electronics

MDPI

Article

Deep Learning Framework for Vehicle and Pedestrian Detection in Rural Roads on an Embedded GPU

Luis Barba-Guaman [1,2,*], José Eugenio Naranjo [2] and Anthony Ortiz [1]

[1] Artificial Intelligent Lab, Universidad Técnica Particular de Loja, San Cayetano Alto 1101608, Loja, Ecuador; ajortiz4@utpl.edu.ec

[2] INSIA, Universidad Politénica de Madrid, Carretera de Valencia, km.7, 28031 Madrid, Spain; joseeugenio.naranjo@upm.es

* Correspondence: lrbarba@utpl.edu.ec; Tel.: +593-073701444

Received: 19 February 2020; Accepted: 27 March 2020; Published: 31 March 2020

Abstract: Object detection, one of the most fundamental and challenging problems in computer vision. Nowadays some dedicated embedded systems have emerged as a powerful strategy for deliver high processing capabilities including the NVIDIA Jetson family. The aim of the present work is the recognition of objects in complex rural areas through an embedded system, as well as the verification of accuracy and processing time. For this purpose, a low power embedded Graphics Processing Unit (Jetson Nano) has been selected, which allows multiple neural networks to be run in simultaneous and a computer vision algorithm to be applied for image recognition. As well, the performance of these deep learning neural networks such as ssd-mobilenet v1 and v2, pednet, multiped and ssd-inception v2 has been tested. Moreover, it was found that the accuracy and processing time were in some cases improved when all the models suggested in the research were applied. The pednet network model provides a high performance in pedestrian recognition, however, the sdd-mobilenet v2 and ssd-inception v2 models are better at detecting other objects such as vehicles in complex scenarios.

Keywords: Jetson Nano Nvidia; object detection; neuronal networks; hardware; computer vision

1. Introduction

Computer vision systems have undergone a great development in the field of artificial intelligence in recent years. One of the aspects to consider for this growth has been to not limit itself only to niches as robotics and manufacturing, but also to other areas such as home automation, intelligent detection, medical image analysis, food industry, autonomous driving, among others [1]. Since the beginning, the objective of computer vision systems has been the automatic processing, analysis and interpretation of images [2], to be precise with some classic algorithms including: local descriptor [3], Haar like features [4], SIFT [5], Shape Contexts [6], Histogram of Gradients (HOG) [7] and Local Binary Patterns (LBP) [8]. In 2012, significant advances were made in image processing methods [9], one of which was the use of deep learning techniques. This has led to further research and application, the results of which have shown progress in the majority of computer vision challenges. For example, there are many research activities that are conducted in the area of computer vision, one of the main ones is object detection, which aims to detect and position objects in the images such as traffic signs, vehicles, buildings, and people, to mention some. In contrast to the significant progress in object detection focusing on still images, video object detection has received less attention. Generally, object detection for videos is realized by fusing the results of object detection on the current frame and object tracking from the previous frames.

1.1. Challenges in Object Detection

Liu et al. [9] mentioned that the ideal detection of generic objects is to develop a general purpose algorithm that achieves two important goals, namely high quality/accuracy and high efficiency, as shown in Figure 1. Although the problem of object detection may seem easy, there are several aspects to consider, which make it a real challenge (see Figure 2). In the first place, the variability of objects contained in the same class is one of the biggest difficulties, as well as a change in perspective, the presence of partial occlusions, and changes in illumination, which can create shadows or reflections and cause a significant loss of information. Secondly we have the problem of time efficiency, memory management and storage needed to train these detectors.

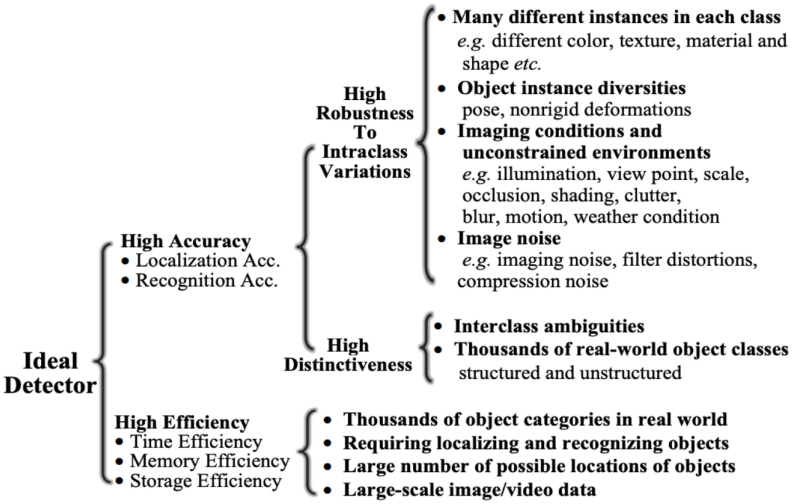


Figure 1. Challenges in generic object detection [9].

Figure 2. Typical problems in object detection in the same class with variation conditions (**a**–**h**). Problem in the variations in what is meant to be a single object class (**i**). Objects appear very similar, but in fact, are from four different object classes (**j**) [9].

1.2. Graphics Processing Units (GPUs)

GPUs consist of many processing cores, and are accelerators that are optimised for performing fast matrix calculations in parallel. These devices are typically very affordable, since their development is motivated by the gaming industry.

Feng et al. [10] mentions that the advances in the development of computer vision algorithms are not only based on deep learning techniques and large data sets, but also relies on advanced parallel computing architectures that enable efficient training of multiple layers of neural networks. Furthermore, a modern GPUs is not only a powerful graphics engine but also a highly parallelized computing processor featuring high throughput and high memory bandwidth for massive parallel algorithms.

Although GPUs were initially intended for high-performance gaming as well as graphics rendering [11], new techniques in the NVIDIA-developed Compute Unifed Device Architecture (CUDA) [12] and CUDA Deep Neural Network Library (cuDNN) [13] have enabled users to adapt them for particular purposes, enhancing system performance. CUDA cores or stream processors are the smallest processing units of NVidia GPUs, and each task can be assigned to one of them.

Zhang et al. [14] states that GPUs are typically built with thousands of cores and operate exceptionally well on the rapid matrix multiplications that are required for neural network training. Therefore, higher memory bandwidth is provided to CPUs and the learning process is dramatically accelerated.

HajiRassouliha et al. [15] provides suitable considerations for the selection of hardware in computer vision and image processing tasks, the devices assessed are Digital Signal Processors (DSPs), Faithful-programmable gate arrays (FPGAs) and Graphics Processing Units (GPUs). In this research, the advantages and disadvantages of development time, tools and utilities, and type of hardware accelerator are discussed, as well as the fact that GPUs are well suited for implementing deep neural network algorithms, because of the similarity between the mathematical basis of neural networks and image manipulation tasks of GPUs. Given their advantages, GPUs have been the most common choice of hardware implementation.

Basulto-Lantova et al. [16] shown the comparison between Jetson TX2 and Jetson Nano performance by using their development kits. Evidence from tests with different image sizes shows that Jetson TX2 is faster than Jetson Nano, which is expected due to the differences in hardware characteristics. However, both computers maintain a relatively low processing time when performing image processing tasks.

1.3. Convolutional Neural Network Algorithms

These advances in multicore architecture have enabled the use of so-called deep convolutional neural network (CNN) architectures for object detection and classification [17,18]. Mauri et al. [2] mentioned that the CNN-based methods have two main categories: the first one is the one-stage methods, this one enables to perform the location and classification of objects in a single network, and the second one is the two-stage methods. The latter contains two separate networks with the purpose that each one of them performs only one task.

In order to mention some examples of these categories, in the first stage we have the Single-Shot Detector (SSD) [19] and You Only Look Once (YOLO) [20,21], these architectures use delimiter boxes for each detected object, additionally the class and the confidence score are demonstrated. An example of the second stage is RCNN (Region-proposal CNN) [22,23] with its enhanced versions, which are based on two independent neural networks, a region-proposal network, and a classification network.

Previous methods can be applied in various fields, one of the most known and interesting within our research is the use of object recognition systems installed in normal and autonomous vehicles [24], which is where many researchers have developed various security solutions and services such as traffic signal recognition [25,26], line recognition on the road [27,28], parking support [29,30], detection of pedestrians or other objects on the urban road [31–35]. However, there is little information or work

developed applying the deep learning techniques in rural areas. Some studies that can be mentioned are presented in [36] such as the detection of the edge of the road where there are no road-markings, and in [37] a segmentation technique for detecting the road in rural areas. Furthermore, [38] mentioned a system for obstacle detection using a LIDAR camera, and in [39] the use of the mobile LiDAR system (MLS) to extract road edges in highly complex surroundings was explained.

1.4. Jetson Nano Modules (NVIDIA JetPack)

Jetson Nano is one of the latest offerings from NVIDIA for AI applications in edge computing. NVIDIA offers a variety of Jetson modules with different capabilities, each module is a computing system packaged as a plug-in unit (System on Module) [40]. NVIDIA JetPack (Figure 3) includes OS images, libraries, APIs, samples, developer tools, and documentation. JetPack SDK provides the following modules:

- TensorRT—SDK for high-performance deep learning inference.
- CUDA—CUDA Toolkit provides a comprehensive development environment for C and C++ developers building GPU-accelerated applications.
- cuDNN—GPU accelerated library for deep neural networks.
- Multimedia API—Video encoding and decoding.
- Computer Vision—Toolkit for computer vision and vision processing.
- Developer Tools—The toolkit includes Nsight Eclipse Edition, debugging and profiling tools including Nsight Compute, and a toolchain for cross-compiling applications.

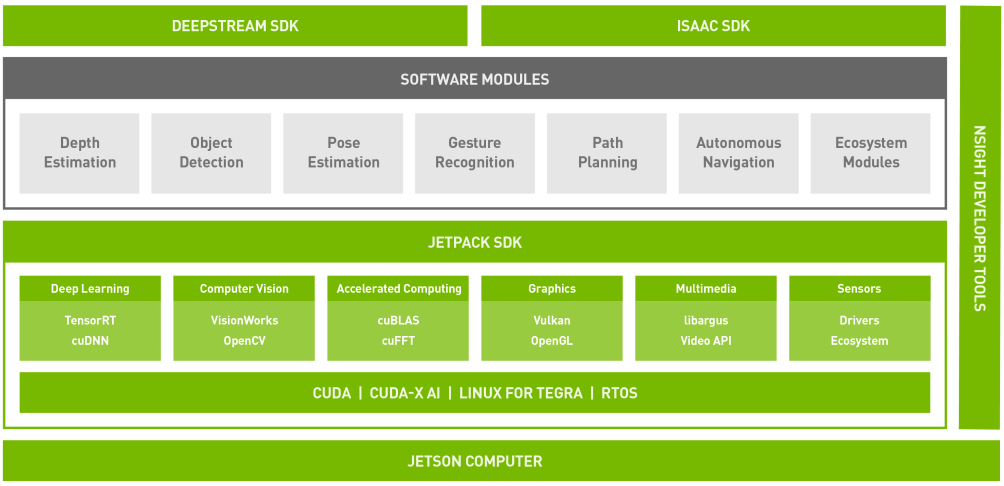


Figure 3. NVIDIA JetPack SDK.

Given this background, the purpose of the research is to check the accuracy and processing time of each model analyzed, however this research has been exclusively in the area of object detection on rural roads and in complex environments, these models have been implemented in the Jetson Nano NVIDIA device, whose results will allow to evaluate if it is suitable to be used in future projects in the autonomous vehicle field.

This article is organized in the following order. Section 2 describes the materials and methods used in the work. The results of the models evaluated as well as the different tests are introduced in Section 3. The discussion and future work is displayed in Section 4 and finally, the conclusion of the research is reached in Section 5.

2. Materials and Methods

2.1. Materials

Algorithms using deep learning technique need a high computational cost in the large scale data set training process. In this research the Jetson Nano NVIDIA [41] device is used (Figure 4).

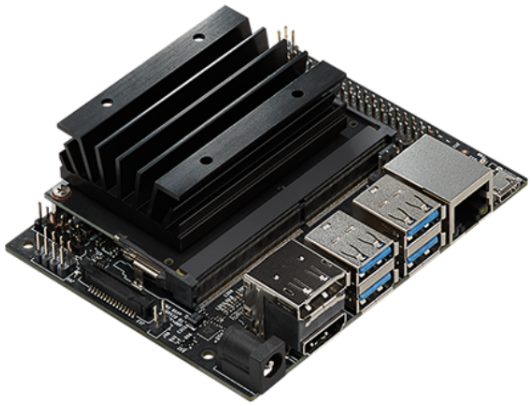


Figure 4. Jetson Nano NVIDIA device.

As a small and powerful machine, this device allows to execute numerous neural network algorithms in parallel. It helps to create applications such as: audio detection, image classification, object detection, segmentation and audio or speech processing. Some of its important features are its small size (69.6 × 45 mm), having a 4-core ARM Cortex-A57 MPCore CPU (capable of providing 472 gigaflops of power), an Nvidia Maxwell GPU with 128 CUDA cores (which can run the CUDA-X AI data processing library), 4 GB of RAM, 16 GB of storage and 4 USB 3.0 ports. In addition, the Jetson Nano NVIDIA is compatible with the most popular artificial intelligence frameworks: TensorFlow, PyTorch, Caffe, Keras, MXNet, etc. In addition, the Isaac Sim Sim package (included in the Isaac SDK by NVIDIA) is intended to provide a training environment for autonomous machines. Full compatibility with these frameworks makes it easier to deploy AI-based inference to Jetson Nano.

NVIDIA's Jetson is a new emerging integrated accelerator hardware, which is widely used with automatic learning algorithms. Such capabilities enable the creation of some applications including autonomous multi-sensor robots, intelligent IoT devices and advanced artificial intelligence systems. Also, the technique known as learning transfer is possible to implement with the pre-training networks that have Jetson Nano, and which use Machine Learning [42]. The key features of this device include its lightweight and low power consumption. Nevertheless, obtaining Jetson's full potential and achieving real-time performance involves an optimization phase for both the hardware and the different algorithms.

In [43], an interesting question is presented: Which edge hardware and what type of network should we bring together in order to maximize the accuracy and speed of deep learning algorithms? The results of this research show that the Jetson Nano NVIDIA and the Coral Dev Board [44] have a high performance in parameters such as processing time and accuracy. The Jetson Nano again achieved good results, although these were relative, despite the fact that edge devices are often equipped with very limited amounts of on-board memory and low cost.

2.2. Metholodogy Process

Figure 5 illustrates the key steps of the proposed object detection method. Three main elements compose the process, these are: image data set, convolutional neural network frameworks and object detection and classification modules.

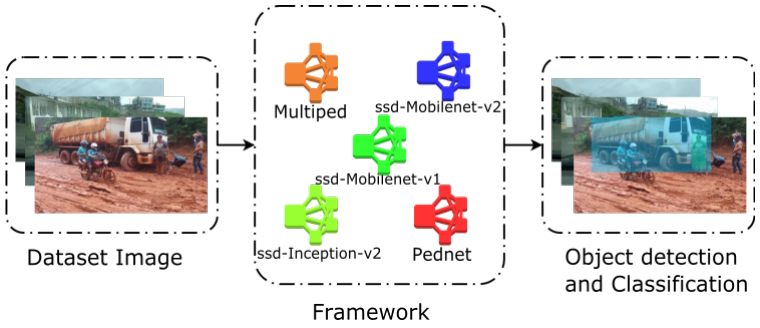


Figure 5. Main steps of the method proposed.

2.2.1. Dataset

Although rural road images are very limited, data augmentation techniques [45] can be used to generate new images for testing purposes. In this research, the number of images of rural roads is 7150, i.e., all images are in RGB with an aspect ratio of 1280 × 720 and 1920 × 1080 pixels. The images show objects such as vehicles and pedestrians in multiple locations in complex situations, including lack of lighting, small objects and different points of view. This is organized into five subsets corresponding to different locations and environments. The defining characteristics of the five datasets are described below

- Dataset 1—Containing 1340 images, the image resolution is 1280 × 720.
- Dataset 2—Containing 2540 images, the image resolution is 1280 × 720.
- Dataset 3—Containing 1180 images, the image resolution is 1920 × 1080.
- Dataset 4—Containing 650 images, the image resolution is 1920 × 1080.
- Dataset 5—Containing 1440 images, the image resolution is 1280 × 720.

2.2.2. Deep Learning Libraries and Models

The Jetson Nano NVIDIA device can run a wide variety of advanced neural networks, including full native versions of popular Machine Learning (ML) frameworks such as TensorFlow, PyTorch, Caffe/Caffe2, Keras, MXNet and others [14] (p. 2234), [41]. The device utilizes the NVIDIA TensorRT accelerator library included with JetPack 4.2. In addition, Jetson Nano is capable of real-time performance in many scenarios and is able to process multiple high-definition video streams.

Some deep neural network architectures have been proposed in the past [9], however, we have focused our implementation on the latest architectures that are rapid and well described in [46,47]. The models referred to in this study are available for download through the NVIDIA Jetson Nano card download tool [48]. Such models are applied in different research areas, such as image recognition, object detection and semantic segmentation.

The models used in the research are detailed below. Figure 6 displays a plot of the deep learning algorithm list used in the embedded system for object detection. A description of the models used is presented as follows:

- SSD-MobileNet v1 and v2: Overall, the Single Shot Multibox Detector (SSD) is employed for reducing the size and complexity of the model. It works by using a multiple feature map along a single network, so in order to increase speed and eliminate proposed regions, the network can use this information to predict very large objects through deeper layers, as well as to predict very small targets by means of shallow layers. This is especially valuable for applications on mobile devices and embedded vision devices [9].
- SSD-Inception v2: This version enhanced accuracy while reducing computational complexity. Chengcheng proposed the detection of objects using Inception Single Shot MultiBox Detector, which improved the SSD algorithm, this means, increasing its classification accuracy without affecting its speed [49]. The purpose in this approach is to reduce the data representation or

bottleneck, i.e., to reduce sometimes the dimensions in the image and this can cause the loss of data, it is also referred to as a "representational bottleneck" and the use of intelligent factorization methods and convolution operations can make it more efficient in terms of computational complexity [11].

- Pednet and multiped: The pednet model (ped-100) is designed specifically to detect pedestrians, while the multiped model (multiped-500) allows to detect pedestrians and luggage [41]. The main advantage of Pednet is its unique design to perform the segmentation from frame to frame, using the previous time information and the next frame information to segment the pedestrian in the current frame [50].

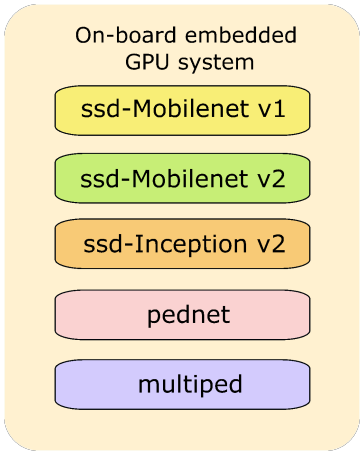


Figure 6. List of deep learning models implemented in the On-board embedded GPU system.

3. Model and Results Assessment

3.1. Metrics of Assessment

Target detection and classification are major challenges in most practical computer vision applications. Hence the necessity to use metrics that allow to validate the capacity that the system has in the detection of objects through significant dimensions.

There are two important and distinct tasks to measure in object detection, these are: (a) to determine if an object is present and (b) where it is located within the image. Target recognition metrics provide a measure for evaluating the performance of the model in an object detection task. Some performance metrics are Precision, Accuracy and Recall. The most common definition is shown in the following equations:

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

where TP stands for the number of true positives, TN means the number of true negatives, FP represents the number of false positives and FN signifies the number of false negatives. Recall and accuracy are the most widely used metrics and are evaluated in most investigations. In the following section, we describe the experiments and results.

3.2. Performance Analysis

On the basis of the rural roads data set, all models are pre-tested, tested and evaluated, their efficiency and development is done via the Tensorflow framework that serves as the backend of cuDNN.

The metrics in vehicle detection are shown in Table 1. The networks used in this process are ssd-mobilenet v1 and v2 and ssd-inception-v2. Performance results are presented below.

Table 1. Vehicles detection result.

Dataset	Models	Rec (%)	Pre (%)	Acc (%)
1	ssd-mobilenet-v1	13.73	93.33	34.56
	ssd-mobilenet-v2	43.56	97.78	57.04
	ssd-Inception-v2	41.75	100	55.88
2	ssd-mobilenet-v1	49.50	96.15	80.59
	ssd-mobilenet-v2	61.90	92.86	66.07
	ssd-Inception-v2	66.07	91.36	84.10
3	ssd-mobilenet-v1	60.56	91.36	84.10
	ssd-mobilenet-v2	64.40	100	67.31
	ssd-Inception-v2	65.99	100	68.69
4	ssd-mobilenet-v1	65.63	93.33	69.88
	ssd-mobilenet-v2	66.67	80.77	65.56
	ssd-Inception-v2	73.21	69.49	63.74
5	ssd-mobilenet-v1	52.38	100	81.31
	ssd-mobilenet-v2	48.15	100	76.27
	ssd-Inception-v2	39.29	95.65	71.31

Table 2 shows the pedestrian detection metrics. Pednet, Multiped, ssd-mobilenet v1 and v2, and ssd-inception-v2 were used in the pedestrian detection process.

Table 1 summarizes the process of vehicle detection in the different data sets. It is possible to observe that the accuracy of the ssd-mobilinet v1, v2 and ssd-inception-v2 models provide good results in detecting the vehicles, and the average values obtained for the accuracy variable are: 70.08%, 66.45%, and 68.74% respectively.

Table 2 provides the metrics in pedestrian detection. It can be seen from this table that the Pednet model offers the maximum accuracy in object detection, namely high precision in almost all scenarios, with an average value of 78.71%. Other average values obtained in the variable accuracy were as follows: Multiped reached a value of 59.03%, ssd-mobilenet-v1 achieved a value of 64.02%, ssd-mobilenet-v2 got an average of 65.72% and finally, ssd-inception-v2 obtained a value of 64.77%. In some cases the problems mentioned in the Figure 2 such as illumination, occlusion, movement, scale do not allow a correct detection.

Nevertheless, the GPU's device cannot be afforded to keep a heavy use of memory in an application, due to its restrictions on both shared and global memory capacity; further understanding of the models discussed is provided in Figures 7 and 8. Certainly, the Pednet model has the highest accuracy in recognizing pedestrians (see Figure 7) in complex situations in all five sets, yet the advantage of this model is that it has only a single class to recognize, an advantage that can be exploited in future application development. Regarding the accuracy in detecting vehicles (see Figure 8) it can be seen that ssd-mobilenet v1 and v2 and ssd-inception-v2 have benefits compared to each other, this is due to the architecture of each network; however, by analyzing the best average through the accuracy in the five data sets it can be stated that ssd-mobilenet-v1 has better performances.

Table 2. Pedestrian detection result.

Dataset	Models	Rec (%)	Pre (%)	Acc (%)
1	pednet	42.99	100	55.47
	multiped	86.58	76.79	72.81
	ssd-mobilenet-v1	16.04	89.47	34.06
	ssd-mobilenet-v2	18.69	76.92	34.97
	ssd-Inception-v2	17.48	75.00	34.53
2	pednet	17.39	80.00	84.38
	multiped	73.33	16.42	62.26
	ssd-mobilenet-v1	6.67	37.50	81.64
	ssd-mobilenet-v2	11.11	71.43	83.53
	ssd-Inception-v2	12.50	66.67	82.42
3	pednet	87.50	100	90.23
	multiped	91.33	84.95	81.93
	ssd-mobilenet-v1	65.22	99.17	71.98
	ssd-mobilenet-v2	64.77	97.66	71.02
	ssd-Inception-v2	66.33	100	72.54
4	pednet	22.22	100	89.39
	multiped	83.33	4.81	22.48
	ssd-mobilenet-v1	11.11	14.29	69.88
	ssd-mobilenet-v2	64.77	97.66	71.02
	ssd-Inception-v2	11.11	25.00	63.74
5	pednet	83.93	74.60	74.09
	multiped	96.34	50.00	55.68
	ssd-mobilenet-v1	69.52	64.04	62.56
	ssd-mobilenet-v2	75.00	70.43	68.06
	ssd-Inception-v2	77.19	73.95	70.62

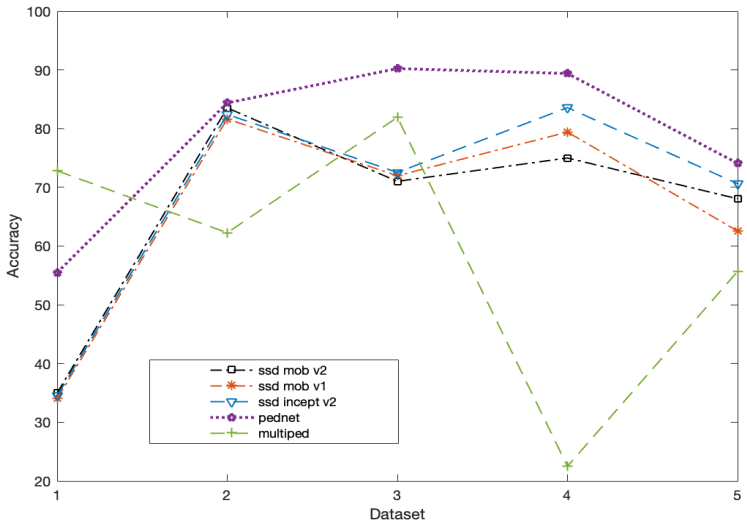


Figure 7. Accuracy pedestrian detection.

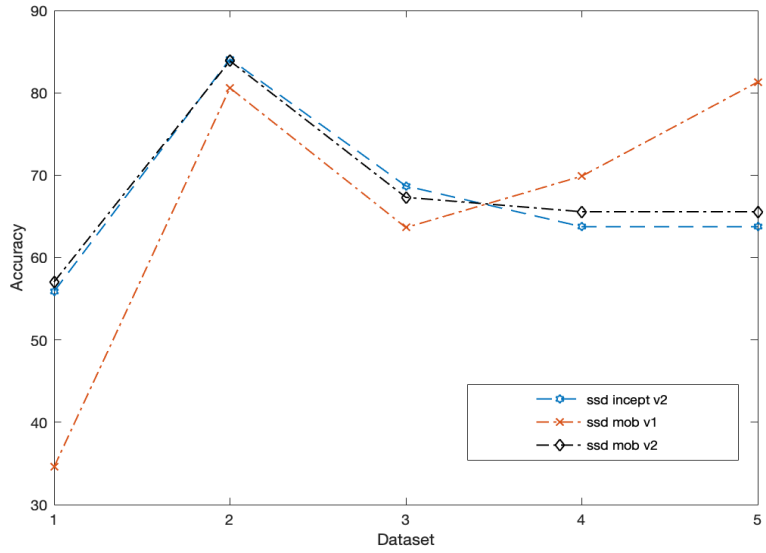
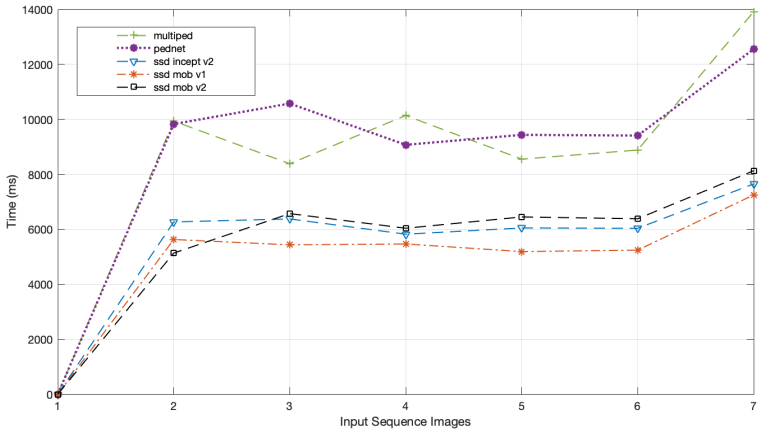


Figure 8. Accuracy vehicle detection.

3.3. Processing Time

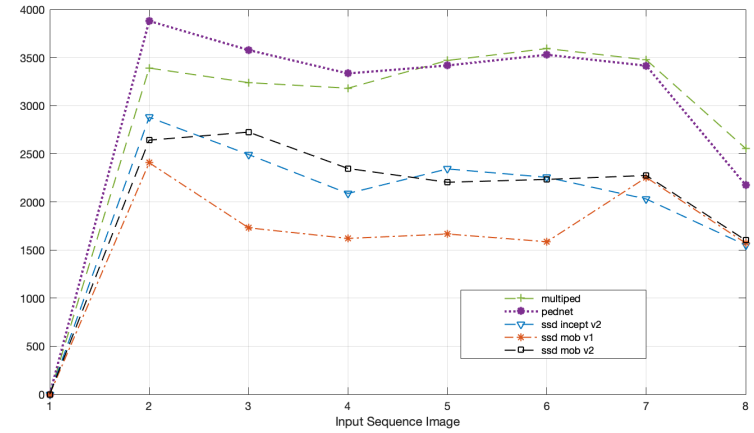
Concerning the processing time, testing consists of examining the model that has the best performance time in a given data set.

Figure 9 shows the values of test that the models that use less time in their performance are ssd-mobilenet-v1, ssd-mobilenet-v2 and ssd inception-v2. The number of tests performed on the data sets was between seven and eight subsets of images. The outcome of this behavior can be explained in part by the differences in network model architectures. For example in Figure 9b, tests were developed on data set 3, it can be seen that the processing times of the Pednet and Multiped models are almost similar, i.e., between 2200 and 3800 ms, while the remaining three models are between 1500 and 3000 ms. Similar analysis occurs for Figure 9a,c.

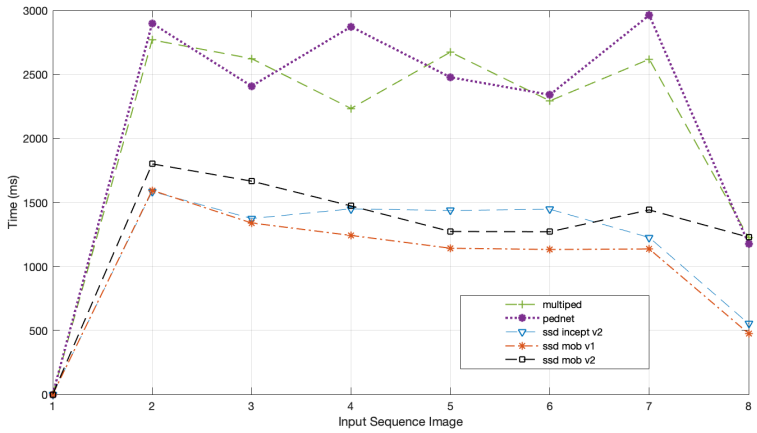


(**a**) Processing time tests of object detection models with dataset 2.

Figure 9. *Cont.*



(**b**) Processing time tests of object detection models with dataset 3.



(**c**) Processing time tests of object detection models with dataset 4.

Figure 9. Evaluation of the processing time in the detection of vehicles and pedestrian in rural roads using different image sets.

3.4. Results

The results of the process of detecting objects through different models of neural networks in the Jetson Nano NVIDIA are presented to support the findings of the research. The main focus has been on models that allow for the detection of pedestrians and vehicles in rural areas. After performing the tests in the previous sections, i.e., the analysis of the variable accuracy and processing time, the results can be seen in Figures 10 and 11. Due to the different resolution of the images that compose the base data set, the normalization of all the images to a fixed dimension was first performed, after this step the spatial size of each image was 512 × 512 px, this is a general requirement of these neural network models.

Considering that Jetson nano is an embedded device of low cost, it enables to import trained models from every deep learning framework into TensorRT, therefore it is possible to optimize neural network models trained in all major frameworks, calibrate for lower precision with high accuracy.

It is necessary to take into account the processing time obtained in the results. Table 3 illustrates that even though the (cuDNN) GPU accelerated library for deep neural networks is used, the response times could be considered high when developing a real time application, thus other models would have to be assessed in order to improve the processing time variable. Although it was not part of the study, it was possible to notice that when processing for several hours a dataset that contains a great amount of images, the Jetson nano device experiences a rise in temperature, so a ventilation system should be installed, according to the device's indications.

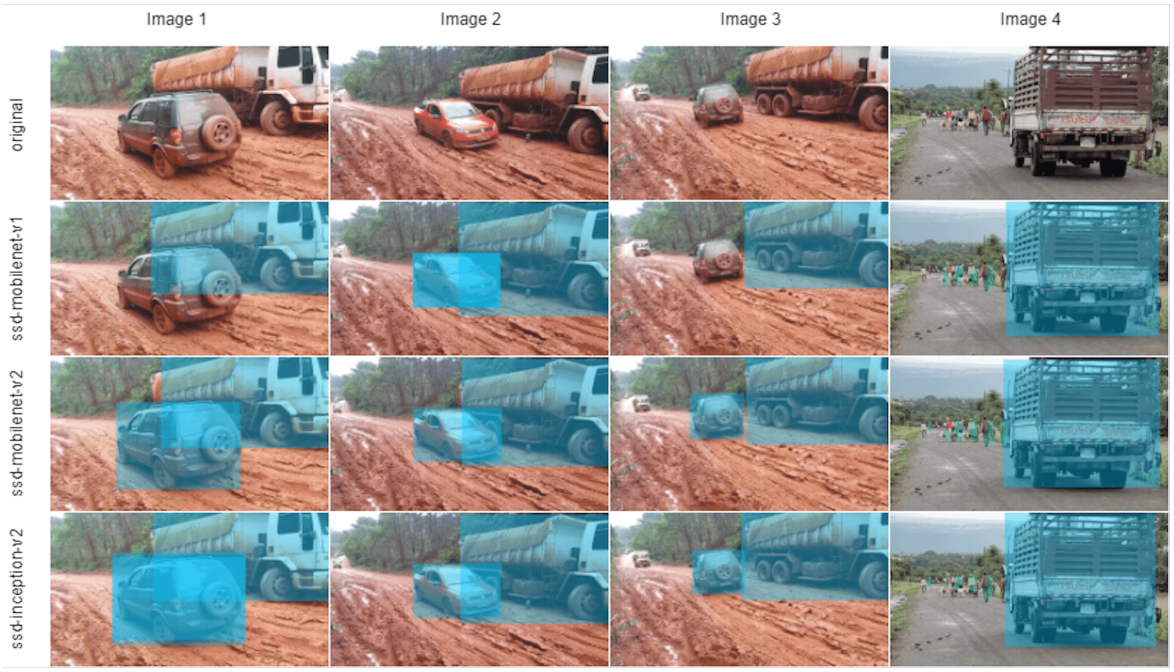


Figure 10. Results for the object detection of the vehicles in each of the three networks on four different test images. Blue boxes represent normal detections.

Figure 10 illustrates the recognition of vehicles and people, it can be appreciated that in the column Image 2 and Image 3 the implemented models cannot detect the vehicle located at a great distance, which is still a challenge in the area of computer vision. Based on the test images it can be seen that the models ssd-mobilenet-v2 and ssd-inception-v2 have better results, although the average of the accuracy variable in Table 4 shows that mobilenet-v1 has better results in general. Figure 11 shows the detection of people, here the five models were tested, and it is remarkable to be able to visualize the accuracy of the Pednet model in all the test images.

It should be emphasized that the number of images from the five datasets is not the same, this is due to the fact that the objects (vehicles and pedestrians) in the actual rural areas are very scarce. The combined database contains 837 individual pedestrians and 681 vehicles. Proposed real-life scenarios permitted to check the accuracy of each model at the time of object detection. Yet, the exactness results versus the processing time of each model, allows us to choose the best model that suits and is more reliable to the solution.

Table 3. Average processing time on embedded board system.

Model	Processing Time (ms)		
	Dataset 2	Dataset 3	Dataset 4
Pednet	**10,146.83**	**3332.42**	**2447.55**
Multiped	**9970.23**	**3271.93**	**2347.68**
ssd-mobilenet-v1	5703.37	1833.16	1151.39
ssd-mobilenet-v2	6452.71	2289.19	1450.08
ssd-Inception-v2	6369.07	2233.57	1295.55

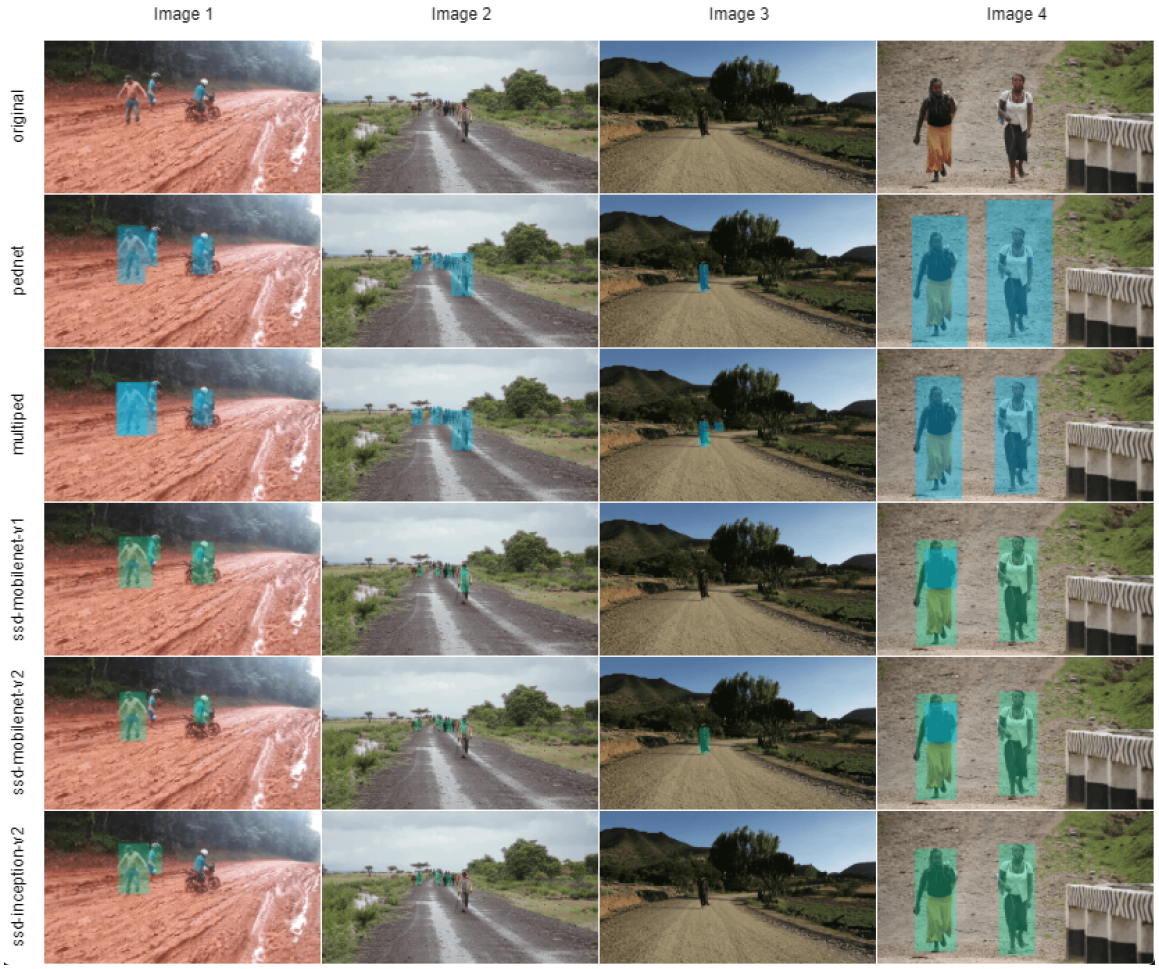


Figure 11. Results for the object detection of the pedestrians in each of the five networks on four different test images. Blue and green boxes represent normal detections. Pednet predicts all pedestrians in the images with the highest accuracy followed by Multiped.

Table 4. Average performance of models on embedded board system.

Model	Cars		Pedestrian	
	Accuracy (%)	Precision (%)	Accuracy (%)	Precision (%)
Pednet	-	-	78.71	90.92
Multiped	-	-	59.03	46.59
ssd-mobilenet-v1	70.08	94.83	64.02	60.89
ssd-mobilenet-v2	66.45	94.28	65.72	82.82
ssd-Inception-v2	68.74	91.30	64.77	68.12

4. Discussion and Future Work

Object recognition is a major and challenging problem in the area of computer vision, and has gotten significant focus from many researchers. Nevertheless, most of the research is applied in the urban area, probably due to the fact that there are a greater number of inconveniences and many solutions have been created.

Our work in a first effort is to be able to use a data set of images of rural roads with different environmental conditions and specifically to validate the ability to detect objects such as people and vehicles and the time that the models are used to perform that process, thereby checking whether they present similar or different behavior in these environments, in parallel using low-cost but high-performance hardware to support the research. Employing different models of deep neural

networks in an embedded device such as the Jetson Nano NVIDIA demonstrated that the models tested provide good accuracy in detecting whether an object (either vehicles or pedestrians) is in the road. To prove this claim, metrics known as accuracy and precision are used.

Table 3 displays that average processing time in the different datasets, for example in dataset 2, the models having higher processing time are Pednet (10,146.83 ms) and multiped (9970.23 ms) while ssd-mobilenet-v1 (5703.37 ms), ssd-mobilenet-v2 (6452.71 ms) and ssd-Inception-v2 (6369.07 ms) models are faster when processing images. Likewise, in dataset 3 the models that need more time to process images are Pednet (3332.42 ms) and multiped (3271.93 ms), while the models ssd-mobilenet-v1 (1833.16 ms), ssd-mobilenet-v2 (2289.19 ms) and ssd-Inception-v2 (2233.57 ms) require less processing time. Finally, in dataset 4 the models Pednet (2447.55 ms) and multiped (2347.68 ms) are the ones that require more time to process the images, while ssd-mobilenet-v1 (1151.39 ms), ssd-mobilenet-v2 (1450.08 ms) and ssd-Inception-v2 (1295.55 ms) are the ones that need less time to process them. Therefore, it can be noticed that in the datasets two, three and four the relation that the Pednet and multiped models have with the processing time, i.e., more time is required in order to detect and locate the objects in the images, while the fastest models are ssd-mobilenet-v1, ssd-mobilenet-v2 and ssd-Inception-v2. This is due to the different architectures of each model and the complexities such as occlusion, scale, illumination, and others that exist in the test environment.

Table 4 summarizes the average performance of the models employed have similar characteristics in each set of images tested, i.e., the best results show, for example, that Pednet can detect pedestrian regions with great accuracy (78.71%), and ssd-mobilenet-v1 (70.08%), ssd-mobilenet-v2 (66.45%) and ssd-inception-v2 (68.74%) are the best models for detecting vehicles in rural road images. Certainly, there are cases in which detection fails and no model can detect the object, this is caused by the problems that have been commented that computer vision exists.

As future work, our mission is to conduct object tracking tests in real conditions on rural roads, i.e., to first consider only the metrics of accuracy, precision, processing time and energy consumption in order to evaluate the best models, this type of implementation can be used for instance in the agricultural sector of the country; and secondly to evaluate the ability and performance of these models in challenging scenarios.

5. Conclusions

This paper has presented the testing of object detection (pedestrians and vehicles) in rural roads through the use of the Jetson Nano NVIDIA embedded device. According to the tests performed, Jetson Nano has a good performance, which is not the same as the most expensive models with better features, but its development kit allows the implementation of complex models to create a variety of applications. Additionally, NVIDIA's hardware platform is constantly being updated with new deep neural network libraries in an efficient and practical way, with the objective of improving the performance of computer vision tasks.

It is evident that the model that detects pedestrians with more accuracy is Pednet, clearly with a greater processing time than the others, meanwhile the other models such as ssd-mobilenet-v1, ssd-mobilinet-v2 and ssd-inception-v2 perform well when detecting vehicles. In addition, these models have a lower processing time, which is a useful advantage when designing applications in embedded systems.

The different modules that can be used through the JetPack permit easily to implement and develop different types of applications. In this case, the TensorRT module with its different pre-trained models made it possible to optimize neural network models trained in all major frameworks, as well as calibrate for lower precision with high accuracy.

Results suggest that Jetson Nano has performance that can be considered as acceptable in object detection and classification tasks, which has been able to be analyzed through processing time variables and accuracy; nevertheless, the processing time is low when the environmental conditions are not complex, however, otherwise it may need a lot of time to perform the processing. Through our

implementation, Jetson Nano's performance in object detection was evaluated. While it is true that we cannot generalise the results of this evaluation, they may be a reference for modern edge devices to offer in object detection applications.

Finally, the purpose of the research was focused exclusively on the detection of objects in embedded systems in order to find the optimal models for the creation of future applications, so these tests only require the use of still images. This work is expected to be useful when it is required to decide which model to use in object detection processing projects.

Author Contributions: L.B.-G. was in charge of the conceptualization of the technical writing of the paper and oversaw the implementation and algorithms test; J.E.N. oversaw the model, review the writing and results of the project; A.O. was in charge of implementing the algorithms and testing. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Artificial Intelligence Laboratory of Universidad Técnica Particular de Loja, Ecuador, and Ecuadorian Corporation for the Development of Research and Academia, CEDIA-Ecuador.

Acknowledgments: We would like to thank the University Institute of Automobile Research (INSIA) from Spain, Artificial Intelligence Laboratory of Universidad Técnica Particular de Loja, Ecuador, and Ecuadorian Corporation for the Development of Research and Academia, CEDIA.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Talukdar, J.; Gupta, S.; Rajpura, P.; Hegde, R. Transfer Learning for Object Detection using State-of-the-Art Deep Neural Networks. In Proceedings of the 2018 5th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 22–23 February 2018; pp. 78–83.
2. Mauri, A.; Khemmar, R.; Decoux, B.; Ragot, N.; Rossi, R.; Trabelsi, R.; Boutteau, R.; Ertaud, J.; Savatier, X. Deep Learning for Real-Time 3D Multi-Object Detection, Localisation, and Tracking: Application to Smart Mobility. *Sensors* **2020**, *20*, 532. [CrossRef] [PubMed]
3. Mikolajczyk, K.; Schmid, C. A Performance Evaluation of Local Descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1615–1630. [CrossRef] [PubMed]
4. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Kauai, HI, USA, 8–14 December 2001; IEEE: Piscataway, NJ, USA, 2001; pp. 511–518.
5. Lowe, D. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
6. Belongie, S.; Malik, J.; Puzicha, J. Shape Matching and Object Recognition Using Shape Contexts Serge. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 509–522. [CrossRef]
7. Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893.
8. Ojala, T.; Pietikainen, M.; Maenpaa, T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *24*, 971–987. [CrossRef]
9. Liu, L.; Ouyang, W.; Wang, X.; Fieguth, P.; Chen, J.; Liu, X.; Pietikäinen, M. Deep learning for generic object detection: A survey. *Int. J. Comput. Vis.* **2020**, *128*, 261–318. [CrossRef]
10. Feng, X.; Jiang, Y.; Yang, X.; Du, M.; Li, X. Computer vision algorithms and hardware implementations: A survey. *Integr. VLSI J.* **2019**, *69*, 309–320. [CrossRef]
11. Rosebrock, A. *Deep Learning for Computer Vision with Python: ImageNet Bundle*; PyImageSearch: New York, NY, USA, 2017.
12. Nickolls, J.; Buck, I.; Garland, M.; Skadron, K. Scalable parallel programming with CUDA. *ACM Queue* **2008**, *6*, 40–53. [CrossRef]
13. Chetlur, S.; Woolley, C.; Vandermersch, P.; Cohen, J.; Tran, J.; Catanzaro, B.; Shelhamer, E. cudnn: Efficient primitives for deep learning. *arXiv* **2014**, arXiv:1410.0759
14. Zhang, C.; Patras, P.; Haddadi, H. Deep learning in mobile and wireless networking: A survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2224–2287. [CrossRef]

15. HajiRassouliha, A.; Taberner, A.; Nash, M.; Nielsen, P. Suitability of recent hardware accelerators (DSPs, FPGAs, and GPUs) for computer vision and image processing algorithms. *Signal Process. Image Commun.* **2018**, *68*, 101–119. [CrossRef]
16. Basulto-Lantsova, A.; Padilla-Medina, J.; Perez-Pinal, F.; Barranco-Gutierrez, A. Performance comparative of OpenCV Template Matching method on Jetson TX2 and Jetson Nano developer kits. In Proceedings of the 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 6–8 January 2020; pp. 0812–0816.
17. Wang, X. Deep learning in object recognition, detection, and segmentation. *Found. Trends Signal Process* **2016**, *8*, 217–382. [CrossRef]
18. Pathak, A.R.; Pandey, M.; Rautaray, S. Application of Deep Learning for Object Detection. *Procedia Comput. Sci.* **2018**, *132*, 1706–1717. [CrossRef]
19. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; Springer: Cham, Switzerland, 2016; pp. 21–37.
20. Shafiee, M.; Chywl, B.; Li, F.; Wong, A. Fast YOLO: A fast you only look once system for real-time embedded object detection in video. *arXiv* **2017**, arXiv:1709.05943.
21. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
22. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014; pp. 580–587.
23. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
24. Nvidia Embedded Systems for Next-Generation Autonomous Machines. NVidia Jetson: The AI Platform for Autonomous Everything. Available online: https://www.nvidia.com/en-gb/autonomous-machines/embedded-systems/ (accessed on 27 February 2019).
25. Yoneda, K.; Suganuma, N.; Aldibaja, M. Simultaneous state recognition for multiple traffic signals on urban road. In Proceedings of the 2016 11th France-Japan 9th Europe-Asia Congress on Mechatronics (MECATRONICS)/17th International Conference on Research and Education in Mechatronics (REM), Compiegne, France, 15–17 June 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 135–140.
26. Zhou, Y.; Chen, Z.; Huang, X. A system-on-chip FPGA design for real-time traffic signal recognition system. In Proceedings of the 2016 IEEE International Symposium on Circuits and Systems (ISCAS), Montreal, QC, Canada, 22–25 May 2016; pp. 1778–1781.
27. Lee, S.; Kim, J.; Shin Yoon, J.; Shin, S.; Bailo, O.; Kim, N.; Lee T.; Hong, H.; Han, S.; Kweon, I. Vpgnet: Vanishing point guided network for lane and road marking detection and recognition. In Proceedings of the IEEE International Conference on Computer Vision, Venecia, Italy, 22–29 October 2017; pp. 1947–1955.
28. Ozgunalp, U.; Fan, R.; Ai, X.; Dahnoun, N. Multiple lane detection algorithm based on novel dense vanishing point estimation. *IEEE Trans. Intell. Transp. Syst.* **2016**, *18*, 621–632. [CrossRef]
29. Krasner, G.; Katz, E. Automatic parking identification and vehicle guidance with road awareness. In Proceedings of the 2016 IEEE International Conference on the Science of Electrical Engineering (ICSEE), Eilat, Israel, 16–18 November 2016; pp. 1–5.
30. Heimberger, M.; Horgan, J.; Hughes, C.; McDonald, J.; Yogamani, S. Computer vision in automated parking systems: Design, implementation and challenges. *Image Vis. Comput.* **2017**, *68*, 88–101. [CrossRef]
31. Sistu, G.; Leang, I.; Yogamani, S. Real-time joint object detection and semantic segmentation network for automated driving. *arXiv* **2019**, arXiv:1901.03912.
32. Chen, X.; Ma, H.; Wan, J.; Li, B.; Xia, T. Multi-view 3d object detection network for autonomous driving. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1907–1915.
33. Barba Guamán, L. Utilización de métodos de visión artificial para pc como apoyo en la automoción. Master's Thesis, Polytechnic University of Madrid, Madrid, Spain, July 2015.
34. Mazzia, V.; Khaliq, A.; Salvetti, F.; Chiaberge, M. Real-Time Apple Detection System Using Embedded Systems With Hardware Accelerators: An Edge AI Application. *IEEE Access* **2020**, *8*, 9102–9114. [CrossRef]

35. Son, S.; Baek, Y. Design and implementation of real-time vehicular camera for driver assistance and traffic congestion estimation. *Sensors* **2015**, *15*, 20204–20231. [CrossRef]
36. Seo, Y.; Rajkumar, R. Detection and tracking of boundary of unmarked roads. In Proceedings of the 17th International Conference on Information Fusion (FUSION), Salamanca, Spain, 7–10 July 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1–6.
37. Yadav, S.; Patra, S.; Arora, C.; Banerjee, S. Deep CNN with color lines model for unmarked road segmentation. In Proceedings of the 2017 IEEE International Conference on Image Processing (ICIP), Beijing, China, 17–20 September 2017; pp. 585–589.
38. Carlino A.; Altomare, L.; Darin, M.; Visintainer, F.; Marchetto, A. Automotive LIDAR-Based Strategies for Obstacle Detection Application in Rural and Secondary Roads. In *Advanced Microsystems for Automotive Applications*; Lecture Notes in Mobility; Springer: Cham, Switzerland, 2015.
39. Yadav, M.; Singh, A.K. Rural Road Surface Extraction Using Mobile LiDAR Point Cloud Data. *Indian Soc. Remote Sens.* **2018**, *46*, 531–538. [CrossRef]
40. NVIDIA JetPack. Available online: https://developer.nvidia.com/embedded/jetpack (accessed on 15 February 2019).
41. Jetson Nano Nvidia. Available online: https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/ (accessed on 1 December 2019).
42. Jetson Nano Brings AI Computing to Everyone. Available online: https://devblogs.nvidia.com/jetson-nano-ai-computing/ (accessed on 1 February 2020).
43. Tryolabs. Machine Learning Edge Devices: Benchmark Report. Available online: https://tryolabs.com/blog/machine-learning-on-edge-devices-benchmark-report/ (accessed on 10 January 2020).
44. Google Coral AI. Aviable online: https://coral.ai/products/dev-board/ (accessed on 12 January 2020).
45. Fawzi, A.; Samulowitz, H.; Turaga, D.; Frossard, P. Adaptive data augmentation for image classification. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3688–3692.
46. Chen, T.; Moreau, T.; Jiang, Z.; Zheng, L.; Yan, E.; Shen, H.; Cowan, M.; Wang, L.; Davis, U.; Hu, Y.; et al. TVM: An automated end-to-end optimizing compiler for deep learning. In Proceedings of the 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18), Carlsbad, CA, USA, 8–10 Octuber 2018; pp. 578–594.
47. Ordóñez, F.J.; Roggen, D. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors* **2016**, *16*, 115. [CrossRef] [PubMed]
48. Nvidia Deploying Deep Learning. Available online: https://github.com/dusty-nv/jetson-inference (accessed on 15 December 2019).
49. Ning, C.; Zhou, H.; Song, Y.; Tang, J. Inception Single Shot MultiBox Detector for object detection. In Proceedings of the 2017 IEEE International Conference on Multimedia and Expo Workshops (ICMEW), HongKong, China, 10–14 July 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 740–755.
50. Ullah, M.; Mohammed, A.; Alaya Cheikh, F. PedNet: A Spatio-Temporal Deep Convolutional Neural Network for Pedestrian Segmentation. *Imaging* **2018**, *4*, 107. [CrossRef]