Journal of Automation, Electronics and Electrical Engineering

THE CONCEPT OF A MOBILE SYSTEM FOR DETECTION FIRE PHENOMENA BASED ON CONVOLUTIONAL NEURAL NETWORKS

Sebastian TATKO

Military University of Technology, Faculty of Automation, Electronics, Electrical Engineering and Space Technologies, gen. Sylwestra Kaliskiego 2, 00-908 Warsaw, Poland, sebastian.tatko@wat.edu.pl

DOI: https://doi.org/10.24136/jaeee.2023.006

Abstract – The research problem taken up in the article is the development of an efficient, mobile and effective fire detection algorithm based on the architecture of artificial neural networks. Both the process of training and inference of CNNs is burdened with a high demand for computing power. In the case of desktop devices, equipped with powerful processors and graphics cards, this process is largely facilitated and does not cause great difficulties. Another situation, however, is the desire to create a detection algorithm that in its performance will not differ from the stationary version, nevertheless its additional feature will be mobility. The desire to supervise vast areas of critical infrastructure using an unmanned aerial vehicle, imposes peculiar hardware limitations, which mainly include weight and size. The creation of an algorithm that will carry out real-time fire detection under the above-mentioned assumptions will therefore be a task that will require the optimization of a trained neural network model, into a format supported by popular mobile systems such as the *Raspberry Pi*.

Key words – drone, fire detection, convolutional neural network, unmanned aerial vehicle, *Coral AI*

INTRODUCTION

Exposure to human life, damage to infrastructure and the environment are just a few of the consequences of fire phenomena. Increasingly dynamic industry, a warming climate and people's carelessness mean that fires are occurring with increasing frequency. In 2022, firefighters recorded more than 608,000 interventions, the highest number of incidents recorded in the 30-year history of the National Fire Service. This is 29,102 more interventions than in 2021. In addition, firefighters intervened 21% more often than five years ago, when 502,055 incidents were recorded, and 36% more times than in 2016, when 447,003 incidents occurred [1]. With the goal of early detection and increased safety, more and more effective fire alarm systems (FAS) are being developed. A wide range of detectors, thermal cameras, and building surveillance systems are making the surveillance of facilities an increasingly efficient process [2]. Complications arise when there is a need for surveillance of vast spaces such as forests.

Traditional methods of detecting forest fires are mostly based on the detection of hazardous phenomena using human patrols, smoke detectors, thermal sensors, observation towers equipped with optoelectronic cameras, satellite imagery and patrols by conventional aircraft. Smoke detectors and thermal sensors in their principle of operation are characterized by pointiness. Their use cannot reliably provide information on the location and size of a fire. Human patrols are characterized by a limited field of vision, the need to perform shifts due to the need for the human body to recover, and a high rate of false alarms. Other methods of detecting fires over wide areas also include the use of satellite imagery. Satellites provide comprehensive coverage of the area, but nevertheless the resolution of the images and the ability to observe fire hazards in real time pose major problems, at the same time making this system ineffective [3].

In this article, the conceptual fire detection system is based on artificial intelligence algorithms and the system medium of an unmanned aerial vehicle.

1 ARTIFICIAL INTELLIGENCE VERSUS FIRE DETECTION

The original goal of the neural network approach was to create a computing system that could solve problems just like the human brain. Over time, engineers and scientists have led to the rapid development and evolution of artificial intelligence. Creating facial images of non-existent people, *chatGPT*, autonomous devices are just examples of the use of artificial intelligence algorithms [4].

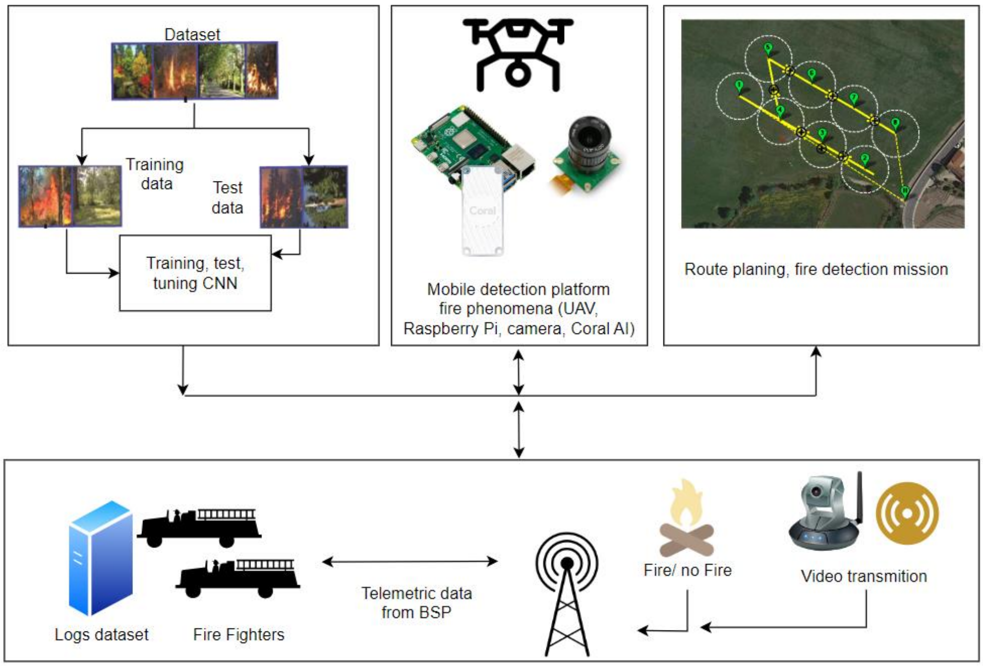

Fig. 1. Conceptual scheme of the fire identification and detection system (in-house development)

One of the chapters of machine learning is convolutional neural networks (*CNNs*). The presented idea of fire detection assumes in its operation the use of an unmanned aerial vehicle equipped with a computational system with the necessary sensors and a flame detection algorithm. The implementation of the system presented in Figure 1 belongs to a very complex task, the first step of which is the creation of a smoothly working detection algorithm based on possibly lightweight and energy-efficient components.

The process of building neural networks from scratch is one of the difficult and very time-consuming tasks. To create their own CNNs, engineers create pre-trained neural models on huge training databases. The ability to choose the right architecture [5] of a neural network, having an efficient computing unit and a large database of images are the basic factors that can result in an effective detection algorithm. In the case of creating a detection algorithm for fire phenomena, the first problem that arises is the collection of a suitable database of images. The movement of the unmanned aircraft at a certain height requires the use of a database of photos of burning areas taken from a bird's eye view. Publicly available databases do not offer large resources of the required images, nevertheless, current augmentation techniques [6,7] will allow the construction of a sufficient database. Another major obstacle is the hardware implementation of a system that will be as efficient as possible to run the inference process of an extended neural network while ensuring compactness, lightness, and energy efficiency. In order to develop a technique for dealing with the desire to implement a fire detection algorithm, Chapter 3 presents speed studies of various hardware and software architectures. Due to the limitations of the ability to cause a controlled fire, the research was conducted on a neural network with SSD MobileNet V2 FPNLite 640x640 architecture trained on images containing images of three people. It should be emphasized that this article does not focus on creating the most accurate CNN possible for detection. The first phase of the ongoing research is to test the speed and correctness of the neural network on the *Raspberry Pi* chip. Confirmation of expectations and the validity of the developed concept will result in the implementation of the fire detection algorithm on a real system.

```
INFO:tensorflow:Step 100 per-step time 1.657s
I0727 15:47:53.671432 13868 model_lib_v2.py:705] Step 100 per-step time 1.657s
INFO:tensorflow:{'Loss/classification_loss': 0.31943825,
 'Loss/localization_loss': 0.13623703,
 'Loss/regularization_loss': 0.15187965,
 'Loss/total_loss': 0.6075549,
 'learning_rate': 0.0319994}
I0727 15:47:53.686922 13868 model_lib_v2.py:708] {'Loss/classification_loss': 0.31943825,
 'Loss/localization_loss': 0.13623703,
 'Loss/regularization_loss': 0.15187965,
 'Loss/total_loss': 0.6075549,
 'learning_rate': 0.0319994}
```

Fig. 2. The process of training a neural network with SSD MobileNet V2 FPNLite 640x640 architecture.

2 INCREASE SYSTEM PERFORMANCE, CORAL AND QUANTIZATION

The process of training and inference of neural networks is one of the tasks charged with a high demand for computing power. Given the performance of processors, graphics cards and the ability to use *Google Colab* GPU, the process of training the network is not one of the most difficult tasks. The computing power in the network learning phase only affects the time in which the model will be properly trained. For the real-time inference process of a trained CNN tasked with image detection, the situation requires a different approach. The shortage of computing power, the incorrectness of the model architecture will directly affect the speed and correctness of object detection. Due to the desire to create a mobile fire detection system, it was decided to use the *Raspberry Pi* platform. The most important parameters of the chip that contributed to its selection were its weight and compatibility with a number of external chips. Additional components used in the study of the layout were the UC-325 camera with a resolution of 1920x1080 pixels and the *Coral Edge* TPU accelerator chip.

Many solutions in the currently available literature present tests conducted only in the experimental process, i.e., detection is carried out on uploaded images on the other hand there are hardware solutions with low efficiency. Despite the high potential of *Raspberry Pi* chips, it was additionally decided to test Google's accelerator chip. *Coral* is a chip with a built-in Edge TPU coprocessor, which is capable of performing 4 trillion operations per second (*TOPS*), while consuming 0.5 watts per TOPS. The system is capable of operating a neural network model such as MobileNet v2, for example, at nearly 400 frames per second in a very energy-efficient manner. Considering the above data presented by the manufacturer [8], the chip forms the basis of a conceptual fire detection system. However, it should be noted that the process of running a custom model on the *Coral* chip is not a simple task and carries many requirements.

The detection system described in the previous chapter should be modified accordingly in order to run it on the real system. The resulting output neural network file is *Tensorflow Lite (.tflite).* The Lite version is a lighter version of the full *Tensorflow* library, whose models achieve low-latency inference in much smaller binary models. *Tensorflow* models can be made even smaller and more efficient by the quantization process.

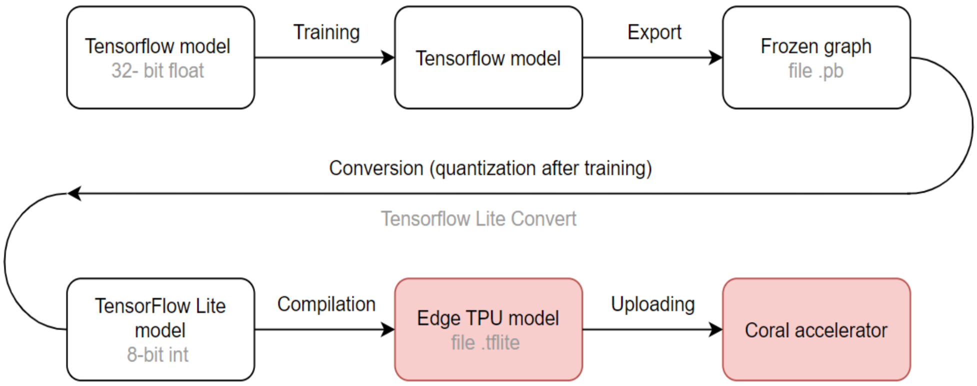

Fig. 3. Diagram of the process of creating a basic model compatible with the *Edge TPU* coprocessor (own elaboration based on [2])

Quantization refers to the process of reducing the number of bits representing a number. Deep neural networks typically contain tens or hundreds of millions of weights, which are represented by very precise numerical values. Working with such numbers requires significant computing power, bandwidth and memory. Quantization optimizes the deep learning model by representing model parameters with low-precision data types, such as *int8* and *float16*, without incurring a significant loss of accuracy. Storing model parameters with low-precision data types not only saves bandwidth and memory, but also speeds up computation [9]. Converting parameters to 8 - bit contributes to reducing the size of the model, for example, quantizing a 200 MB Alexnet model contributes to a 50 MB file size. For the test model, the reduction in file size takes on a value of about 70%. Downloading numbers in 8-bit format from RAM requires only 25% of the bandwidth of the standard 32-bit format; moreover, CNN quantization results in speeding up the inference process by 2 to 4 times. Fast arithmetic is one of the biggest benefits of the described process.

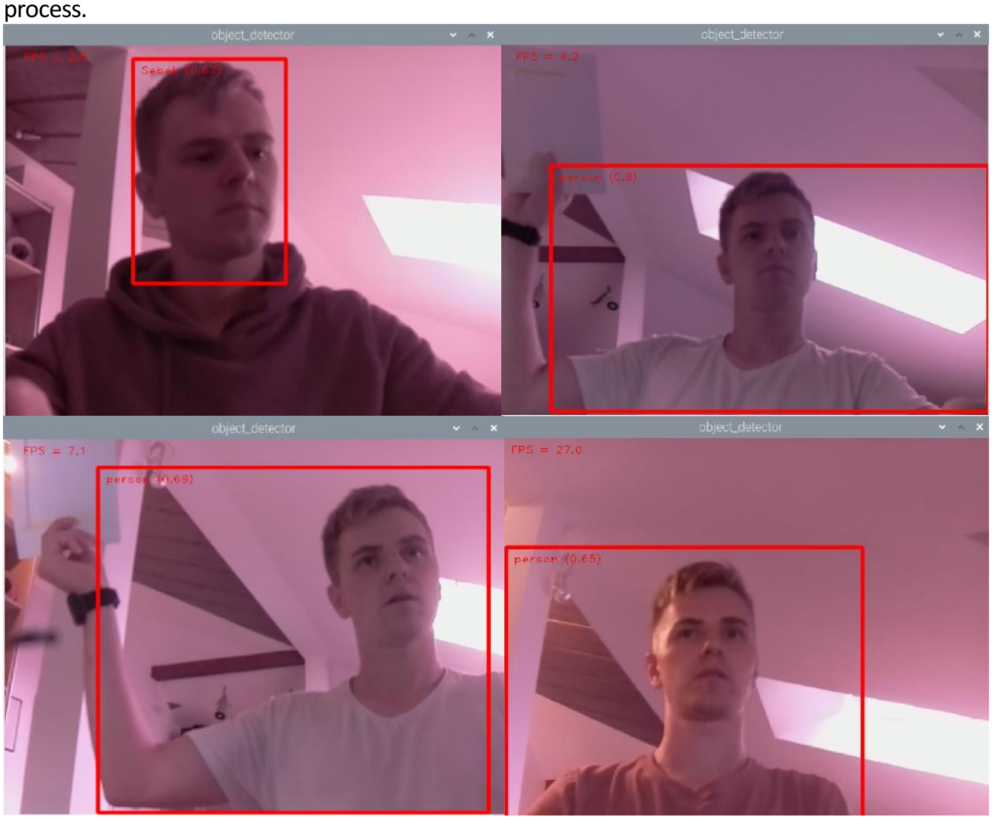

Fig. 4. The detection rate of a person depending on the hardware configuration and the occurrence of the quantization process.

The research presented in Figure 4 focuses on checking the speed and correctness of the test CNN. The scale of the program's speed, and thus its ability to detect an object as quickly as possible, is displayed in the upper left corner, and is measured in frames per second (FPS). The object to be detected, which in this case is a person, is marked with a red frame. The value contained in the detection frame is the probability of correctly detecting an object of the person class, the scale of this value is 0-1. All the tests carried out were performed under the same conditions. The parameters of the camera were not adjusted to the prevailing lighting in the room, but it should be noted that in the conducted tests the main importance was attached to the influence of the system configuration on the detection speed. The first circuit tested was a *Raspberry Pi3 A*. The performance of the system is not satisfactory in any respect. The processor of the system, after starting the neural network, quickly reached a high temperature thus, after several seconds, the system shut down automatically. Detection of a person, however, was recorded at 2.5 FPS. According to the documentation, the safe threshold temperature of the Pi3 A processor is 60°C, exceeding this value results in throttling the inference process. The third-generation *Raspberry Pi* model A chip does not work with the *Coral* accelerator, which is due to the lack of a USB 3.0 interface.

Increased RAM by 3 GB, faster processor clock speed by 7.14% are the key figures *of Raspberry Pi 4B* with respect to its older precursor. Inference of the neural network model based on this subassembly reached a speed of about 4.2 FPS. The detection speed of 5 frames per second in its operation results in a very slow response to changes and high latency. The circuit in this setup performed stably while its operating temperature took values within the manufacturer's catalos data.

Table 1. Detection speed depending on system configuration and program

Layout and program setup	Detection speed (FPS)
Raspberry Pi A	2 – 2.5 (overheating CPU)
Raspberry Pi 3A + Coral	No possibility to launch Coral
Raspberry Pi 4B	4.2 - 4.7 FPS
Raspberry Pi 4B partial quantization	6.9 – 7.5 FPS
Raspberry Pi 4B fully quantized model (compatible with Edge TPU)	27 – 30 FPS

Performing the quantization process on the base model resulted in a reduction in file size and an acceleration in object detection speed. Unfortunately, the acceleration of 2.5 FPS compared to the previous configuration does not have a significant effect and the image continues to be visibly framed. The reason for the relatively small acceleration of the neural network model by the *Coral* accelerator is its incomplete quantization. Incomplete or insufficient quantization means that the CNN architecture currently under study uses built-in functions that are not among the possibly quantized ones. The solution to the described problem is to change the configuration file of the test neural network or to use CNN architectures that are compatible with the *Edge TPU*, which was provided by the producer. In order to illustrate the full performance of the *Raspberry Pi 4B* circuit in combination with the *Coral* accelerator, a test of the speed of the architecture that is the MobileNet v1 network pre-trained on the *COCO 2017* dataset was performed. The achieved speeds of such a solution reach 30 frames per second, while ensuring the smoothness of detection and the speed of response of the CNN to any changes.

3 CONCLUSION

The quantization process of the author's neural network does not always involve achieving the intended goal. The desire to speed up the test model involves the need to change functions not supported by the *Edge TPU*. Nevertheless, it should be noted that there is a set of CNN architectures compatible with *Coral*, which in their operation should guarantee full use of the external processor. However, the presented research results indicate that the creation of a compatible and energy-efficient detection system is as possible. The next stage of the ongoing work will be to use the collected database of fire images to train and tune a neural network compatible with the presented system. The mobile detection platform will eventually find its application on an unmanned aerial vehicle.

BIBLIOGRAPHY

[1] https://www.gov.pl/web/kgpsp/interwencje-podsumowanie-2022, dostęp: 25.09.2023 r.

[2] A. Gaur et al., "Fire Sensing Technologies: A Review," in IEEE Sensors Journal, vol. 19, no. 9, pp. 3191-3202, 1 May1, 2019, doi: 10.1109/JSEN.2019.2894665

[3] RS Allison i in., „Airborne Optical and Thermal Remote Sensing for Wildfire Detection and Monitoring", tom. 16, (8), 2016

[4] T. Wu et al., "A Brief Overview of ChatGPT: The History, Status Quo and Potential Future Development," in IEEE/CAA Journal of Automatica Sinica, May 2023, doi: 10.1109/JAS.2023.123618

[5] P. Nagababu, K. Dhakshitha, G. Chandrika and U. R. Chowdary, "Automated Fire Detection System Using Image Surveillance System (ISS) and Convolutional Neural Networks (CNN)," 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2023

[6] A. Jellali, I. B. Fredj and K. Ouni, "Data Augmentation for Convolutional Neural Network DeepFake Image Detection," 2023 IEEE International Conference on Advanced Systems and Emergent Technologies (IC_ASET), Hammamet, Tunisia

[7] Jakubowski, K., Paś J., Duer, S., Bugaj, J.: Operational Analysis of Fire Alarm Systems with a Focused, Dispersed and Mixed Structure in Critical Infrastructure Buildings. Energies. 2021; 14, 7893. https://doi.org/10.3390/en14237893.

[8] C. Yuan, Y. M. Zhang, and Z. X. Liu, "A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques," Canadian Journal of Forest Research, vol. 45, no. 7, 2015, pp. 783–792.

[9] F. Yizhou, M. Hongbing. Video-based Forest fire smoke recognition, Journal of Tsinghua University. 2015, 55(2): 243-250, 256.

[10] Paś J., Exploitation of electronic security systems, Military University of Technology, Warsaw 2023.