BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Publicat de
Universitatea Tehnică „Gheorghe Asachi” din Iaşi
Volumul 69 (73), Numărul 4, 2023
Secţia
ELECTROTEHNICĂ. ENERGETICĂ. ELECTRONICĂ
DOI:10.2478/bipie-2023-0022

REAL-TIME ASPHALT POTHOLE DETECTION USING THE YOLOv5 ALGORITHM AND THE NVIDIA JETSON NANO EMBEDDED BOARD

BY

MARIUS-EMANUEL OBREJA[1,*], DAN-MARIUS DOBREA[1,2] and RAREŞ-PETRU IBĂNIŞTEANU[1]

[1]“Gheorghe Asachi” Technical University of Iași,
Faculty of Electronics, Telecommunications and Information Technology, Iași, Romania
[2]Institute of Computer Science, Romanian Academy Iași Branch, Romania

Received: September 2, 2024
Accepted for publication: December 29, 2024

Abstract. The present scientific research focuses on the creation of a system for detecting potholes in asphalt to avoid traffic accidents. The motivation for choosing this topic of wide interest comes from the desire to apply the new modern technologies of artificial intelligence and deep learning in an extremely important field of road safety. The use of platforms and systems based on artificial intelligence in smart cities by correctly and quickly detecting potholes in asphalt is very important for autonomous vehicles. To achieve this goal, we used the YOLOv5 algorithm, one for real-time object detection, and the built-in Jetson Nano system, a powerful and efficient platform for AI applications. The final goal of this project is to develop a high-performance system that can recognize and classify potholes in asphalt from images and video streams in real time, with as much precision as possible and that can be used in smart cars.

Keywords: Object detection, artificial intelligence, deep learning, smart city, autonomous vehicles.

*Corresponding author; *e-mail*: marius-emanuel.obreja@academic.tuiasi.ro

1. Introduction

With the technological development of increasingly independent vehicles, the detection of potholes in the roadway is increasingly important. They provide essential information to drivers regarding the road, indicating traffic problems and sending warning messages related to possible dangers on the road. Ignoring asphalt potholes can lead to accidents or damage to cars. The rapid urbanization of our world seems to lead to an increase in the number of vehicles on public roads. Thus, the automatic detection of road defects can be defined as an impetuous necessity (Yu *et al*., 2024).

For the implementation of these ideas, we chose an embedded Jetson Nano platform from NVIDIA because it offers increased performance compared to other platforms that can detect objects in real time. It uses a software library optimized for this type of requirement and allows the implementation of learning and recognition models for the detection of potholes on the road. Training such a model involves challenges, such as variation in lighting conditions, size and frequency of pits, as well as other interferences such as reflections.

To implement this goal, we went through several essential stages: collecting and processing a relevant data set from the city of Iasi and its surroundings, training using the YOLOv5 model, implementing and optimizing the algorithm on the Jetson Nano, and finally, testing and evaluating the system's performance. The application of such a detection system has many practical benefits: - The field of autonomous vehicles: It helps vehicles to recognize and respect traffic rules, contributing to safe movement. – Advanced driver-assistance systems (ADAS): They can warn drivers about road defects, helping to prevent accidents and improve road safety. - Monitoring the road infrastructure: The authorities can use this information to check and maintain the road in optimal parameters. The main objective in carrying out this project was to show how artificial intelligence can improve road safety and bring concrete benefits in everyday traffic.

2. Theoretical foundations

This chapter introduces fundamentals of object detection, computer vision, convolutional neural networks, and the YOLO (You Only Look Once) algorithm. These concepts are essential to understanding how computers can analyze and interpret images, enabling the development of automated systems capable of detecting and recognizing objects in various applications, from autonomous vehicles to video surveillance and augmented reality.

Object detection represents a fundamental pillar in the evolution of modern technologies, having vast applications in various fields. In the context of autonomous vehicles, object detection is essential to ensure the safety and efficiency of navigation. Autonomous vehicles must identify and react quickly to

pedestrians, other vehicles, traffic sings and unforeseen obstacles (such as potholes), using advanced computer vision systems to make real-time decisions. Algorithms like YOLOv5 enable fast and accurate detection of these objects, helping to reduce the risk of accidents and improve traffic flow.

In recent years, the automotive field has managed to integrate the concept of autonomous driving, making significant progress that has transformed the way we perceive mobility. From conventional vehicles that require constant driver intervention, the automotive industry has evolved to develop cars capable of operating autonomously using advanced technologies and artificial intelligence. Self-driving vehicles are equipped with a diverse array of state-of-the-art sensors, video cameras, radars and LIDARs (light detection and ranging), all interconnected through sophisticated machine learning algorithms. These technologies enable vehicles to perceive and interpret their surroundings, make real-time decisions, and navigate safely without human intervention. In the context of rapid urbanization and the need to improve transport efficiency, autonomous driving promises innovative solutions to reduce traffic congestion and accidents (Balasubramaniam and Pasricha, 2022).

Computer Vision (CV), a branch of artificial intelligence (AI), studies the development of models and techniques that allow computers to interpret representations of the world with the goal of automating these tasks. A large part of its activities are related to the interpretation of image data, but there are also sub-domains that focus on automatic video-based interpretations. Through computer vision, it is possible to classify images (for example, recognizing obstacles / potholes for a autonomous vehicle) and even describe them in natural language, thus facilitating user interaction with devices without the need to know programming languages. In a general way, computer vision equips robotic systems with "eyes" to understand the world and perform various tasks.

Convolutional neural networks (CNNs) have had a major impact in the field of artificial intelligence, revolutionizing the way computers can understand and interpret visual data. CNNs have been instrumental in the development of image recognition technologies, object classification, face detection, and numerous other applications that require image and video analysis and processing. These networks are designed to automatically recognize complex visual features in image data, mimicking the way the human visual system processes information. By using convolutional layers, CNNs can extract meaningful features from images, such as edges, textures, and shapes (Kiiskilä M. and Kiiskilä P., 2024). The impact of these networks is evident in various fields, from automatic facial recognition in security and authentication to medical diagnostics through imaging and autonomous vehicles.

CNNs are made up of several specialized layers that work together to extract and analyze features from input data such as images. These layers are designed to automatically learn representations at different levels of granularity, from simple contours to complex structures, through a series of convolutional

operations, activations, and aggregations. The main layers used in CNNs include convolutional layers, activation layers, pooling layers, fully connected layers, and normalization layers.

YOLO (You Only Look Once) is a convolutional neural network specialized in object detection, renowned for its extremely fast execution time and impressive accuracy. This is accomplished by using a distinct neural network that simultaneously predicts the classes and bounding rectangles of objects in images. Due to its efficiency and accuracy, YOLO has become one of the most popular object detection systems. Over time, several versions of YOLO have been developed, each bringing improvements and advances in object detection. YOLOv5 (You Only Look Once) is a deep learning system with a single stage (one-stage detector) that detects objects using a convolutional neural network. The basic principle of this model is to use the full image as input and directly predict the bounding box and category to which the object belongs (Wu *et al*., 2024). YOLOv5, developed by Ultralytics, is available in five main variants: YOLOv5n (Nano), YOLOv5s (Small), YOLOv5m (Medium), YOLOv5l (Large) and YOLOv5x (Extra Large). Each of these models is optimized to meet different requirements in terms of inference speed, accuracy, and complexity.

NVIDIA Jetson Nano is a kit specially designed for AI applications, widely used in various fields. Its ability to run complex deep learning and CNN algorithms makes the Jetson Nano a preferred choice in research, education and commercial development (Bennett, 2024). Also, its efficient power consumption makes it ideal for applications that require an optimal combination of performance and resource economy, being suitable for IoT projects, embedded systems and other applications where energy efficiency is crucial. Similar to the Raspberry Pi, the Jetson Nano appeals to users due to its affordability and flexibility, making it a popular platform for developing DIY and educational projects.

3. Methodology

A first step in the neural network learning algorithm is the training of the data set for the identification of asphalt potholes. For this work, we used a class with different areas where road defects appeared in the county of Iasi. The images were taken with the phone, having a resolution of 3024x4032 pixels. They were later resized to 640x640 pixels to facilitate processing and training the neural network on a personal computer. In order to be able to train the desired neural network, it is essential that each image is labeled. This tagging must have the same name as the image and include details about the location of the object of interest and the category it belongs to. In the original dataset we had 10391 images and corresponding tags. The images were taken while driving the personal car from different angles, from different heights and on different backgrounds and at different driving speeds.

In YOLOv5, labels are essential to efficiently train the object detection model. Accurate labeling allows the model to learn to recognize and correctly locate objects of interest in an image. In the attached image, relevant details related to labeling and project implementation are described and explained. Below, in Fig. 1 is an example of tagging an image of a pothole.

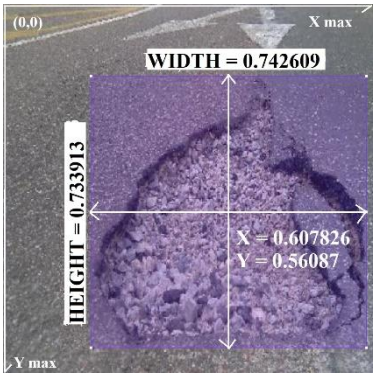

Fig. 1 – Pothole labeling.

In the image, we have a graphical representation of a labeled pit for training an object detection model. Here are the relevant details: 1. Coordinates of the reference system: - (0, 0) represents the upper left corner of the image; - X max and Y max represent the right and bottom limits of the image, respectively. 2. Relative dimensions of the pothole: - The width of the pothole is 0.742609 units (in relation to the total width of the image); - The height of the pothole is 0.733913 units (relative to the total height of the image). 3. Positioning the center of the pothole: - The center of the pit has the coordinates (X = 0.607826, Y = 0.56087), which indicates its position in the image.

In YOLOv5, each object in an image must be correctly labeled for the model to learn to detect objects effectively. Labeling includes: - Object class: The textual or numerical label indicating the type of object (in this case, the pit). - Bounding Box: A rectangle enclosing the object, defined by the coordinates of its center, relative width and height. Each .txt file associated with an image contains one or more lines, each line representing a tagged object. Each line is structured as follows in Fig. 2:

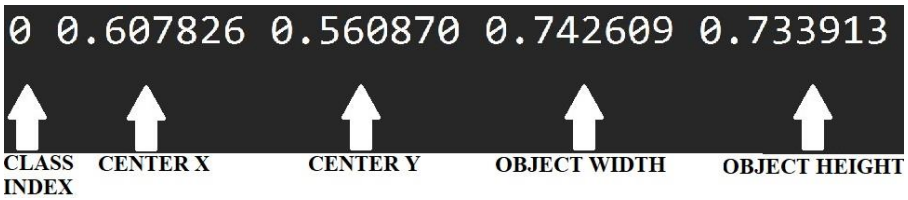

Fig. 2 – File data associated with the tagged image.

The .txt files are used to provide labeled data to the YOLOv5 model during training, allowing the model to learn to detect and localize objects based on these labels. These files are also used in the validation and testing stages to check the performance of the model by evaluating its ability to correctly detect labeled objects.

To obtain accurate results and improve the learning process of the YOLOv5 model, we performed extensive modeling of the dataset using various preprocessing techniques. These techniques aimed to simulate various lighting and environmental conditions to ensure robust model performance in different scenarios. First, we applied color adjustments to diversify the dataset. We used a variation of -3 to +3 in the hue parameter, which allowed us to change the hue of the images. Second, I adjusted the saturation parameter with values between -25% and +25%. This method helped us to simulate different lighting conditions, such as rainy days or cloudy skies, when color saturation can vary significantly. To simulate natural light variations, we applied brighten and darken techniques. These adjustments were made with values between -15% and +15%, allowing the model to learn to recognize objects in different lighting conditions.

In addition, we included noise in our images, to achieve a similarity with the natural imperfections of the photos, thus helping to train a more robust and accurate model in object identification. By applying these dataset preprocessing and modeling techniques, we were able to expand the volume of images and corresponding annotations which not only improved the accuracy of the model, but also ensured greater adaptability in various real-world operating conditions in real-time and different traffic conditions.

4. Results and Discussion

To train the model, we used a personal computer equipped with an NVIDIA 3070 GPU with 8 GB of dedicated memory, as this hardware offers clearly superior performance compared to the Jetson Nano. After preparing the dataset, we resized the images to 640x640 pixels as required by the YOLOv5 model. We created a YAML file where we defined all the object classes in the dataset. An epoch is a complete pass through the entire training data set. During each epoch, the YOLOv5 model adjusts its weights based on observed errors. We chose to train the model for 40 epochs to allow the model to learn enough from the available data, balancing training time and model performance. A batch is a subset of the training data processed at the same time by the model.

Training on batches instead of the entire data set allows for efficient memory usage and speeds up the training process. We selected a batch size of 16 to optimize GPU memory usage and ensure stable and efficient model convergence. We chose the YOLOv5-L version, which is an extended version of the model, optimized for higher accuracy at the expense of training time, but also larger model size and processing time, so it may affect real-time operation.

Training was performed on the GPU, using 2.85 GB of GPU memory. The training process took 9 hours, covering multiple epochs and batches to ensure optimal convergence and robust model performance. By using the NVIDIA 3070 GPU, we were able to significantly reduce the training time and improve the overall efficiency of the process, achieving an accurate and high-performance object detection model.

The YOLO model generates a number of key results and graphs that provide detailed information about its object detection performance and accuracy. These results include metrics such as accuracy, detection rate, and loss values during training. In addition, graphs are provided that illustrate the evolution of these metrics over time, allowing us to better analyze and understand the behavior of the model.

In Fig. 3 they are presented a series of plots that illustrate the evolution of different performance indicators and losses (losses) during the training and validation of the traffic pothole detection model: Train Box Loss; Train Object Loss; Train Class Loss; Metrics Precision; Metrics Recall; Validation Box Loss; Validation Object Loss; Validation class loss; mAP@0.5; mAP@0.5:0.95.

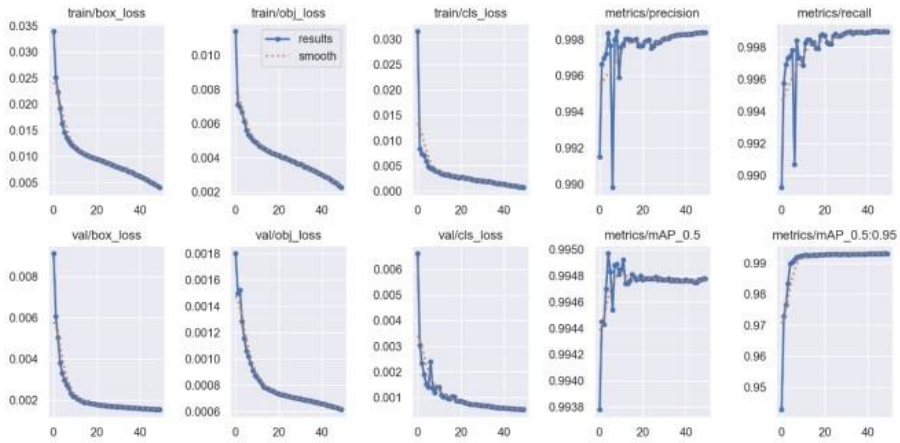

Fig. 3 – Evolution of different performance indicators and losses.

1. Train Box Loss (train/box_loss) represents the loss associated with the localization of the bounding boxes during training. A constant decrease in the loss is observed, indicating that the model becomes increasingly accurate in the localization of the bounding boxes during training.

2. Train Object Loss (train/obj_loss) represents the loss associated with the detection of objects during training. The loss gradually decreases, showing that the model becomes more efficient in detecting objects in images.

3. Train Classification Loss (train/cls_loss) represents the loss associated with correctly classifying objects during training. The loss decreases significantly, indicating an improvement in the model's classification accuracy.

4. Precision (metrics/precision) represents the accuracy of the model over the training epochs. We observe that the precision increases and stabilizes near the value 1.0, indicating a very high accuracy in correctly identifying traffic potholes.

5. Recall (metrics/recall) represents the recall of the model over the training epochs. Recall increases and stabilizes near the value 1.0, indicating a very high rate of correctly detecting traffic potholes.

6. Validation Box Loss (val/box_loss) represents the loss associated with locating the bounding boxes during validation. Similar to the training loss, we observe a constant decrease, which suggests a good generalization of the model.

7. Validation Object Loss (val/obj_loss) represents the loss associated with detecting objects during validation. The loss decreases similarly to the training loss, indicating consistency between training and validation.

8. Validation Classification Loss (val/cls_loss) represents the loss associated with correctly classifying objects during validation. The loss decreases, suggesting consistent classification performance during validation as well.

9. mAP@0.5 (metrics/mAP_0.5) represents the average precision (mAP) at a threshold of 0.5. mAP@0.5 increases and stabilizes near the value 0.995, indicating a very good performance of the model at this threshold.

10. mAP@0.5:0.95 (metrics/mAP_0.5:0.95) represents the average accuracy (mAP) over a threshold range from 0.5 to 0.95. The mAP over this range increases and stabilizes near the value 0.99, indicating excellent model performance over a wide range of thresholds.

In the specialized literature, these indicators have similar values for the YOLOv5 model as presented in: *A Comparative Study of YOLO-V5 Variants Performance for Object Detection* (Singht and Akoula, 2024). This is how the average of experimental results are presented to show that the YOLO-V5 medium variant network outperforms its other variants in terms of accuracy with 0.922 precision, 0.852 recall, 0.868mAP@0.5, and 0.649 mAP@0.5:0.95.

To ensure the efficient operation of the real-time YOLOv5 model on the Jetson Nano platform, we followed several essential steps. First, we set up the working environment on the Jetson Nano, installing all the necessary components, including YOLOv5, CUDA and other relevant libraries for GPU acceleration. This allowed full use of the Jetson Nano's hardware capabilities for fast image processing. After confirming the performance of the model on the PC, we transferred the trained model to the Jetson Nano and performed real-time testing. However, a major problem was the large size of the data set, which put considerable strain on the Jetson Nano, causing it to run very slowly. This sub-optimal performance highlights the limitations of the Jetson Nano hardware in handling very large data sets in real time. The model's performance on the dataset is the best evaluated on the server, having 1.28 fps and we will evaluate in the future what values we will obtain directly for the Jetson Nano board in real time.

For testing the implemented model, we used another 1535 different images with potholes and for the final validation we used another separate set of 1799 representations of images taken from traffic in real time. The results showed an accuracy rate of pothole identification in the rural area between 0.81 and 0.96, as highlighted in Fig. 4 and in the urban area between 0.81 and 0.95, as described in Fig. 5, the values also depending on the running speed (between 30 km/h and 50 km/h). At a lower running speed, we can achieve higher pothole identification performance.

Fig. 4 – Real time testing in rural area.

Fig. 5 – Real time testing in urban area.

5. Conclusion

This work has successfully demonstrated the applicability of deep learning technologies for asphalt pothole detection using YOLOv5 neural network and NVIDIA Jetson Nano hardware platform. Implementation and testing of the proposed system highlighted the efficiency and accuracy of the YOLOv5 model, which demonstrated high accuracy in detecting and classifying traffic potholes under various conditions. NVIDIA Jetson Nano hardware performance has shown that the platform can effectively support object detection algorithms, achieving a very good performance. That value provides adequate performance for road applications; although dataset size and model complexity may affect processing speed. The system has been successfully tested in various lighting conditions and environments, demonstrating its ability to maintain high performance, essential for practical use in autonomous or semi-autonomous vehicles.

In conclusion, the development and implementation of such a system can significantly contribute to increasing road safety by warning drivers in real time about potholes and helping autonomous vehicles to navigate safely. The collected data can be used for improving road infrastructure and optimizing traffic management.

Future directions for this project include optimization of the model and hardware resources to improve processing speed and system efficiency, expansion of the dataset to include images of pits from various regions and conditions, and integration with other systems of driver assistance to create a complete ecosystem for smart vehicles. Carrying out extensive validation studies in real conditions, on various types of roads and in different weather conditions, will provide a more accurate assessment of the performance and reliability of the proposed system, thus contributing to the safety and efficiency of road traffic.

REFERENCES

Balasubramaniam A., Pasricha S., *Object Detection in Autonomous Vehicles: Status and Open Challenges*, Colorado State University, January 2022, https://arxiv.org/abs/2201.07706, accessed on 22.08.2024.

Bennett J., *Smart AI Pothole Detector*, https://developer.nvidia.com/embedded/community/jetson-projects/ai_pothole_detector, accessed on 22.08.2024.

Kiiskilä M., Kiiskilä P., *Evolving Deep Architectures: A New Blend of CNNs and Transformers Without Pre-training Dependencies*, In: Fred A., Hadjali A., Gusikhin O., Sansone C. (eds) *Deep Learning Theory and Applications*, DeLTA, 2024. Communications in Computer and Information Science, vol 2171. Springer, Cham. https://doi.org/10.1007/978-3-031-66694-0_10, accessed on 22.08.2024.

Singht M., Akoula A., *A Comparative Study of YOLO-V5 Variants Performance for Object Detection*, 2024 IEEE 5th India Council International Subsections Conference (INDISCON), Chandigarh, India, https://ieeexplore.ieee.org/document/10744406.

Wu W., Zou X., Fang Z, Fang W., Song X., Yang A., Liu Z., *Research on Asphalt Pavement Crack Detection using YOLOv5 Model*, 2024 7th International Conference on Advanced Algorithms and Control Engineering (ICAACE), March 2024, Shanghai, China.

Yu J., Jiang J., Fichera S., Paoletti P., Layzell L., Mehta D., Luo S., *Road Surface Defect Detection–From Image-Based to Non-Image-Based: A Survey*, IEEE Transactions on Intelligent Transportation Systems (Early Access), April 2024, 1-23.

DETECȚIA DE GROPI DIN ASFALT ÎN TIMP REAL FOLOSIND ALGORITMUL YOLOv5 ȘI PLACA ÎNCORPORATĂ NVIDIA JETSON NANO

(Rezumat)

Prezenta cercetare științifică se concentrează pe crearea unui sistem de detectare a gropilor în asfalt pentru evitarea accidentelor de circulație. Motivația pentru alegerea acestei teme de larg interes vine din dorința de a aplica noile tehnologii moderne de inteligență artificială (IA) și de învățare profundă într-un domeniu extrem de important al siguranței rutiere. Utilizarea platformelor și sistemelor bazate pe inteligență artificială în orașele inteligente prin detectarea corectă și rapidă a gropilor din asfalt este foarte importantă pentru vehiculele autonome. Pentru a atinge acest obiectiv, am folosit algoritmul YOLOv5, unul pentru detectarea obiectelor în timp real și sistemul Jetson Nano încorporat, o platformă puternică și eficientă pentru aplicații IA. Scopul final al acestui proiect este de a dezvolta un sistem performant care poate recunoaște și clasifica gropile din asfalt din imagini și fluxuri video în timp real, cu cât mai multă precizie și care să poată fi folosit la mașinile inteligente.