ISSN: 2582-4686 SJIF 2021-3.261,SJIF 2022-2.889
ResearchBib Impact Factor: 8.848 / 2023

THE MULTIDISCIPLINARY JOURNAL OF SCIENCE AND TECHNOLOGY

VOLUME-3, ISSUE-3

Traffic Signs Recognition System Development

Vladyslav Nikitin, Svitlana Maksymova, Vladyslav Yevsieiev

Department of Computer-Integrated Technologies, Automation and Robotics, Kharkiv National University of Radio Electronics, Ukraine

Abstract

In this article, the authors consider the feasibility of developing systems for recognizing road signs, as well as road markings, registration and recognition of obstacles, and other traffic objects. The authors consider the possibility of using artificial neural networks to solve this problem. In particular, they propose using a multilayer perceptron to achieve sufficient recognition accuracy. In the future, it is planned to develop software for the implementation of the developed road sign recognition system.

Key words: Tenserflow, Machine learning, Object recognition, Usability, Computer Vision.

Introduction

Computer vision systems are of great importance in recognizing objects of various categories and types [1]-[13].

As the autonomy of mobile devices increases, systems that make it possible to recognize signs indicating the direction of movement, as well as rules of movement, in other words, road signs, are becoming increasingly important. Among them we can distinguish systems that allow recognizing road signs that comply with traffic rules. At the same time, we must understand the increased attention to traffic safety, since such traffic occurs on highways with the participation of a large number of people. It should be noted that such research is extremely timely. They must include recognition of road signs, road markings, obstacles on the roads, as well as other road users.Accordingly, the accuracy and speed of recognition should be close to the reaction speed of the human body, and it is better if these indicators are improved.

Today, there are approximately 169.5 thousand km of state roads in Ukraine. The network of main routes is spread throughout the country and connects all major cities of Ukraine, as well as provides cross-border routes with neighboring countries.

Due to such an increase in cars on the roads of Ukraine, the number of offenses is increasing, according to statistics, one offense or traffic accident occurs every 15 minutes [14]-[16].

For assistance and control of traffic signs, navigation systems are most often used, such systems use pre-loaded maps. Usually, such navigation systems use a touch screen to control the device, the quality of the screen in navigation systems is much worse than in modern smartphones, except for the most modern systems, but usually their cost is very high. Another exception may be untimely updating of traffic signs, or termination of support by the manufacturer, as well as outdated maps in such products [17], [18].

Also, navigation systems integrated into the car are used to help drivers, such systems can be supplemented with external cameras that analyze the situation around the car in real time, and can warn the driver if the speed limit is exceeded, relying on traffic signs. However, such systems

ISSN: 2582-4686 SJIF 2021-3.261,SJIF 2022-2.889
ResearchBib Impact Factor: 8.848 / 2023

do not have a mass character in the automotive world, and are most often installed on cars of the highest class, of which there are few on the roads.

Thus, taking into account all of the above, the research topic related to the development of a system that will recognize road signs is relevant and extremely timely.

Related works

Due to the constant increase in traffic, the demand for automated or even automated traffic sign monitoring systems is constantly increasing. That is, the development of systems that can recognize road signs and then transmit the corresponding instructions to the control systems of certain devices is becoming more and more relevant. In [19] authors prove that currently there are all necessary conditions to develop control systems of traffic regulations based on artificial intelligence with logical reasoning. Let's look at at least some of the latest research on this topic.

Campbell, A and co-uthors in [20] explore the possibility of using deep learning to produce an autonomous system for detecting traffic signs on GSV images to assist in traffic assets monitoring and maintenance.

Many scientists are trying to recognize at least traffic light signs. And also control traffic lights using artificial intelligence methods. For example, in [21] authors propose to create a new traffic simulator CityFlow with fundamentally optimized data structures and efficient algorithms. They say that besides traffic signal control, CityFlow could serve as the base for other transportation studies and can create new possibilities to test machine learning methods in the intelligent transportation domain.

Lee, W. H. & Chiu, C. Y. [22] designed and implemented their smart traffic signal control (STSC) system. It supports several smart city transportation applications including emergency vehicle signal preemption (EVSP), public transport signal priority (TSP), adaptive traffic signal control (ATSC), eco-driving supporting, and message broadcasting.

In [23] researchers note that a core feature of autonomous vehicle systems is the identification of the traffic sign. They implement the spatial pyramid pooling (SPP) principle to boost Yolo V3's backbone network for the extraction of functionality. Their work uses SPP for more comprehensive learning of multiscale object features.

Authors [24] note that the automatic traffic sign detection and recognition system is very important research in the development of advanced driver assistance systems. They show current issues and challenges of the existing technologies with brief suggestions and a discussion on the progress of driver assistance system research in the future.

Based on the analyzed literature, we can conclude that road sign recognition systems can be complex, that are able to recognize not only road signs, but also markings, obstacles, and other road users. And there are also simple systems that are capable of recognizing only one type of object, for example, a traffic light signal.

Later in this article we will look at a road sign recognition system based on a technical vision system.

The traffic sign recognition system development

Various technologies are suitable for recognizing road signs. One of the most popular OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning library.

ISSN: 2582-4686 SJIF 2021-3.261,SJIF 2022-2.889
ResearchBib Impact Factor: 8.848 / 2023

The library contains more than 2,500 optimized algorithms, including a complete set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in video, track camera movements, track moving objects, extract 3D object models, create 3D point clouds from stereo cameras, connect images to obtain high resolution. image of an entire scene, find similar images in an image database, remove red-eye from flash images, track eye movements, recognize scenery and set markers to overlay it with augmented reality, and more. OpenCV has a community of over 47,000 users and an estimated number of downloads exceeding 18 million. The library is widely used by companies, research groups, and government agencies.

OpenCV has C++, Python, Java, and MATLAB interfaces and supports Windows, Linux, Android, and Mac OS, and is also mostly targeted at real-time vision applications and takes advantage of MMX and SSE instructions when available.

With the help of OpenCV, it is possible to create applications that will qualitatively recognize road markings and work on many platforms. At the same time, the problem arises that for each device you need to create your own application and use different technologies, which is a very time-consuming process that requires in-depth knowledge of various programming languages. Also, with the help of OpenCV, it is not possible to create an application for devices based on IOS.

The solution to this problem is the creation of a web application. With the help of this approach, it is possible to solve several problems at once. Today, most of the devices that people use every day have access to the Internet using browsers. Having created a web application, it becomes possible to use the same application, regardless of which operating system a person uses. That is, the developer must possess only one set of skills, and close the need for users of any devices. Another problem that the creation of a web application solves is the problem of memory. The browser is installed by default on most devices, the user does not need to download any additional applications, only have the address to the web application, which does not take up additional space on the device, and always works with the latest version available.

To solve this problem, you can use TensorFlow.js, an open source library that can be used to define, train and run machine learning models entirely in the browser using Javascript and a high-level API.

The use of the TensorFlow.js model has grown exponentially over the past few years, and many JavaScript developers are now looking to take existing state-of-the-art models and retrain them to work with user data unique to their industry. The act of taking an existing model (often called a base model) and using it in a similar but different domain is known as transfer learning.

Transfer learning has many advantages over starting learning from a completely blank model. You can reuse the knowledge gained from the previously trained model and you will need fewer examples of the new item you want to classify. In addition, training is often much faster because only the last few layers of the model architecture need to be retrained instead of the entire network. For this reason, transfer learning is very well suited for a web browser environment where resources may vary depending on the execution device, but also has direct access to sensors for easy data collection.

Client side in a web browser using vanilla JavaScript:

ISSN: 2582-4686 SJIF 2021-3.261,SJIF 2022-2.889
ResearchBib Impact Factor: 8.848 / 2023

– Server side and even IoT devices like Raspberry Pi that use Node.js;

– Desktop applications using Electron;

– Native mobile apps using React Native.

TensorFlow.js also supports multiple backends in each of these environments (the actual hardware environments in which it can run, such as CPU or WebGL. "Backend" in this context does not mean a server-side environment - the backend for execution can be on client-side, such as in WebGL) to ensure compatibility as well as ensure fast performance. TensorFlow.js currently supports:

– running WebGL on the device's graphics card (GPU) is the fastest way to run larger models (larger than 3MB) with GPU acceleration;

- web assembly execution (WASM) on the CPU - to improve CPU performance on various devices, including, for example, old-generation mobile phones. This is better for smaller models (less than 3MB in size), which can actually run faster on the CPU with WASM than with WebGL due to the overhead of loading content to the GPU;

– CPU execution is a backup option if none of the other environments are available. This is the slowest of the three, but always handy.

Running TensorFlow.js in a web browser on a client machine can lead to several benefits.

- Privacy, it is possible to train and classify data on the client machine without sending the data to a third-party web server.

- Speed, because there is no need to send data to a remote server, inference (the act of classifying data) can be faster. Even better, there is direct access to device sensors such as camera, microphone, GPS, accelerometer, etc. if the user grants access.

- Anyone in the world can click on the link, open the web page in their browser and use the app. There's no need for complex server-side Linux setup with CUDA drivers and more to just use a machine learning framework.

- No server costs mean the only thing you need to pay for is a CDN to host your HTML, CSS, JS and model files. The cost of a CDN is much cheaper than keeping a server (perhaps with a connected video card) running 24/7.

Server functions, using the TensorFlow.js implementation in Node.js, provide many features. Full CUDA support, server-side, for GPU acceleration, you must install NVIDIA CUDA drivers to enable TensorFlow to work with the GPU (unlike a browser that uses WebGL - no installation required). However, with full CUDA support, you can fully utilize the capabilities of your lower-level graphics card, resulting in faster learning and inference times. Performance is on par with the Python implementation of TensorFlow because they both use the same C++ server.

For the latest research models, it is possible to work with very large models, perhaps gigabytes in size. These models cannot currently be run in a web browser due to memory usage limits for each browser tab. To run these larger models, you need to run Node.js on your own server with the hardware specifications required to run such a model efficiently.

IOT, Node.js is supported on popular single-board computers such as the Raspberry Pi, which in turn means that it is possible to run TensorFlow.js models on such devices.

Node.js is written in JavaScript, which means it benefits from compile-time. This means that you can see performance gains when using Node.js because it will be optimized at runtime, especially for any pre-processing that is done.

ISSN: 2582-4686 SJIF 2021-3.261,SJIF 2022-2.889
ResearchBib Impact Factor: 8.848 / 2023

Running ML(machine learning) in the browser means that there is no need to install libraries or drivers from the user's point of view. Just open a web page and your app is ready to run. In addition, it is ready to work with GPU acceleration. TensorFlow.js automatically supports WebGL and will accelerate your code as soon as the GPU is available. Users can also access a web page from a mobile device, in which case your model can use data from sensors, such as a gyroscope or accelerometer. Importantly, all data remains with the client, making TensorFlow.js useful for low-latency output as well as privacy-preserving applications.

There are three workflows you can use when working with TensorFlow.js:

– it is possible to import an existing, pre-trained model for output. If an existing TensorFlow or Keras model that was previously trained offline is available, it can be converted to TensorFlow.js format and loaded into the browser for output;

– the ability to use transfer learning to supplement an existing model trained offline using a small amount of data collected in the browser using the Image Retraining method. This is one way to quickly train an accurate model using only a small amount of data;

- in-browser model generation with TensorFlow.js to fully define, train, and run models in the browser using Javascript and high-level API layers.

TensorFlow.js also includes a low-level API (formerly deeplearn.js) and Eager execution is shown in Figure 1

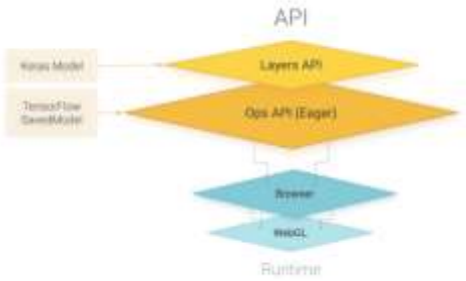

Figure 1: TensorFlow.js Structure

TensorFlow.js is based on WebGL and provides a high-level API for defining models and a low-level API for linear algebra and automatic differentiation. Tensor Flow.js supports importing TensorFlow SavedModels and Keras models.

TensorFlow.js, a JavaScript ecosystem for machine learning, is the successor to deeplearn.js, now called TensorFlow.js Core. TensorFlow.js also includes the Layers API, which is a high-level library for building machine learning models using Core, as well as tools for automatically migrating TensorFlow SavedModels and Keras hdf5 models.

At work, transfer learning is used, it involves the use of already acquired knowledge to help learn different, but similar things. Humans do this all the time, there are a bunch of neurons in the brain that know how to identify tree-like objects, and other neurons that are good at finding long straight lines. This can be reused to quickly classify a willow tree, which is a tree-like object with many long, straight, vertical branches. Similarly, if there is a machine learning model already trained on a domain, such as image recognition, it is possible to reuse it for a different but related task.

ISSN: 2582-4686 SJIF 2021-3.261,SJIF 2022-2.889
ResearchBib Impact Factor: 8.848 / 2023

This is exactly what can be done with an advanced model like MobileNet, which is a very popular research model that can perform image recognition on 1000's of different object types. From dogs to cars, it was trained on a huge dataset known as ImageNet, which contains millions of labeled images.

During training, this model has learned to pick out the common features important to all of these 1,000 objects, and many of the lower-level features it uses to identify such objects can be useful for detecting new objects that it has never seen before. saw After all, in the end, everything is just a combination of lines, textures, and shapes.

Looking at a traditional Convolutional Neural Network (CNN) architecture (similar to MobileNet) shows how transfer learning can use this trained network to learn something new. The image below shows a typical architecture of a CNN model, which in this case was trained to recognize the handwritten digits 0 to 9 in Figure 2.

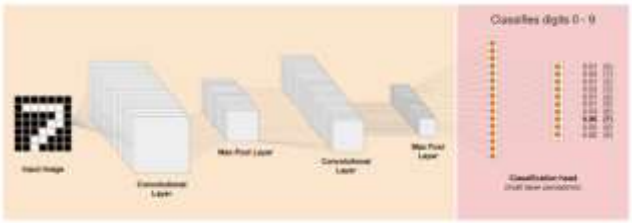

Figure 2: Traditional Convolutional Neural Network Architecture

If it were possible to separate the pre-trained lower-level layers of an existing trained model, as shown on the left, from the end-of-model classification layers shown on the right (sometimes called the model's classification head), then using the lower-level layers to generate output functions for any given image based on the raw data it was trained on. The same network with removed classification head is presented in Figure 3.

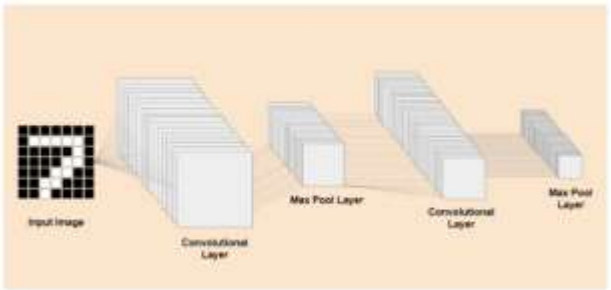

Figure 3: Network with Removed Classification Head

Assuming that the new subject being recognized can also use the same input features that the previous model trained, then there is a good chance that they can be reused for a new purpose.

In the diagram above, this hypothetical model was trained on numbers, so what it already learned about numbers can also be applied to letters like a, b, and c.

So now we can add a new classification head that will try to predict a, b or c instead, as shown in Figure 4.

ISSN: 2582-4686 SJIF 2021-3.261,SJIF 2022-2.889
ResearchBib Impact Factor: 8.848 / 2023

THE MULTIDISCIPLINARY JOURNAL OF SCIENCE AND TECHNOLOGY

VOLUME-3, ISSUE-3

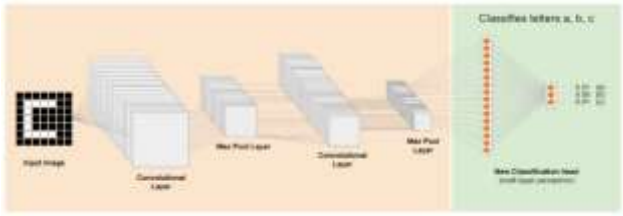

Figure 4: New Clasiification Head

Here, the lower-level layers are frozen and untrained, only the new classification head will be updated to learn the features provided from the pre-trained sliced model on the left. The act of doing this is known as transfer learning, and that's what the Teachable Machine does behind the scenes. It can also be seen that the multilayer perceptron at the very end of the network trains much faster than if you had to train the entire network from scratch.

Conclusion

In the course of the work, an analysis was made of modern computer vision systems and software for their development.

This article discusses the relevance and feasibility of developing a road sign recognition system. Various types of such systems are presented and analyzed. It has been established that there are complex systems that are able to recognize not only road signs, but also markings, obstacles, and other road users. And there are also simple systems that are capable of recognizing only one type of object, for example, a traffic light signal.

Possible approaches to solving the problem of implementing a traffic sign recognition system are analyzed. It is proposed to develop a system using artificial neural networks. A multilayer perceptron was chosen as such a network.

In the future, it is planned to implement a software-developed system.

References:

1. Lyashenko V., & et al. (2023). Automated Monitoring and Visualization System in Production. Int. Res. J. Multidiscip. Technovation, 5(6), 09-18.

2. I. Nevliudov, & et al. (2022). Object Recognition for a Humanoid Robot Based on a Microcontroller. In IEEE XVIII International Conference on the Perspective Technologies and Methods in MEMS Design (MEMSTECH), IEEE, 61-64.

3. Maksymova, S., Velet, A., (2022)Development of an Automated System of Terminal Access to Production Equipment Using Computer Vision. In Manufacturing & Mechatronic Systems 2022, 22-23.

4. Maksymova, S., Nikitin, V., Software for Monitoring Traffic Signs In Manufacturing & Mechatronic Systems 2022, 27-30.

5. Lyashenko, V., Kobylin, O., & Ahmad, M. A. (2014). General methodology for implementation of image normalization procedure using its wavelet transform. International Journal of Science and Research (IJSR), 3(11), 2870-2877.

6. Baker, J. H., Laariedh, F., Ahmad, M. A., Lyashenko, V., Sotnik, S., & Mustafa, S. K. (2021). Some interesting features of semantic model in Robotic Science. SSRG International Journal of Engineering Trends and Technology, 69(7), 38-44.

7. Abu-Jassar, A. T., Al-Sharo, Y. M., Lyashenko, V., & Sotnik, S. (2021). Some Features of Classifiers Implementation for Object Recognition in Specialized Computer systems. TEM Journal: Technology, Education, Management, Informatics, 10(4), 1645-1654.

ISSN: 2582-4686 SJIF 2021-3.261,SJIF 2022-2.889
ResearchBib Impact Factor: 8.848 / 2023

8. Al-Sharo, Y. M., Abu-Jassar, A. T., Sotnik, S., & Lyashenko, V. (2021). Neural Networks As A Tool For Pattern Recognition of Fasteners. International Journal of Engineering Trends and Technology, 69(10), 151-160.

9. Ahmad, M. A., Sinelnikova, T., Lyashenko, V., & Mustafa, S. K. (2020). Features of the construction and control of the navigation system of a mobile robot. International Journal of Emerging Trends in Engineering Research, 8(4), 1445-1449.

10. Abu-Jassar, A. T., & et al.. (2023). Access Control to Robotic Systems Based on Biometric: The Generalized Model and its Practical Implementation. International Journal of Intelligent Engineering & Systems, 16(5), 313-328.

11. Al-Sharo, Y. M., Abu-Jassar, A. T., Sotnik, S., & Lyashenko, V. (2023). Generalized Procedure for Determining the Collision-Free Trajectory for a Robotic Arm. Tikrit Journal of Engineering Sciences, 30(2), 142-151.

12. Matarneh, R., & et al.. (2019). Development of an Information Model for Industrial Robots Actuators. IOSR Journal of Mechanical and Civil Engineering (IOSR-JMCE), 16(1), 61-67.

13. Lyashenko, V., & Sotnik, S. (2020). Analysis of Basic Principles for Sensor System Design Process Mobile Robots. Journal La Multiapp, 1(4), 1-6.

14. Minenko, E., Pyna, O., Belenchuk, O., & Bondar, T. (2021). Analysis and results of measures to ensure road safety in Ukraine for the period.

15. Holovkin, B. M. (2022). Assessment of Road Traffic Accidents and the Severity of Its Consequences in Ukraine. Probs. Legality,156, 52.

16. Karbovska, L., & et al. (2019). State and trends of the road goods transportation field development in Ukraine. Journal of Advanced Research in Law and Economics, 10(4 (42)), 1022-1031.

17. Elfring, J., Dani, S., Shakeri, S., Mesri, H., & van den Brand, J. W. (2020, September). Vehicle localization using a traffic sign map. In 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), IEEE, 1-6.

18. Kumar, R., & et al. (2023). Real-time data sharing, path planning and route optimization in urban traffic management. Multimedia Tools and Applications, 1-19.

19. Aladin, D. V., & et al. (2019). Logic-based artificial intelligence in systems for monitoring the enforcing traffic regulations. In IOP Conference Series: Materials Science and Engineering. IOP Publishing, 534(1)

20. Campbell, A., & et al. (2019). Detecting and mapping traffic signs from Google Street View images using deep learning and GIS. Computers, Environment and Urban Systems, 77, 101350.

21. Zhang, H., & et al. (2019).. Cityflow: A multi-agent reinforcement learning environment for large scale city traffic scenario. In The world wide web conference, 3620-3624.

22. Lee, W. H., & Chiu, C. Y. (2020). Design and implementation of a smart traffic signal control system for smart city applications. Sensors, 20(2), 508.

23. Tai, S. K., & et al. (2020). Deep learning for traffic sign recognition based on spatial pyramid pooling with scale analysis. Applied Sciences, 10(19), 6997.

24. Wali, S. B., & et al. (2019). Vision-based traffic sign detection and recognition systems: Current trends and challenges. Sensors, 19(9), 2093.