in the collected data from sensors

[22]. Many sensors can be used for perception. Cameras are often used as passive sensors that collect information without emitting any signals that could interfere with other signals in the environment [25, 26]. Cameras are cheap and simple to use, but they are heavily light and weather dependent, which is a disadvantage in autonomous driving tasks which require high accuracy detection in any road condition. Lidars and radars are also often used in autonomous driving tasks. These sensors give a tridimensional representation of the environment and allow the vehicle to have a perception on the distance of many objects. They can also be influenced by poor weather conditions, such as heavy fog, rain or snow [22, 25]. Lidars and radars are active sensors that emit signals (infrared signals in case of lidars, and radio waves in case of radars) towards other objects in the environment. By measuring the time necessary for the signal to travel to the object and back, it is possible to calculate distance between the sensor and the object [25].

Perception includes many smaller tasks, such as object detection, object tracking, road and lane detection, just to name a few. Object detection is a task in which it is necessary to find out the location of an object, as well as to estimate its size. In terms of autonomous vehicles, it refers to the detection of objects such as traffic signs, traffic lights, crossroads, other vehicles, pedestrians etc. Therefore, object detection must be done fast, in real time, with high accuracy. Object tracking is necessary in complex driving scenarios, and it refers to the estimation of the direction and the speed of dynamic objects in order to avoid collision. Road and lane detection is a task of recognizing the part of the environment on which the vehicle should drive, i.e. the road and the lane. This kind of a detection is somewhat different than object detection since roads are continuous surfaces. It is also necessary that the vehicle understands the semantics of the road, which is a complex task [25].

While perception focuses on recognizing other objects in the environment, localization is a task of finding out the location of the vehicle in relation to the environment. The vehicle must be aware of its own position in order to keep driving correctly [25]. Localization algorithms calculate the vehicle's position and orientation while it's moving and estimate its next position based on the previous and the current one. Deep learning can be used in order to recognize the driving scene i.e., whether the vehicle is currently driving in the city, on the highway, on a parking lot etc. [22].

Route planning is a task of finding out the best path between the starting position of the vehicle and its destination. While route planning, all obstacles should be considered so that the vehicle doesn't get into a collision with any of them [22]. Generally, route planning can be divided into two smaller tasks: global route planning and local route planning. Global route planning is a task of discovering the best path in the map of roads that would get the vehicle from its starting point to its destination, which can be done by navigation systems that use GPS and offline maps, which are already well developed and present in many cars that are used nowadays. On the other hand, local route planning takes into consideration all obstacles in the vehicle's environment and decides on the vehicle's trajectory on a lower level [25].

Finally, motion control refers to a system that takes all previous modules into consideration so that the vehicle would finish the desired task, and selects values of the control variables which are in charge of the movement of the vehicle [24]. Therefore, motion controllers give out actual commands which control the vehicle [22], such as defining velocity, steering angle, braking etc. [27]. Control system should also react in case of an error in the planned motion [24]. Safe vehicle control requires the usage of sensors which would monitor the inner state of the vehicle. This information should be available at all times [25].

4. ROAD CLASSIFICATION

The road surface is particularly important, not only for autonomous vehicles, but for all vehicles in general, as it determines the speed, braking, driving style etc. An arising problem in many developed countries is the fact that the road infrastructure is aged, lacks maintenance and therefore it may become even more damaged [28]. It can be noted that many research focus on road damage detection, and not road type classification in particular. However, in many undeveloped countries, the road infrastructure problems are even more noticeable, as there are many areas with unpaved and/or damaged roads.

Changes in road infrastructures could help in the development of autonomous vehicles. However, improving roads could be justified when it comes to the roads used for public transportation. When it comes to private cars and vehicles, this could be overly expensive and time consuming, therefore making it too complex of a task to be considered. There are ideas to make special highways for autonomous vehicles only which would be fully adapted to this kind of vehicles [26].

Road classification algorithms could be a helpful addition to vehicles as a part of a driving assistance system that would allow them to adapt to the current driving scene and improve their safety and performance. New technologies allow the vehicle to alert the driver about road conditions. In terms of an autonomous vehicle, they are especially important for motion control and decision-making. It is necessary that the actuators react in real-time to the changing road scenarios, and to adapt their parameters. However, more time to adapt to the road condition can be acquired if the road detection and classification algorithm recognizes the road on which the vehicle is about to drive somewhat before the vehicle gets there. This way, there would also be less requirements for the actuators that must be used in the vehicles, i.e. simpler, slower and cheaper actuators could also give good results with reduced costs [29]. This could also be helpful for trajectory planning of the vehicle as it could plan better in case it approaches wet or snowy roads, for example [8].

Brake system of a vehicle is closely related to the road surface type. A better braking system could be achieved with road type classification. Road type classification requires the usage of sensors that would collect data on the road infrastructure. Much research has been done with various types of sensors used to collect data for road classification or damage detection. First of all, sensors that can be used can either be in-road sensors or in-vehicle sensors. Inroad sensors refer to those sensors that are placed in the road surface to measure its characteristics. After collecting data, it can then be sent to the vehicle for the following steps. Sensors such as temperature sensors, vibration sensors, cameras, light sensors etc. can be used as inroad sensors. On the other hand, in-vehicle sensors are sensors that are placed on the vehicle. For example, acoustic sensors, acceleration sensors, ultrasonic distance sensors, video and stereo cameras, radars, lidars etc. can be applied for the task of road type recognition. In many applications, deep learning methods including convolutional neural networks are used for classification tasks [29].

When it comes to image-based road surface classification, there have been approaches based on texture information, which is helpful to differentiate concrete from cobblestone, for example, but it is not that good at differentiating dry from wet surfaces [8]. Other approaches are color-based approaches which investigate color distributions in order to classify road types [30].

5. EXPERIMENTAL RESULTS

In this paper, the ability of convolutional neural networks to classify road surface types has been tested. As a dataset, the *Road Surface Image Dataset with Detailed Annotations* [31] was used. However, the number of classes that this dataset originally had was reduced to only 6 different classes. The original 27 classes differentiated not only surface types, but also the amount of damage and wetness of the road. When it comes to the road damage, there were three categories – smooth, slight and severe, and when it comes to the amount of water on the road, there were also three categories – dry, wet and water. For example, *dry_concrete_smooth* and *dry_concrete_severe* are two different classes, and so are the *dry_concrete_smooth* and *water_concrete_smooth*. The variety of classes makes this dataset very versatile as it can be used for many purposes that concern autonomous driving and assistance systems. The dataset includes 370151 240×360 sized images of different road surfaces.

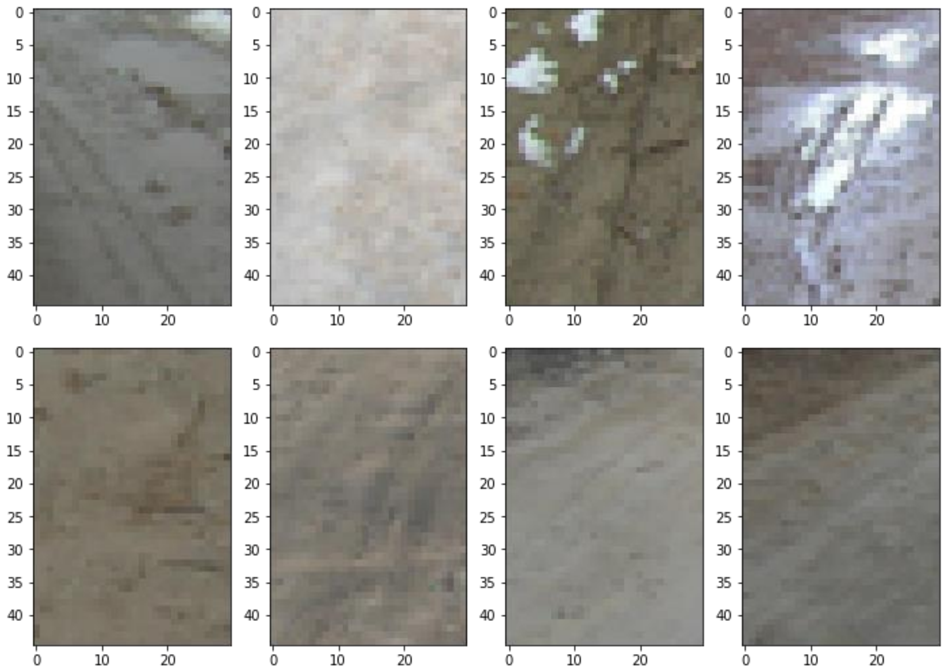

Fig. 5 An example of the images included in the dataset.

However, for the purpose of this project, there was no need to classify roads based on the amount of water or damage on the road, which is why classes that include one surface type were merged into a single one. Also, the number of images was greatly reduced so that each of the remaining classes had 3000 or less images. Images were resized to 30×45 pixels. An example of used images is shown in Figure 5, whereas Table 1 shows the classes used in this project.

Table 1 Classes with their respective labels

Class label	Surface type
0	mud
1	gravel
2	concrete
3	asphalt
4	snow
5	ice

In terms of the convolutional neural network that was used in the project, it consisted of 4 convolutional layers in total, with 32, 128, 512 and 512 filters of size 3×3 and *ReLU* activations. After each convolutional layer, maximum pooling layers were used. Lastly, there were three fully connected layers – two with *ReLU* activation functions and one with *softmax* function used to classify road types and give an output prediction of the network. Also, dropout layers and batch normalization were used. The network was trained using *adam* optimizer in 10 epochs. After the training, the network was tested using test data, and the results in terms of training and validation accuracy and loss are presented in Figure 6. The achieved training accuracy was 85,95%, while the validation accuracy was 76%. After testing, the achieved accuracy was 75%. As it can be noted, achieved results are good even with such little amount of data used.

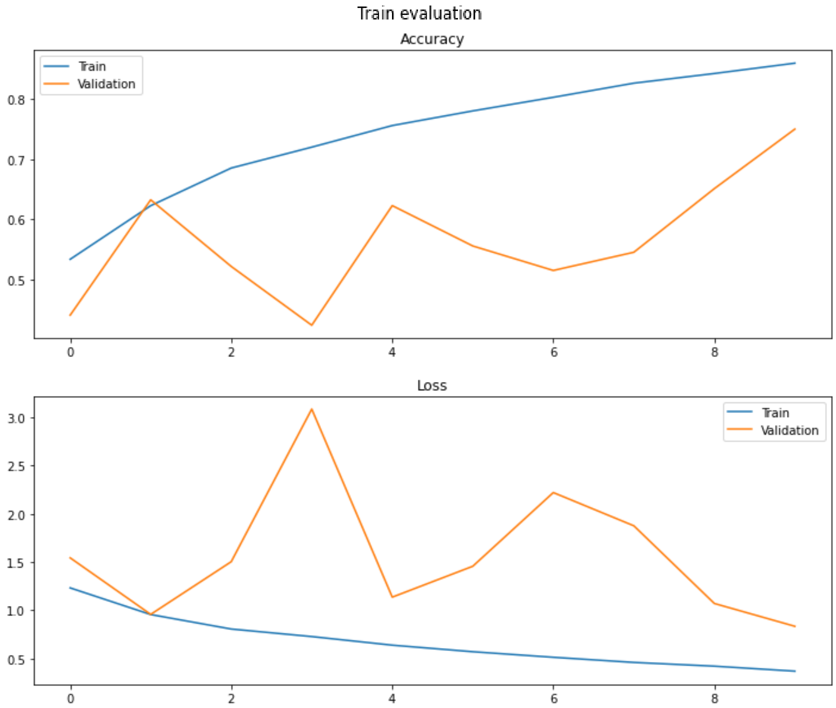

Fig. 6 Accuracy (above) and loss (below) during training (blue line) and validation (orange line).

6. CONCLUSIONS

In this paper, a brief introduction to convolutional neural networks and autonomous vehicles is given, with the focus on the application of convolutional neural networks in autonomous driving and/or assistance systems that could recognize road surface type. This problem is very important in an autonomous driving system since the road type directly influences the driving policy and style. There is a need for real-time road surface recognition and therefore the algorithms that are used must be fast and reliable. Using the information on the current road surface type or road surface type that the vehicle is approaching to, the control system of the vehicle can change its parameters to adapt to that specific road surface. This task is a very complex one and there needs to be more research done in order to create a generally applicable system that could recognize any type of road in any given weather and light condition.

Research in this field focused on recognizing the whole driving scene that is larger than the road itself, on discovering the road friction coefficient as it is directly connected with the road type, on the classification of paved and unpaved roads and the recognition of potholes and classification of the roads based on the amount of damage, as well as on classifying different road types. All of these problems can use different type of data based on the sensors that the vehicle uses to collect data, and many of them use convolutional neural networks for the classification task.

Convolutional neural networks are a good option for fast road type classification. They can work well even with little data available. However, future research should include more types of roads and more images, and even a larger image size for better recognition of nuances that differentiate very similar road surfaces.

Acknowledgement: *This work has been supported by the Ministry of Education, Science and Technological Development of the Republic of Serbia. Funded by the European Union under project Smart Products and Services Engineering (SPaSE) 101047566-ERASMUS-JMO-2021-MODULE. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or EACEA. Neither the European Union nor the granting authority can be held responsible for them.*

REFERENCES

[1] V. Tumen, O. Yildirim and B. Ergen, "Recognition of Road Type and Quality for Advanced Driver Assistance Systems with Deep Learning", Elektronika Ir Elektrotechnika, vol. 24, no. 6, pp. 67-74, December 2018, doi: 10.5755/j01.eie.24.6.22293.

[2] R. Valiente, M. Zaman, S. Ozer and Y. P. Fallah, "Controlling steering angle for cooperative self-driving vehicles utilizing CNN and LSTM-based deep networks", IEEE Intelligent Vehicles Symposium (IV), IEEE, pp. 2423-2428, June 2019.

[3] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao and K. Zieba, "End to End Learning for Self-Driving Cars", April 2016, [Online]. Available: https://arxiv.org/pdf/1604.07316v1.pdf

[4] Y. Satılmış, F. Tufan, M. Şara, M. Karslı, S. Eken and A. Sayar, "CNN based traffic sign recognition for mini autonomous vehicles", Information Systems Architecture and Technology: Proceedings of 39th International Conference on Information Systems Architecture and Technology – ISAT 2018, Advances in Intelligent Systems and Computing, vol. 853, pp. 85-94, August 2018, doi: 10.1007/978-3-319-99996-8_8.

[5] G. Rjoub, O. A. Wahab, J. Bentahar and A. S. Bataineh, "Improving autonomous vehicles safety in snow weather using federated YOLO CNN learning", Mobile Web and Intelligent Information Systems – MobiWIS 2021, Lecture Notes in Computer Science, vol. 12814, pp. 121-134, August 2021, doi: 10.1007/978-3-030-83164-6_10.

[6] F. Pizzati, M. Allodi, A. Barrera and F. García, "Lane detection and classification using cascaded CNNs", Computer Aided Systems Theory – EUROCAST 2019, Lecture Notes in Computer Science, vol. 12014, pp. 95-103, April 2020, doi: 10.1007/978-3-030-45096-0_12.
[7] J. Yoo, C. H. Lee, H. M. Jea, S. K. Lee, Y. Yoon, J. Lee, K. Yum and S. U. Hwang, "Classification of Road Surfaces Based on CNN Architecture and Tire Acoustical Signals", Applied Science, vol. 12, no. 19, 9521, September 2022, doi: 10.3390/app12199521.
[8] M. Nolte, N. Kister and M. Maurer, "Assessment of deep convolutional neural networks for road surface classification", 21st International Conference on Intelligent Transportation Systems (ITSC), IEEE, pp. 381-386, November 2018.
[9] C. Seeger, A. Müller, L, Schwarz and Michael Manz, "Towards road type classification with occupancy grids", IEEE Intelligent Vehicles Symposium 2016 Workshop: DeepDriving - Learning Representations for Intelligent Vehicles, June 2016, [Online]. Available: http://iv2016.berkeleyvision.org/papers/seeger.pdf
[10] V. Zhou, "CNNs, Part 1: An Introduction to Convolutional Neural Networks", November 2019, [Online]. Available: https://victorzhou.com/blog/intro-to-cnns-part-1/
[11] K. Džomba, Konvolucijske neuronske mreže (diplomski rad). Zagreb: Prirodoslovno-matematički fakultet, 2018.
[12] M. Arsenović, Detekcija bolesti biljaka tehnikama dubokog učenja (doktorska disertacija). Novi Sad: Fakultet Tehničkih nauka, 2020.
[13] R. Yamashita, M. Nishio, R. Kinh Gian Do and K. Togashi, "Convolutional neural networks: An overview and application in radiology", Insights Imaging, vol. 9, pp. 611-629, June 2018.
[14] K. O'Shea and R. Nash, "An Introduction to Convolutional Neural Networks", December 2015, [Online]. Available: https://arxiv.org/pdf/1511.08458.pdf
[15] M. Dabović and I. Tartalja, "Duboke konvolucijske neuronske mreže – koncepti i aktuelna istraživanja", Zbornik 61. Konferencije za elektroniku, telekomunikacije, računarstvo, automatiku i nuklearnu tehniku - ETRAN 2017, June 2017
[16] J. Wu, "Introduction to Convolutional Neural Networks", May 2017, [Online]. Available: https://cs.nju.edu.cn/wujx/paper/CNN.pdf
[17] J. Brownlee, "How do Convolutional Layers Work in Deep Learning Neural Networks?", April 2019, [Online]. Available: https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/
[18] L. Mehinović, Konvolucijske neuronske mreže za klasifikaciju objekata (diplomski rad). Pula: Sveučilište Jurja Dobrile u Puli, Fakultet informatike u Puli, 2020.
[19] M. Minđek, Kvantizacija konvolucijskih modela za raspoznavanje slika (završni rad). Zagreb: Sveuciliště u Zagrebu, Fakultet elektrotehnike i računarstva, 2021.
[20] U. Karn, "An Intuitive Explanation of Convolutional Neural Networks", August 2016, [Online]. Available: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/
[21] S. Saha, "A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way", December 2018, [Online]. Available: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
[22] S. Grigorescu, B. Trasnea, T. Cocias and G. Macesanu, "A Survey of Deep Learning Techniques for Autonomous Driving", Journal of Field Robotics, vol. 37, no. 3, pp. 362-386, March 2020.
[23] J. Brownlee, "Crash Course in Convolutional Neural Networks for Machine Learning", June 2016, [Online]. Available: https://machinelearningmastery.com/crash-course-convolutional-neural-networks/
[24] B. Paden, M. Čáp, S. Z. Yong, D. Yershov and E. Frazzoli, "A Survey of Motion Planning and Control Techniques for Self-driving Urban Vehicles", IEEE Transactions on Intelligent Vehicles, vol. 1, no. 1, pp. 33-55, April 2016.
[25] E. Yurtsever, J. Lambert, A. Carballo and K. Takeda, "A Survey of Autonomous Driving: Common Practices and Emerging Technologies", IEEE Access, vol. 8, April 2020.
[26] M. Bertozzi, A. Broggi and A. Fascioli, "Vision-Based Intelligent Vehicles: State of the Art and Perspectives", Robotics and Autonomous Systems, vol. 32, no. 1, pp. 1-16, July 2000.
[27] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. Al Sallab, S. Yogamani and P. Pérez, "Deep Reinforcement Learning for Autonomous Driving: A Survey", IEEE Transactions on Intelligent Transportation Systems, January 2021, [Online]. Available: https://arxiv.org/abs/2002.00444
[28] H. Maeda, Y. Sekimoto, T. Seto, T. Kashiyama and H. Omata, "Road Damage Detection Using Deep Neural Networks with Images Captured Through a Smartphone", February 2018, [Online]. Available: https://arxiv.org/pdf/1801.09454.pdf
[29] E. Šabanovič, V. Žuraulis, O. Prentkovskis and V. Skrickij, "Identification of Road-Surface Type Using Deep Neural Networks for Friction Coefficient Estimation", Sensors, vol. 20, no. 3, January 2020.
[30] I. Tang and T. P. Breckon, "Automatic Road Environment Classification", IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 2, pp. 476 – 484, June 2011.
[31] T. Zhao and Y. Wei, "Road Surface Image Dataset with Detailed Annotations", Mendeley Data, V2, July 2022, doi: 10.17632/w86hvkrzc5.2.