NURE
Kharkiv National University
of Radioelectronics

M&MS 2022, 21-22 October, Kharkiv, UKRAINE

VI International Conference

MANUFACTURING & MECHATRONIC SYSTEMS

УДК: 005:004.896:62-65:338.3

Виробництво & Мехатронні Системи 2022: матеріали VI-ої Міжнародної конференції, Харків, 21-22 жовтня 2022 р.: тези доповідей / [редкол. І.Ш. Невлюдов (відповідальний редактор)].-Харків: [електронний друк], 2022. – 136 с

У збірник включені тези доповідей, які присвячені сучасним тенденціям розвитку технологій та засобів виробництва та мехатронних систем, передовому досвіду та впровадженню їх в галузях систем промислової автоматизації та керування виробництвом; системній інженерії; CAD/CAM/CAE системах; мехатроніці (електро-механічних системах, електронних інструментах систем керування, механічних CAD системах); робототехніці та засобахм інтелектуалізації; MEMS (сучасних матеріалів та технологіях виготовлення MEMS) та компонентах і технологіях автоматизації видобутку, переробки та транспортування нафти та газу.

Manufacturing & Mechatronic Systems 2022: Proceedings of VIst International Conference, Kharkiv, October 21-22, 2022: Thesises of Reports / [Ed. I.Sh. Nevlyudov (chief editor).] .- Kharkiv .: [electronic version], 2022. - 136 p.

The collection includes the thesises of reports on modern trends in the development of technologies and means of production and mechatronic systems, top experience and implementation of them in fields of: industrial automation and production management systems; systems engineering; CAD/CAM/CAE systems; mechatronics (electrical and mechanical systems, electronic control tools, mechanical CAD systems); robotics and intellectual toolls; MEMS (modern materials and manufacturing technologies MEMS) and components and technologies for the automation of oil, gas and oil extraction, processing and transportation.

Kharkiv National University of Radio Electronics

Міністерство освіти і науки України (МОНУ)
Харківський національний університет радіоелектроніки (ХНУРЕ)
Варшавський університет сільського господарства (WULS - SGGW)
Азербайджанський державний університет нафти і промисловості
Національний університет «Львівська політехніка»
Festo Didactic Україна
Jabil Circuit Ukraine Limited
ТОВ «Науково-виробниче підприємство «УКРІНТЕХ»»
Факультет автоматики і комп'ютеризованих технологій (АКТ)
Кафедра комп'ютерно-інтегрованих технологій, автоматизації та мехатроніки (КІТАМ),
Державне підприємство «Науково-дослідний технологічний інститут приладобудування»
Державне підприємство «Південний державний проектно-конструкторський та науково-дослідний інститут авіаційної промисловості»

МАТЕРІАЛИ

VI-ої Міжнародної Конференції
ВИРОБНИЦТВО
&
МЕХАТРОННІ СИСТЕМИ 2022
(21-22 жовтня 2022)
Харків, Україна

M&MS 2022, 21-22 October, Kharkiv, UKRAINE

Software for Monitoring Traffic Signs

Svitlana Maksymova[1], Vladyslav Nikitin[1]

1. CITAM Department, Kharkiv National University of Radio Electronics, UKRAINE, Kharkiv, Nauki Ave. 14., e-mail: igor.nevliudov@nure.ua

Abstract: In this work, an analysis of existing libraries for the creation of traffic sign control was carried out. Analysis of existing devices and applications to ensure the control of road signs, to assist the driver and control them.

Keywords: Tenserflow, Machine Learning, Web Application, Usability, Computer Vision.

I. Introduction

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.

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.

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 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.

So, our research theme is extremely topical.

II. Software 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.

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:

– 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.

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 support Fig.1.1

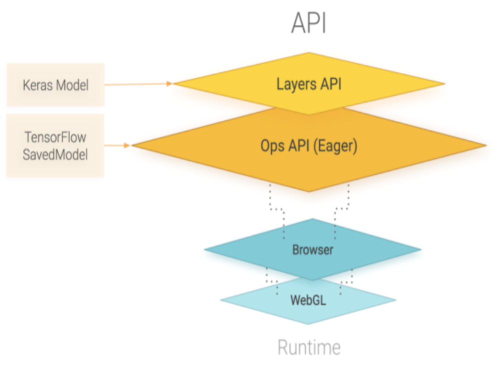

Figure 1.1 – TensorFlow.js Structure

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.

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 Fig. 1.2.

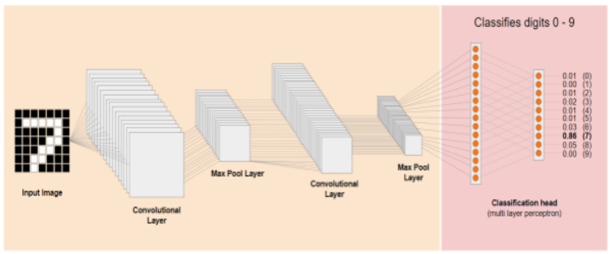

Figure 1.2 - Traditional Convolutional Neural Network Architecture

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 on Fig. 1.3.

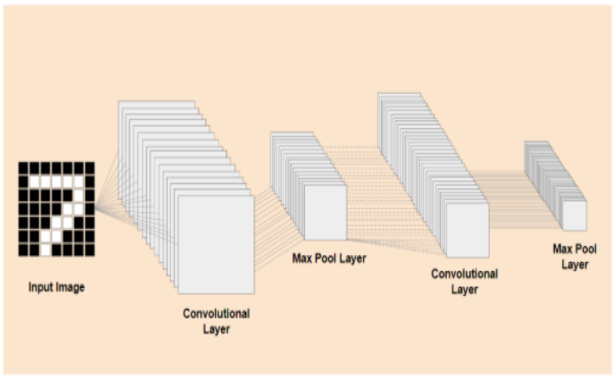

Figure 1.3 - Network with Removed Classification Head

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 Fig. 1.4.

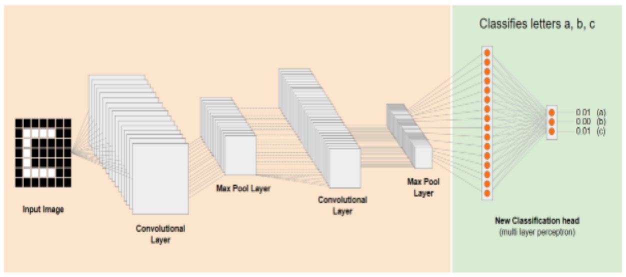

Figure 1.4 – New Clasiification Head

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.

III. Conclusion

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

The developed web application allows you to use it on any device, it reduces the time for its creation and maintenance, it does not take up extra space and does not require additional utilities. With its help, it is possible to control traffic signs, which adds safety when driving road transport.

References

[1]. Akshay Kulkarni, Adarsha Shivananda, " Natural Language Processing Projects. 1st Ed. Akshay Kulkarni, Adarsha Shivananda," pp.38-254, 2022.

[2] Andre Ye "Modern Deep Learning Design and Application Development," in 1st Int. Workshop TLM, Apress. Victoria, Canada, Nov. 2021, pp. 104-328.

[3] Attar, H., & et al.. (2022). Control System Development and Implementation of a CNC Laser Engraver for Environmental Use with Remote Imaging. Computational Intelligence and Neuroscience, 2022, Article ID 9140156, https://doi.org/10.1155/2022/9140156.

[4] Abu-Jassar, A. T., Attar, H., Yevsieiev, V., Amer, A., Demska, N., Luhach, A. K., & Lyashenko, V. (2022). Electronic User Authentication Key for Access to HMI/SCADA via Unsecured Internet Networks. Computational Intelligence and Neuroscience, 2022, Article ID 5866922. https://doi.org/10.1155/2022/5866922.

[5] Attar, H., & et al.. (2022). Zoomorphic Mobile Robot Development for Vertical Movement Based on the Geometrical Family Caterpillar. Computational Intelligence and Neuroscience, 2022, Article ID 3046116, https://doi.org/10.1155/2022/3046116.

[6] Khalid, M. S., Yevsieiev, V., Nevliudov, I. S., Lyashenko, V., & Wahid, R. (2022). HMI Development Automation with GUI Elements for Object-Oriented Programming Languages Implementation. International Journal of Engineering Trends and Technology, 70.1, 139-145.

[7] Nevliudov, I., & et al.. (2021). Development of a cyber design modeling declarative Language for cyber physical production systems, J. Math. Comput. Sci., 11(1), 520-542.

[8] Nevliudov, I., & et al.. (2021). GUI Elements and Windows Form Formalization Parameters and Events Method to Automate the Process of Additive CyberDesign CPPS Development. Advances in Dynamical Systems and Applications, 16(2), 441-455.

[9] Nevliudov, I., & et al.. (2020). Method of Algorithms for Cyber-Physical Production Systems Functioning Synthesis. International Journal of Emerging Trends in Engineering Research, 8(10), 7465-7473.