An Evaluation of Neural Network Efficacies for Image Recognition on Edge-AI Computer Vision Platform

Jie Zhao, Meng Su

***Abstract*—**Image recognition enables machine-like robotics to understand a scene and plays an important role in computer vision applications. Computer vision platforms as physical infrastructure, supporting Neural Networks for image recognition, are deterministic to leverage the performance of different Neural Networks. In this paper, three different computer vision platforms – edge AI (Jetson Nano, with 4GB), a standalone laptop (with RTX 3000s, using CUDA), and a web-based device (Google Colab, using GPU) are investigated. In the case study, four prominent neural network architectures (including AlexNet, VGG16, GoogleNet, and ResNet (34/50)), are deployed. By using public ImageNets (Cifar-10), our findings provide a nuanced perspective on optimizing image recognition tasks across Edge-AI platforms, offering guidance on selecting appropriate neural network structures to maximize performance under hardware constraints.

***Keywords*—**AlexNet, VGG, GoogleNet, ResNet, ImageNet, Cifar-10, Edge AI, Jetson Nano, CUDA, GPU.

I. Introduction

A. Overview of Image Recognition

IMAGE recognition stands as one of the most dynamic and integral components of computer vision, a field that imparts machines with the ability to interpret and understand visual information from the world. As a cornerstone of computer vision, image recognition involves the identification and analysis of objects, features, and patterns within images to emulate human vision using digital systems. The pursuit of image recognition began as an endeavor to mimic human visual perception, a complex process where the brain interprets visual stimuli conveyed by the eyes. Early efforts in the field involved simple pattern recognition, which evolved with the advent of machine learning algorithms, paving the way for more advanced image analysis. Advancements in machine learning, particularly deep learning, have propelled image recognition forward. Neural Networks, throughout Artificial Neural Networks (ANNs), especially Convolutional Neural Networks (CNNs), have become standard tools for tackling image recognition tasks due to their architecture, which mirrors the hierarchical pattern recognition of the human visual cortex. With the unprecedented development of AI technologies, computer vision applications using image recognition will trigger the revolution in remote sensing and industry automation [1]-[5].

Jie Zhao, Meng Su are with School of Engineering, Penn State University, The Behrend College, USA (e-mail: jkz5273@psu.edu, mus11@psu.edu).

B. Hardware Platforms in Computer Vision

Computer vision applications are widely recognized for their transformative role in various technological advancements, such as self-driving cars, facial recognition, and diagnostic systems. These innovations are deeply dependent on the underlying computer vision hardware support since they necessitate several key components. Firstly, image sensors, including technologies like lidar, radar, and cameras, are essential for acquiring information and reading data. Secondly, controlling systems are imperative for facilitating communication with peripheral parts or ports. For instance, external devices such as servos are often utilized to maneuver a mounted camera to track mobile objects. Lastly, the incorporation of flexible machine-learning libraries is fundamental for the training of learning algorithms, enabling them to interpret and learn from visual data [6]-[9].

In the realm of computer vision infrastructure, edge AI represents a cutting-edge innovation that has been extensively incorporated into numerous computer vision projects. The widespread adoption of edge AI is attributed to its significant benefits. It is portable and versatile, allowing for easy integration with other devices. Moreover, it offers an on-device vision system that operates independently from cluster computing, eliminating reliance on external computing clusters. This autonomy enhances the efficiency of data processing, enabling faster responses compared to systems that depend on cluster computing [7]-[9].

Therefore, it is imperative to recognize that computer vision platforms, serving as hardware that supports Neural Networks for image recognition, are as vital as the neural network technologies themselves. Consequently, there is a pressing need for these platforms to be more coherently and extensively addressed within the research community, given their crucial role in advancing computer vision applications.

C. Studying Both Neural Networks and Edge-AI

On the other hand, as an important factor of image recognition, constructing Neural Networks is critical for training smart vision machines. Therefore, Neural Networks and edge AI need to be considered together to develop smart vision systems.

This paper tends to answer the following two questions:

1) Are the cross-computer vision platforms' Neural Networks credible for computer vision applications?

2) How do different computer vision platforms influence image recognition?

The structure of the paper is systematically outlined in the following chapters: Section II delves into the various constructions of Neural Networks and ImageNets. Section III presents case studies of image recognition across different computer vision systems. Methodologies are comprehensively detailed in Section IV. Key observations derived from experiments are discussed in Section V. The paper culminates with Section VI, which provides the conclusion.

II. Background and Related Work

A. Architectures of Neural Networks

As LeCun et al. had advanced lenet5 in the late 1990s [10], [11], which excelled in handwritten recognition, the development of Neural Networks has been turned into brand-new pages. The new technologies in Neural Networks focus on the construction of architectures with varying depths and numbers of hidden layers to act as feature maps. In particular, in 2012, AlexNet was proposed with five convolutional hidden layers and three fully connected layers [12]. In 2015, VGG16 was advanced as a representative of successfully studying the depth Neural Networks because VGG16, as self-explanatory, has 16 convolution hidden layers [13]. The significance of VGG16 is that it overcomes the gradient vanish issue, which is the side effect brought by the requirement of pursuing almost no difference between the prediction and targets by constructing very deep Networks. Afterward, research on the depth of Neural Networks has become unprecedented. In the same year, GoogleNet was introduced as an effective tool for image recognition [14]. The evolution narrative concludes with a mention of Resnet [15], which, in its 34- or 50-layer configurations, demonstrated superior efficiency in operation in 2016. Each of these milestones reflects the rapid and substantial progress in the field of deep learning and computer vision.

B. ImageNet and Cifar-10

To leverage different Neural Networks for image recognition, it is necessary to collect image data sets. With the growth of Neural Network developments, massive public image data sets or image databases are accessible [16]. Typically, there are MNIST as handwritten images [17]. There are also large-scale and comprehensive image databases that enable to meet the requirements for different applications in image recognition. The Kaggle as ImageNet is a striking example, where developers can find different human being's faces from different races and apply for facial recognition [18].

In principle, ImageNet is to categorize the images regarding topic-specific class labels, such as traffic, animals, vegetables, and fruits. Therefore, the ImageNet provides both images and class labels in the format of the image's name [16].

Cifar-10 is a public and charge-free ImageNet, with 50K images of training sets and 10K as testing sets. As self-explanatory, Cifar-10 has ten labors, which are: 1). airplane, 2). automobile, 3). bird, 4). cat, 5). deer, 6). dog, 7). frog, 8). horse, 9). ship, 10). Truck. In Cifar-10, the training data sets have 5K sizes of images under each class and the testing sets have 1K size under each class [19].

C. Image Recognition Merits

In assessing the efficacy of various Neural Networks, it is essential to apply a standardized set of criteria. The Image Large Scale Visual Recognition Challenge (ILSVRC) suggests two primary benchmarks for this purpose. The first benchmark is the accuracy rate, which measures the precision of image recognition when utilizing Neural Networks in conjunction with individual computer vision platforms. The second benchmark is the efficiency of the computer vision platforms in executing Neural Networks specifically for the task of image recognition. These benchmarks are critical for understanding the performance and utility of Neural Networks in practical applications [20]-[22].

III. Case Studies

A. Three Different Computer Vision Platforms

To find an optimized edge-AI device for image recognition, Jetson Nano is considered. Because Jetson Nano outperforms for the flexibility of AI library usage. In structure, Jetson Nano is constituted of two parts: hardware and software. The hardware contains a Jetson Nano 4 GB motherboard packaged in an acrylic case, Wi-Fi (with ethernet chip card), and the cameras (raspberry Pi cam). The software is Jetpack SDK_4.6 and ML-related libraries [23].

The second computer vision system is the NVIDIA GeForce RTX 3000 laptop. To use GPU resources, it is necessary to set up CUDA. The configuration of CUDA takes steps, for the installation in the laptop. Step One is to install the CUDA toolkit. As the GeForce RTX 3000 is the pattern in the laptop, the installation version of the CUDA toolkit is selected as 11.5. Step Two is to download the CuDNN library with version 8.3, due to the GeForce RTX 3000 laptop. Step Three is to add related files from the CuDNN to the installed CUDA toolkit V11.5. Step Four is to pip-install PyTorch in the GeForce RTX 3000 laptop.

The third computer vision system is Google Colab, which is web-based. Google Colab has characteristics, that are listed below [24]:

- Executing the code on Google Colab does not require any Python configuration on the local machine.
- Google Colab has its own built-in and ML-related libraries, which is convenient for ML-related applications.
- It has cloud-based running time for ML-related models and gets GPU support.

As a free resource, Google Colab was used for the experiment containing one GPU.

B. Experiment Process

In the experiment, four prominent neural networks—AlexNet, VGG16, GoogLeNet, and ResNet (34/50), have been implemented and evaluated their performance for image recognition. Meanwhile, three different computer vision platforms - the Jetson Nano, a standalone laptop with RTX 3000s using CUDA, and Google Colab with GPU support, are alternatively used for image recognition. This research aims to understand how these Neural Networks' accuracy and time efficiency for image recognition tasks are influenced by the hardware capabilities of the platforms on which they are implemented.

On the other hand, for the employment of Cifar-10, training sets with the size of 50,000 are used for image recognition running in NVIDIA GeForth3000 Laptop and Google Colab. Testing sets with the size of 10,000 in Cifar-10 are used for image recognition on the Jetson Nano platform.

IV. Methodology

The experiments are to focus on the performance evaluation of different four Neural Networks (AlexNet, VGG, GoogleNet, and Resnet). Parameterizations are manipulated to testify to real performance in a neural network. The structure of Neural Networks, size of testing sets in Cifar-10, and confidence level are considered in the experiments [24]-[26]. The second manipulation in the experiment is to obtain the prediction results regarding each class label. Typically, the fed data in the machine learning model are of two types: X (images) and Y (Class labels). In the Jetson Nano experiments, the overall data are one type, which is images. As images in Cifar-10 are categorized and saved in each class label's folder, reading the same folder's files means getting the images under the same class label. In this way, the class labels can be obtained by reading the folder's name [25]-[27].

To specify the data process in experiments, the pseudo codes are shown in Table I.

TABLE I
PSEUDOCODE FOR IMAGE RECOGNITION ON JETSON NANO

```
Algorithm: Image Recognition on Jetson Nano
Input: File path of the image, class label, size of testing set
Output: Classification result, running time, accuracy of testing set

BEGIN
   IMPORT Jetson PyTorch libraries
   IMPORT time library
   INITIALIZE timestamp
   DEFINE class label and size of testing set

   FOR each image in the testing set DO
      READ the file using the file path
      LOAD the image using the file name
      INITIALIZE the neural network using Jetson libraries
      CLASSIFY the image
      RECORD the time taken for the process
      COMPUTE the accuracy for the testing set
   ENDFOR

   DISPLAY running time
   DISPLAY accuracy of the size-specific testing set
END
```

V. Key Observations and Experiment Results

After the experiments are conducted, the results of the comparison between the two CV platforms of the GeForce RTX3060 Laptop and Google Colab are shown in Tables II-IV.

In Table II, except for training time, when training accuracy is compared, the pair-wise data in each network are similar between the GeForce RTX3060 Laptop and Google Colab. For example, when the network of VGG16 is focused if rounded into an integer, both the training accuracy and testing accuracy are the same in the computer vision platforms of Google Colab and GeForce RTX 3000 Laptop. The same is true in the network of AlexNet. Even though the network of Resnet34 has different testing accuracy almost by 1 between the two computer vision platforms of Google Colab and GeForce RTX3000 Laptop, the training accuracy is the same between the two CV platforms.

TABLE II
THE RESULTS UNDER BOTH GOOGLE COLAB AND GEFORCE RTX3000 LAPTOP WITH BATCH SIZE OF 64 AND EPOCH SIZE OF 10

Network	Resnet34		VGG16		Alexnet	
CV Platform	Google Colab	GeFroce RTX3060 Laptop	Google Colab	GeFroce RTX3060 Laptop	Google Colab	GeFroce RTX3060 Laptop
Training time (Minutes)	22.3	17.92	18.1	18.03	180.15	6.23
Training set accuracy (%)	(83.31	83.97)	(90.48	90.48)	(10.02	10.02)
Testing set accuracy (%)	(77.06	78.17)	(82.51	82.35)	(10	10)

TABLE III
THE RESULTS UNDER BOTH GOOGLE COLAB AND GEFORCE RTX3000 LAPTOP WITH BATCH SIZE OF 256 AND EPOCH SIZE OF 10

Network	Resnet34		VGG16		AlexNet	
CV platform	GeoForce RTX3000 Laptop	Google Colab	GeoForce RTX3000 Laptop	Google Colab	GeoForce RTX3000 Laptop	Google Colab
Training time (Minutes)	17.16	20.69	16.36	16.71	5.25	135.66
Training set accuracy (%)	(76.55	79.74)	(84.27	84.16)	(72.94	68.43)
Testing set accuracy (%)	(70.23	73.55)	(78.72	78.94)	(68.94	65.53)

TABLE IV
THE RESULTS UNDER BOTH GOOGLE COLAB AND GEFORCE RTX3000 LAPTOP WITH A BATCH SIZE OF 256 AND EPOCH SIZE OF 50

Network	Resnet34		VGG16		AlexNet	
CV Platforms	GeoForce RTX 3000 Laptop	Google Colab	GeoForce RTX 3000 Laptop	Google Colab	GeoForce RTX 3000 Laptop	Google Colab
Training time (Minutes)	80.44	104.24	94.87	83.56	27.31	663.3
Training set accuracy (%)	(99.29	99.4)	(99.4	99.45)	(94.75	94.52)
Testing set accuracy (%)	(82.21	79.13)	(84.23	84.36)	(76.07	76.18)

In Table III, even though with little difference between pair-wise comparisons of different CV platforms running the same

Neural Network, the increasing size of fed data has similar observations as well as those from Table II.

In Table IV, training accuracy and testing accuracy are high enough to indicate good performances for image recognition. Because the training accuracy in Resnet34 and VGG16 is over 99% in the CV platforms of GeForce RTX3060 Laptop and Google Colab. Even though the shallow net is like AlexNet, the training accuracy is still high at 94% for image recognition. For the testing accuracy, although the difference between the two CV platforms is quite different by 3 in Resnet34, the rest of the testing accuracy is quite good because there are almost the same between pair-wise CV platforms' comparison in each group of Neural Networks. Therefore, the hyperparameter with a batch size of 256 and epoch size of 50 is a very good pattern for image recognition.

The results under Jetson platforms with four different Neural Networks are shown in Tables V-VIII.

In Table V, from left to right, the network follows the ascendent order of running time. From this, GoogleNet is the most efficient because GoogleNet takes the least time for image recognition. However, the size of the testing data set by 1 is not a good choice, because the least class could be recognized regarding a lot of testing accuracy with their values by 0.

TABLE V
FOUR NETWORK RESULTS OF IMAGE RECOGNITION ON JETSON NANO WITH THE CIFAR-10 TESTING SIZE OF 1

Networks	GoogleNet		Resnet50		AlexNet		VGG16	
	Running time (Seconds)	Testing set accuracy (%)	Running time (Seconds)	Testing set accuracy (%)	Running time (Seconds)	Testing set accuracy (%)	Running time (Seconds)	Testing set accuracy (%)
airplane	5.724043	100	6.520989	0	7.77753	100	18.11255	0
automobile	5.427471	100	6.875408	0	7.891306	0	15.53893	0
bird	5.589767	0	6.015244	0	7.815665	100	13.97094	100
cat	5.319031	100	6.099291	100	7.65399	0	14.26445	0
deer	5.323715	100	6.337552	0	7.679433	100	13.19881	0
dog	5.695685	100	5.999331	100	7.833215	0	12.8648	0
frog	5.756029	0	6.012182	0	7.478314	100	13.81173	0
horse	5.556313	0	6.146549	100	7.775621	0	13.80585	0
ship	5.584796	0	6.089871	100	7.384751	0	12.74642	0
truck	5.565146	100	6.095932	0	7.315367	100	11.99069	0

TABLE VI
FOUR NETWORK RESULTS OF IMAGE RECOGNITION ON JETSON NANO WITH THE CIFAR-10 TESTING SIZE OF 5

Networks	GoogleNet		Resnet50		AlexNet		VGG16	
	Running time (Seconds)	Testing set accuracy (%)	Running time (Seconds)	Testing set accuracy (%)	Running time (Seconds)	Testing set accuracy (%)	Running time (Seconds)	Testing set accuracy (%)
airplane	10.89383	60	13.72396	0	20.32813	60	57.77033	0
automobile	10.28322	80	13.08999	20	20.24906	0	65.11058	40
bird	10.15842	60	13.06802	40	20.3465	60	66.35648	20
cat	9.990562	80	13.04256	60	20.52468	60	67.5471	0
deer	10.30233	60	13.46593	20	20.2298	100	62.55218	0
dog	10.07325	80	13.22767	60	20.59694	40	65.97516	0
frog	10.19674	40	13.2374	60	20.19881	60	65.94221	0
horse	10.35573	0	13.11108	60	20.52874	80	64.95192	0
ship	10.17637	60	13.47306	60	20.75798	80	66.49997	0
truck	10.27983	100	13.40179	40	20.19967	60	67.37831	0

TABLE VII
FOUR NETWORK RESULTS OF IMAGE RECOGNITION ON JETSON NANO WITH THE CIFAR-10 TESTING SIZE OF 50

Networks	GoogleNet		Resnet50		AlexNet		VGG16	
	Running time (Seconds)	Testing set accuracy (%)	Running time (Seconds)	Testing set accuracy (%)	Running time (Seconds)	Testing set accuracy (%)	Running time (Seconds)	Testing set accuracy (%)
airplane	61.97039	50	91.97914	48	164.083	46	510.7569	18
automobile	61.88198	60	92.39085	38	164.6572	60	477.3398	20
bird	61.87849	68	92.09842	56	163.8739	72	603.0666	8
cat	62.13872	68	92.24445	42	164.4326	60	617.368	4
deer	61.92081	62	92.25239	46	164.706	70	604.8188	0
dog	62.25101	66	92.18049	50	164.4071	48	557.7404	2
frog	62.09995	30	92.11525	52	165.0242	62	586.9518	4
horse	62.06949	44	92.22779	48	164.6422	62	637.5138	4
ship	61.90225	64	91.87517	54	164.2356	68	602.4838	12
truck	61.92471	52	91.98725	38	164.9788	70	519.6955	8

TABLE VIII
FOUR NETWORK RESULTS OF IMAGE RECOGNITION ON JETSON NANO WITH THE CIFAR-10 TESTING SIZE OF 100

Networks	GoogleNet		Resnet50		AlexNet		VGG16	
	Running time (Seconds)	Testing set accuracy (%)	Running time (Seconds)	Testing set accuracy (%)	Running time (Seconds)	Testing set accuracy (%)	Running time (Seconds)	Testing set accuracy (%)
airplane	127.1792	49	179.3506	49	324.9795	49	1142.905	16
automobile	123.7433	54	179.2401	41	325.1481	59	888.5286	19
bird	121.0072	54	179.1602	57	324.708	68	842.7982	7
cat	120.5396	57	178.9847	39	324.6796	50	847.4688	4
deer	121.3258	62	178.9692	41	324.8438	71	741.342	1
dog	120.5887	66	179.1365	51	323.0592	43	808.4374	4
frog	120.5712	28	179.1445	56	321.5237	65	782.5686	5
horse	121.1093	45	179.2257	42	320.3662	53	761.6431	3
ship	120.9289	56	179.0631	53	320.085	68	1175.652	7
truck	121.3233	56	178.8052	33	319.0226	62	1030.62	11

In Table VI, the performance of VGG16 is not good for image recognition because there are plenty of zeros as the testing accuracy in VGG16. In contrast, image recognition in the rest networks is non-zero. This illustrates that the structure of Neural Networks, either shallow or deep, is not an absolute index to determine the performance of image recognition.

In Table VII, when the size of testing sets increased by 50, GoogleNet outperformed, compared with the rest of the networks. Because GoogleNet takes less time but has relatedly high testing accuracy.

In Table VIII, GoogleNet is superior to the rest networks, for image recognition. Under the condition that testing accuracy in four networks is almost the same, the efficiency running for image recognition becomes critical to deciding the performance of the networks.

To sum up, this study highlights four key observations:

1. Both the GeForce RTX 3060 and Google Colab exhibit comparable training and testing accuracy, diverging mainly in the training time required. This reinforces the credibility of cross-platform Neural Networks for computer vision tasks.
2. Batch size influences image recognition modestly when below 256. Specifically, a batch size of 256 coupled with 50 epochs can yield a training accuracy of nearly 99%. Conversely, a smaller batch size of 64 suggests that batch and epoch sizes collectively affect training outcomes.
3. In terms of quantitative output accuracy, the Jetson Nano falls short of the capabilities offered by the GeForce RTX 3000 and Google Colab due to its limited capacity.
4. GoogleNet and AlexNet strike an optimal balance between efficiency and testing accuracy for image recognition tasks, particularly when utilizing smaller subsets of ImageNet.

VI. CONCLUSION

The study presented offers two notable contributions:

1. It facilitates the comparative evaluation of four Neural Networks for image recognition efficiency and accuracy using a consistent dataset, ImageNet - Cifar10.
2. It examines the impact of various computer vision platforms on the performance of these Neural Networks, using the same dataset and evaluation metrics.

Future research will delve further into image recognition, examining a broader range of data sizes from different computer vision platforms, exploring the architecture of Neural Networks, and adjusting hyperparameters to uncover deeper insights into the interplay among these elements.

REFERENCES

[1] Y. H. Lu et al. “Rebooting Computing and Low-Power Image Recognition Challenge”. In: IEEE/ACM International Conference on Computer-Aided Design. 2015, pp. 927–932.
[2] K. Gauen et al., “Low-power image recognition challenge,” in Proceedings of ASP-DAC, IEEE, 2017, pp. 99–104.
[3] D. Kang, D. Kang, J. Kang, S. Yoo, and S. Ha, “Joint optimization of speed, accuracy, and energy for embedded image recognition systems,” in 2018 Design, Automation Test in Europe Conference Exhibition (DATE), March 2018, pp. 715–720.
[4] Y Shi, H Li, “Beyond cross-view image retrieval: Highly accurate vehicle localization using satellite image”, in Proceedings of the IEEE/CVF 2022
[5] H Yu, Y Luo, M Shu, Y Huo, Z Yang, Y Shi, Z Guo, H Li, X Hu, J Yuan, Z Nie, “Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection”, in Proceedings of the IEEE/CVF, 2022
[6] YJ Li, J Park, M O'Toole, “Modality-agnostic learning for radar-lidar fusion in vehicle detection”, in Proceedings of the IEEE/CVF, 2022
[7] M Bahari, S Saadatnejad, A Rahimi, M Shaverdikondori, AH Shahidzadeh, “Vehicle trajectory prediction works, but not everywhere”, in Proceedings of the IEEE/CVF, 2022
[8] R Xu, X Xia, J Li, H Li, S Zhang, Z Tu, “V2v4real: A real-world large-scale dataset for vehicle-to-vehicle cooperative perception”, in Proceedings of the IEEE/CVF, 2023
[9] H Yu, W Yang, H Ruan, Z Yang, Y Tang, X Gao, X Hao, Y Shi, Y Pan, N Sun, J Song, J Yuan, “V2X-Seq: A Large-Scale Sequential Dataset for Vehicle-Infrastructure Cooperative Perception and Forecasting”, in Proceedings of the IEEE/CVF, 2023
[10] Y LeCun, B Boser, JS Denker, D Henderson, “Backpropagation applied to handwritten zip code recognition”, in Proceedings of the IEEE, 1989
[11] Y LeCun, L Bottou, Y Bengio, “Gradient-based learning applied to document recognition”, in Proceedings of the IEEE, 1998
[12] A Krizhevsky, I Sutskever, “Imagenet classification with deep convolutional neural networks”, in NIPS, 2012
[13] K Simonyan, A Zisserman, “Very deep convnets for large-scale image recognition”, in ICLR, 2014
[14] C Szegedy, W Liu, Y Jia, P Sermanet, “Going deeper with convolutions”, in CVPR, 2015
[15] K He, X Zhang, S Ren, J Sun, “Deep Residual Learning for Image Recognition”, in CVPR, 2016
[16] J Deng, W Dong, R Socher, LJ Li, K Li, “Imagenet: A large-scale hierarchical image database”, in CVPR, 2009
[17] Li Deng, “The mnist database of handwritten digit images for machine learning research”, in IEEE Signal Processing Magazine, Volume: 29 Issue: 6, 2012
[18] Casper Solheim Bojer, Jens Peder Meldgaard, “Kaggle forecasting competitions: An overlooked learning opportunity”, in International

Journal of Forecasting, Volume 37, Issue 2, Pages 587-603, 2021

[19] V Thakkar, S Tewary, “Batch Normalization in Convolutional Neural Networks—A comparative study with CIFAR-10 data”, in EAIT, 2018

[20] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg & Li Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge”, in International Journal of Computer Vision, Volume 115, pages 211–252, 2015

[21] M Everingham, L Van Gool, CKI Williams, J Winn, A Zisserman, “The Pascal Visual Object Classes (VOC) Challenge”, International Journal of Computer Vision, Volume 88, pages 303–338, 2010

[22] Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.

[23] NVIDIA Jetson Nano Developer Kit: https://developer.nvidia.com/embedded/learn/get-started-jetson-nano-devkit

[24] Ekaba Bisong, “Building Machine Learning and Deep Learning Models on Google Cloud Platform”, ISBN- 978-1484244692, Springer, 2019

[25] Sebastian Raschka, Vahid Mirjalili, “Python Machine Learning: Machine Learning and Deep Learning with Python, scikit-learn, and TensorFlow”, ISBN- 978-1789955750, Packt Publishing, 2019

[26] Sebastian Raschka, Yuxi (Hayden) Liu, “Machine Learning with PyTorch and Scikit-Learn: Develop machine learning and deep learning models with Python”, ISBN- 978-1801819312, Packt Publishing, 2022

[27] Sebastian Raschka, David Julian, “Python: Deeper Insights into Machine Learning: Leverage benefits of machine learning techniques using Python”, ISBN- 978-1787128576, Pack Publishing, 2016