JOURNAL
COMPUTER SCIENCES INSTITUTE

JCSI 21 (2021) 303–308
Received: 09 September 2021
Accepted: 19 September 2021

A comparison of conventional and deep learning methods of image classification

Porównanie metod klasycznego i głębokiego uczenia maszynowego w klasyfikacji obrazów

Maryna Dovbnych*, Małgorzata Plechawska–Wójcik

Department of Computer Science, Lublin University of Technology, Nadbystrzycka 36B, 20–618 Lublin, Poland

Abstract
The aim of the research is to compare traditional and deep learning methods in image classification tasks. The conducted research experiment covers the analysis of five different models of neural networks: two models of multi–layer perceptron architecture: MLP with two hidden layers, MLP with three hidden layers; and three models of convolutional architecture: the three VGG blocks model, AlexNet and GoogLeNet. The models were tested on two different datasets: CIFAR–10 and MNIST and have been applied to the task of image classification. They were tested for classification performance, training speed, and the effect of the complexity of the dataset on the training outcome.

Keywords: image classification; machine learning; deep learning; neural networks

Streszczenie
Celem badań jest porównanie metod klasycznego i głębokiego uczenia w zadaniach klasyfikacji obrazów. Przeprowadzony eksperyment badawczy obejmuje analizę pięciu różnych modeli sieci neuronowych: dwóch modeli wielowarstwowej architektury perceptronowej: MLP z dwiema warstwami ukrytymi, MLP z trzema warstwami ukrytymi; oraz trzy modele architektury konwolucyjnej: model z trzema VGG blokami, AlexNet i GoogLeNet. Modele przetrenowano na dwóch różnych zbiorach danych: CIFAR–10 i MNIST i zastosowano w zadaniu klasyfikacji obrazów. Zostały one zbadane pod kątem wydajności klasyfikacji, szybkości trenowania i wpływu złożoności zbioru danych na wynik trenowania.

Słowa kluczowe: klasyfikacja obrazów; uczenie maszynowe; uczenie głębokie; sieci neuronowe

*Corresponding author
Email address: maryna.dovbnych@pollub.edu.pl (M. Dovbnych)

1. Introduction

Nowadays, image classification methods play an important role in a wide variety of areas of life. Image classification is the process of extracting classes of information from a multiband bitmap, in other words, the problem of image classification is receiving an initial image and determine its class (cat, dog, etc.) or a group of probable classes that best characterizing the image. This paper presents a comparison of conventional and deep learning methods of image classification.

Multilayer Perceptron (MLP) is the most popular type of artificial neural networks. It is a class of feedforward artificial neural network. This type of network typically consists of one input layer, several hidden layers and one output layer. Each node in MLP is a neuron with a nonlinear activation function (except of input nodes). Although this type of network ignores the spatial information of the image, a lightweight MLP with 2–3 layers can easily cope with simple data sets like MNIST [1]. The MNIST is a voluminous database of handwritten numeral samples [2]. In the paper [3] MLP network with a single hidden layer was able to reach 43.4% of accuracy. The multilayer perceptron based architecture was once commonly used for computer vision, and is now increasingly being replaced by the Convolutional Neural Network (CNN) [3, 4] and other machine learning methods. For example, the paper [5] compares MLP with other machine learning methods such as decision tree, logistic regression and support vector machine for solving image classification problems.

Artificial networks based on CNN architecture are considered to be universal [6], because they are used for a wide range of tasks, from botany [7, 8, 9, 10] and geography [11] to medical diagnostics [12, 13, 14, 15]. CNN–based models take into account the dimensional information of an image, which gives this type of architecture an advantage over networks with an architecture like MLP for image classification tasks. Another difference between MLP and CNN architectures is that layers in CNN not fully connected like in MLP. Convolutional neural network through the use of a special convolution operation allows to simultaneously reduce the amount of information stored in memory, due to which it copes better with higher–resolution pictures, and to highlight the reference features of the image, such as edges, contours or edges. At the next level of processing, from these edges and faces, you can recognize repeatable fragments of textures, which can then fold into fragments of the image. There are many types of convolutional neural network architectures and their modifications that have been developed to make the trained

model perform better [16, 13]. In the paper [17] proposed methods of the automatic designing CNN architectures using the Genetic image classification algorithm. Not only architectures are being modified, but also ways of solving problems. For example, the paper [18] shows how image segmentation techniques have evolved.

The process of learning a machine itself consists in preparing the appropriate data containing the necessary rules and a description of the object's properties, as well as selecting the optimal parameters for the model which is trained. These factors increase the impact of the selected training data set on training efficiency [19, 10]. The data set is usually divided into several parts: training data, which is used to train the model, validation data, which is used by machine learning engineers during the design phase to tune the hyperparameters of the model, and test data is used to evaluate performance of the already trained model. Sometimes, validation data is used as test data. The number of images in the data set, their size, as well as the number of images in each of the categories by which we will classify them affect the training efficiency. As mentioned above, if the model is be trained successfully, it must be well parametrized and optimized. In the paper [20], the optimization problems faced by a machine learning specialist are described. According to the paper [21] choosing the correct activation function also plays a critical role in model training. The wrong selection of parameters can lead to overfitting or underfitting [22].

Underfitting is a situation when in a parametric family of functions it is not possible to find a function that describes the data well. The most common reason for underfitting is when the complexity of the data structure is higher than the complexity of the model that the researcher came up with. The solution to this problem is to complicate the model and find a better description of the effects that are in the data.

Overfitting is the opposite of underfitting when the model is too complex and universal. The error probability of the trained algorithm on the objects of the test sample turns out to be significantly higher than the average error on the training sample. There are techniques to avoid overfitting the model. For example, increasing the size of the training sample can help, if collecting more data is not possible, then various transformations (rotation, reflection, scaling, etc.) can be performed on an already existing set of images. Techniques such as cross validation, L1/L2 regularization also can help to avoid the problem of overfitting. One of the most effective techniques to prevent the appearance of the overfitting effect is to add dropout layers to the neural network architecture. By using dropout layers model ignore a subset of our network units with a given probability and reduce interdependent learning among units that could lead to overfitting. However, using dropout layers, it will take more epochs for our model to converge.

To predict how the trained model will behave in practice, the performance of the model is evaluated. Different performance metrics are used to evaluate the performance of different algorithms. Metrics such as Confusion Matrix, Accuracy, Precision, Recall, Specificity and F1 Score are commonly used for classification tasks. All of the above metrics use number of true positives, true negatives, false positive and false negative predictions. A true positive is when the model correctly predicted a positive class, and a true negative is when the model correctly predicts a negative class. False positive and false negative, respectively, are cases where the model incorrectly predicted a positive or negative class. Correctly selected metrics are the key to an accurate assessment of model performance.

To carry out this research work, a machine learning framework or library is needed. To solve the problems of image classification in this work, an open source Tensorflow library from Google was chosen. Tensorflow offers many out–of–the–box solutions that make learning model faster and easier. The API of Tensorflow library layer provides a simpler interface to commonly used layers in deep learning models. An example of the classification performance and qualitative analysis using the Tensorflow library can be seen in the paper [23]. A systematic overview of using TensorFlow for image classification can be found in the paper [24].

In this work, it is conducted an experiment that relies on classification performance and qualitative analysis of conventional and deep learning methods of image classification. The thesis of this study is “CNN obtains better performance in the task of image classification than MLP”. Detailed research hypotheses are:

1. CNN based architecture give better accuracy than MLP;
2. models with MLP architecture give lower classification accuracy than CNN–based models when classifying color images;
3. CNN type networks train faster than MLP.

2. Research implementation

The research covered two tests. In the first one, it is checked whether CNN–type architectures give a higher classification accuracy than a multilayer perceptron, and also whether the choice of a black–and–white dataset affects the classification accuracy in the case of using a multilayer perceptron. The second test examines and compares the training speed for the neural network architectures studied in this article.

All tests were carried out on an MSI GL63 8SC laptop with the following specifications:

- CPU: Intel Core i7–8750H;
- CPU Clock Rate: 2.2 GHz / 4.1 GHz;
- GPU: NVIDIA GeForce GTX 1650 GDDR5;
- GPU memory: 4 GB;
- RAM: 16 GB.

Two datasets were chosen for training and evaluating the models: MNIST Database – volume set (60000 train and 10000 test images) of black and white handwritten numbers samples from 0 to 9 (ten classes) size of 28x28 and CIFAR–10 data set [25] consists of color

images in 10 classes size of 32x32. There are 50000 training images and 10000 test images.

The evaluation performance of the model is carried out on the basis of the Accuracy and F1 score metrics, as well as the value of the loss function. Accuracy is way to measure how often the algorithm classifies a data correctly. Accuracy is the number of correctly predicted data points out of all the data points. F1 score is a metric for determining how accurate a test is. It is calculated using the test's precision and recall, with precision equaling the number of true positive results divided by the total number of positive results, including those that were incorrectly identified, and recall equaling the number of true positive results divided by the total number of samples that should have been identified as positive. The F1 score is calculated by taking the harmonic mean of precision and recall.

For the experiment, two models of the MLP type and three convolutional models were chosen: MLP with two hidden layers, MLP with 3 hidden layers, Three VGG blocks model, AlexNet and GoogLeNet.

The MLP is a feedforward neural network having a source neuron input layer, at least one hidden layer (two and three hidden layers in these cases) of computational neurons, and a computational neuron output layer. The input layer receives signals from the environment and redistributes them to all neurons in the hidden layer.

The basic idea behind VGG architectures is to use more layers with smaller filters. There are VGG–16 and VGG–19 versions with 16 and 19 layers respectively. In this experiment, a model with three VGG layers is implemented.

The AlexNet architecture consists of five convolutional layers, between which pooling layers and normalization layers are located, and three fully connected layers complete the neural network.

The GoogLeNet is a deep architecture with 22 layers. The goal was to develop a neural network with the highest computational efficiency. To do this, Google came up with the so–called Inception module — the entire architecture consists of many such modules, following one after another. The idea behind the main Inception module is that it is itself a small local area network. All his work consists in the parallel application of several filters to the original image. The filter data is combined to create an output that goes to the next layer.

2.1. Test 1 implementation

Each model was trained in a loop 20 times to maximum accuracy for MNIST and CIFAR–10 data sets, and the training results (Accuracy, Loss, F1 score) were recorded in a csv format file for further analysis. Chart of accuracy and loss and confusion matrix chart were saved on disk. SGD optimizers and data generators were used for all models. At the beginning of each loop step, a random seeds were set, the training data set was randomly splitted into training (80%) and validation (20%) subsets, a new instance of SGD optimizer, data generator and model were created and compiled. After training, the model was saved to disk, and then its object was deleted, and the session was cleaned.

2.2. Test 2 implementation

All models were trained with the same settings as in section 2.1. 10 times up to 20 epochs on MNIST data set. Accuracy and loss of first, fifth, tenth, fifteenth and twentieth epochs, as well as accuracy, loss and F1 score for test data set were recorded in the csv file at each loop step.

3. Results of first test

During the analysis of the results, the mean value of the accuracy, loss function and F1 score were calculated for each model. Mean loss for each model without division into a data set are presented in the figure 1.

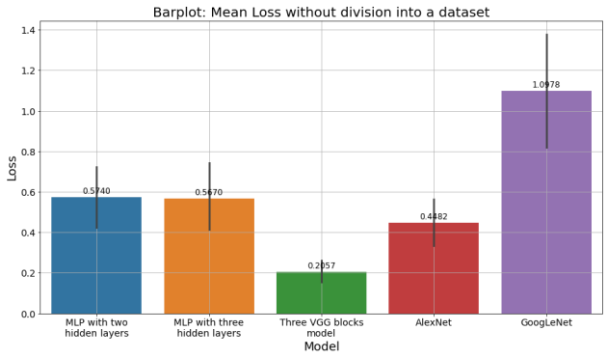

Figure 1: Mean loss for each model without division into a data set

Mean accuracy for each model without division into a data set are presented in the figure 2.

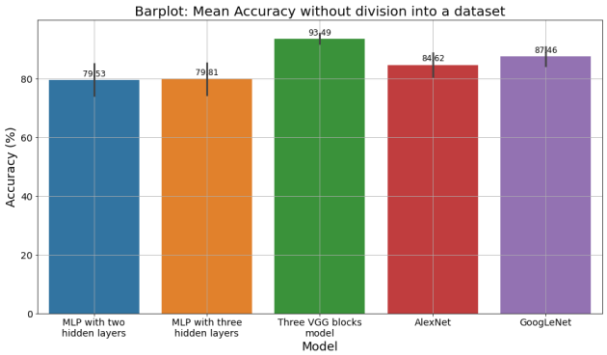

Figure 2: Mean accuracy for each model without division into a data set

Mean F1–score value for each model without division into a data set are presented in the figure 3.

Mean loss for each model on MNIST data set are shown in the figure 4.

Mean accuracy for each model on MNIST data set are presented in the figure 5.

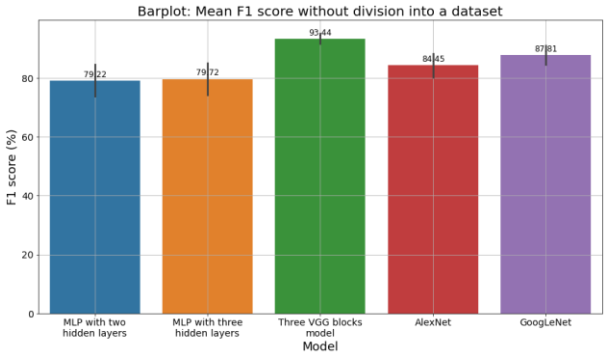

Figure 3: Mean F1–score for each model without division into a data set

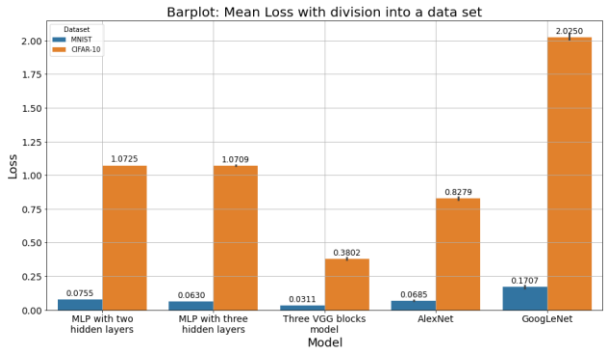

Figure 4: Mean loss for each model with division into a data set

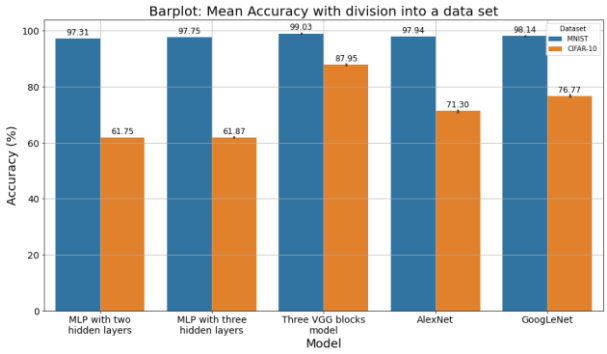

Figure 5: Mean accuracy for each model with division into a data set

Mean F1–score for each model on MNIST data set are shown in the figure 6.

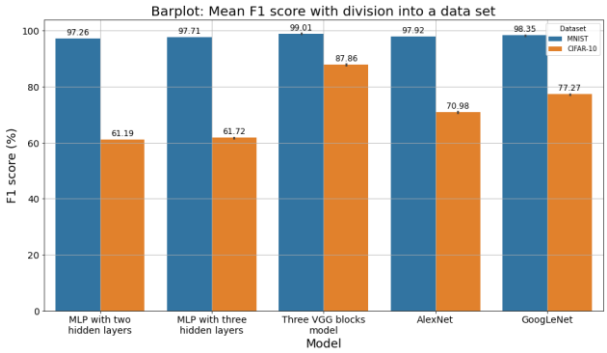

Figure 6: Mean F1–score for each model with division into a data set

As can be seen from the results the MLP–type architectures presented in this paper do a good job with a simple black–and–white MNIST dataset, but their classification accuracy and F1–score for the color CIFAR–10 dataset is much lower than that of the CNN–type architectures. However, the loss function value for both MLP architectures is lower than the value for GoogLeNet, but higher than for another two CNN–type architectures presented in this paper (Three VGG blocks and AlexNet). The three VGG blocks architecture showed the best results in terms of accuracy, loss function and F1 score. AlexNet architecture shows third best results in terms of accuracy and F1 score and second best result in term of loss function.

4. Results of second test

During the analysis of the results, the mean value of the final accuracy, loss function and F1–score were calculated. Also the mean value of the loss function, validation loss function, accuracy and validation accuracy for first, fifth, tenth, fifteenth and twentieth epoch were calculated. Figure 7 shows the change of the classification accuracy for each model during training.

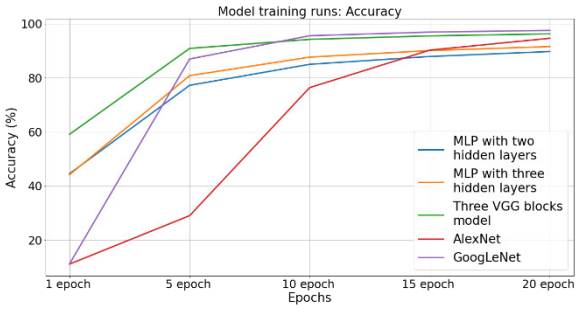

Figure 7: Change of the classification accuracy for each model

Figure 8 shows the change of the validation accuracy values for each model.

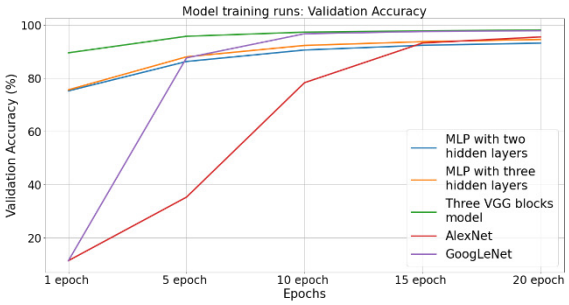

Figure 8: Change of the validation accuracy for each model

Figure 9 shows the change of the loss function values for each model during training.

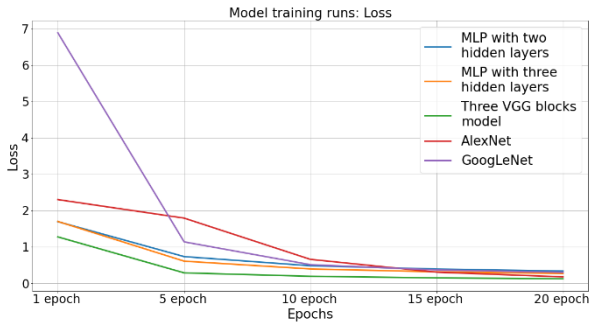

Figure 9: Change of the loss function values for each model

Figure 10 presents the change of the validation loss function values for each model.

Figure 10: Change of the validation loss function values for each model

The results show that initially AlexNet trains the slowest of all and, in terms of classification accuracy, catches up with MLP–type models only closer to the fifteenth epoch. GoogLeNet in the first epoch has worse results than MPL–type models, but quickly overtakes them after the fifth epoch. The three VGG blocks model has the best training speed and retains it throughout each epoch.

5. Conclusions

The aim of the study was to compare the performance of convolutional and traditional neural network architectures. During the research for this thesis, familiarization with machine learning and deep learning issues was required. Convolutional networks are the most widely used neural networks used for image classification tasks, and it was concluded during this study that CNN–type networks are the best choice for this purpose due to the accuracy of the classification and the smaller loss function, than MLP–type architectures. In terms of training speed, three VGG blocks network showed the best results while AlexNet showed the worst. These results are due to the filter size influencing the training speed. As mentioned in section 2, the VGG models use small filters and that's why three block VGG model have best results in training speed test. For the same reason, it cannot be argued that convolutional neural networks train faster than MLP–type networks. It also was confirmed that MLP networks give lower classification accuracy than CNN–based models when classifying color images, but give higher accuracy for simple black and white images comparing to them classifying color images. The obtained results partially confirm the main thesis put on beginning of work.

References

[1] MNIST handwritten digit database, http://yann.lecun.com/exdb/mnist [13.02.2021]

[2] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient–based learning applied to document recognition, Proceedings of the IEEE 86(11) (1998) 2278–2324.

[3] S. B. Driss, M. Soua, R. Kachouri, M. Akil, A comparison study between MLP and convolutional neural network models for character recognition, in SPIE Conference on RealTime Image and Video Processing, Anaheim, United States, 10–11 April (2017) 1022306.

[4] N. Sharma, V. Jain, A. Mishra, An analysis of convolutional neural networks for image classification, Procedia computer science 132 (2018) 377–384.

[5] J. M. Peña, P. A. Gutiérrez, C. Hervás–Martínez, J. Six, R. E. Plant, F. López–Granados, Object–based image classification of summer crops with machine learning methods, Remote Sensing 6(6) (2014) 5019–5041.

[6] D. X. Zhou, Universality of deep convolutional neural networks, Applied and computational harmonic analysis 48(2) (2020) 787–794.

[7] I. M. Dheir, A. S. A. Mettleq, A. A. Elsharif, S. S. Abu–Naser, Classifying Nuts Types Using Convolutional Neural Network, International Journal of Academic Information Systems Research 3(12) (2020) 12–18.

[8] Y. Li, J. Nie, X. Chao, Do we really need deep CNN for plant diseases identification?, Computers and Electronics in Agriculture 178 (2020) 105803.

[9] P. Sharma, Y. P. S. Berwal, W. Ghai, Performance analysis of deep learning CNN models for disease detection in plants using image segmentation, Information Processing in Agriculture 7(4) (2019) 566–574.

[10] J. G. A. Barbedo, Impact of data set size and variety on the effectiveness of deep learning and transfer learning for plant disease classification, Computers and electronics in agriculture 153 (2018) 46–53.

[11] P. T. T. Ngo, M. Panahi, K. Khosravi, O. Ghorbanzadeh, N. Karimnejad, A. Cerda, S. Lee, Evaluation of deep learning algorithms for national scale landslide susceptibility mapping of Iran, Geoscience Frontiers 12(2) (2020) 505–519.

[12] I. Banerjee, Y. Ling, M. C. Chen, S. A. Hasan, C. P. Langlotz, N. Moradzadeh, B. Chapman, T. Amrhein, D. Mong, D. L. Rubin, O. Farri, M. P. Lungren, Comparative effectiveness of convolutional neural network (CNN) and recurrent neural network (RNN) architectures for radiology text report classification, Artificial intelligence in medicine 97 (2019) 79–88.

[13] C. L. Chowdhary, M. Mittal, P. A. Pattanaik, Z. Marszalek, An efficient segmentation and classification system in medical images using intuitionist possibilistic fuzzy C–mean clustering and fuzzy SVM algorithm, Sensors 20(14) (2020) 3903.

[14] T. Nakaura, T. Higaki, K. Awai, O. Ikeda, Y. Yamashita, A primer for understanding radiology articles about machine learning and deep learning, Diagnostic and Interventional Imaging 101(12) (2020) 763–844.

[15] X. Yang, Y. Ye, X. Li, R. Y. Lau, X. Zhang, X. Huang, Hyperspectral image classification with deep learning models., IEEE Transactions on Geoscience and Remote Sensing 56(9) (2018) 5408–5423.

[16] A. F. Agarap, An architecture combining convolutional neural network (CNN) and support vector machine (SVM) for image classification, arXiv:1712.03541v2 (2019).

[17] Y. Sun, B. Xue, M. Zhang, G. G. Yen, J. Lv, Automatically Designing CNN Architectures Using the

Genetic Algorithm for Image Classification, IEEE Transactions on Cybernetics 50(9) (2020) 3840–3854.

[18] F. Sultana, A. Sufian, P. Dutta, Evolution of image segmentation using deep convolutional neural network: A survey, Knowledge–Based Systems 201–202 (2020) 106062.

[19] O. Sbai, C. Couprie, M. Aubry, Impact of base data set design on few–shot image classification, in Computer Vision–ECCV 2020: 16th European Conference, Glasgow, United Kingdom, August 23–28 (2020) 597–613.

[20] C. Gambella, B. Ghaddar, J. Naoum–Sawaya, Optimization problems for machine learning: a survey, European Journal of Operational Research 290(3) (2020) 807–828.

[21] Y. Wang, Y. Li, Y. Song, X. Rong, The Influence of the Activation Function in a Convolution Neural Network Model of Facial Expression Recognition, Applied Sciences 10(5) (2020) 1897.

[22] D. Bashir, G. D. Montanez, S. Sehra, P. S. Segura, J. Lauw, An Information–Theoretic Perspective on Overfitting and Underfitting, in AI 2020: Advances in Artificial Intelligence: 33rd Australasian Joint Conference, Canberra, Australia, November 29–30 (2020) 347–358.

[23] T. Kiran, Computer Vision Accuracy Analysis with Deep Learning Model Using TensorFlow, International Journal of Innovative Research in Computer Science & Technology (IJIRCST) 8(4) (2020) 2347–5552.

[24] T. P. P. Padilha, L. E. A. de Lucena, A Systematic Review About Use of TensorFlow for Image Classification and Word Embedding in the Brazilian Context, Academic Journal on Computing, Engineering and Applied Mathematics 1(2) (2020) 24–27.

[25] The CIFAR–10 dataset, https://www.cs.toronto.edu/~kriz/cifar.html [13.02.2021]