Shallow Convolutional Neural Networks for Pattern Recognition Problems

Oleksii Gorokhovatskyi
Informatics and Computer Technologies
Simon Kuznets Kharkiv National University of Economics
Kharkiv, Ukraine
oleksii.gorokhovatskyi@gmail.com

Olena Peredrii
Informatics and Computer Technologies
Simon Kuznets Kharkiv National University of Economics
Kharkiv, Ukraine

***Abstract*—Paper describes an investigation of possible usage of shallow (limited by few layers only) convolutional neural networks to solve famous pattern classification problems. Brazilian coffee scenes, SAT-4/SAT-6, MNIST, UC Merced Land Use and CIFAR datasets were tested. It is shown that shallow convolution neural networks with partial training may be effective enough to produce the result close to state-of-the-art deep networks but also limitations are found.**

***Keywords*— image; recognition; classification; convolution; shallow neural network; layer; partial training; dataset**

I. Introduction

Artificial neural networks (ANN), deep neural networks (DNN) and convolutional neural networks (CNN) last decades [1] became one of the most effective ways to resolve complex pattern recognition, classification and machine learning problems because of theirs power and huge flexibility. Besides reaching high accuracy ANN have two basic problems, first one is related to the choice of network structure that is effective enough, the other one is related with the requirement to have powerful hardware to train network. Structures of state-of-the-art CNN like PatreoNet [2], AlexNet [3], CaffeeNet [4] (based on AlexNet architecture), VGG [5], GoogLeNet [6] became computationally complex last years, that leads to situation when solving of challenging pattern recognition problems according to a lot of publications seems to be possible only with usage of special hardware or optimization routines like GPU calculations and usage of small (shallow) CNN is underestimated.

The idea of paper is to investigate if some well-known image classification problems may be resolved with shallow CNNs only, which are possible to train and use just on a typical personal computer without special hardware and GPU calculations.

II. Shallow Network Structure

One of the known problem with the implementation of a network is related to the huge amount of calculations, which may require a lot of time. This may be resolved with switching to GPU, that allows increasing speed significantly. Another problem is the complex net structure that may need a lot of memory. Finally, deep structure requires more parameters (like kernel size, quantity of neurons etc.) to be defined somehow. On the other hand, shallow NN may have more advantages to be used on mobile devices.

We are going to pay attention to such NN architectures and such training procedures that are possible to fit in memory at least partially and train in a reasonable amount of time.

A. Quantity of Layers

Looking at modern ANN like AlexNet [3] of GoogLeNet [6] or Microsoft ResNet [7] we may notice the huge amount of layers (from dozens up to hundreds) that require training time within days, weeks or even months using parallel GPU-calculations [8]. It is easy to find papers that describe the extremely effective solution of known pattern recognition problems using ANN mentioned above, but that doesn't mean that it is always reasonable to use NN with millions of parameters.

In order to make such CNN that is possible to train (within hours) and use (within seconds), we will consider shallow architectures that are limited by only ten layers including dropout ones.

B. Layer Types

The structure of traditional convolutional neural network is usually built as a sequence of layers of different neuron types that allows performing of specific operations on each stage of image processing. CNN usually consists of convolution, maxpooling, dropout and dense layers.

Convolution operator in computer vision and image processing problems is mostly used as a filtration layer that allows retrieving of specific features of an image. Let us denote image as I $((w)idth \times (h)eight$ size) and convolution kernel as K with typically small square odd size (3×3 or 5×5 are the most popular) of a kernel k. Mathematically, convolution result is represented as:

$$I * K = \sum_{i=1}^{k}\sum_{j=1}^{k} K_{i,j} I_{i+x-1,y+j-1},$$

$$where\ i = \overline{\lfloor k/2 \rfloor, w - \lfloor k/2 \rfloor},\ j = \overline{\lfloor k/2 \rfloor, h - \lfloor k/2 \rfloor}$$

Convolution means scanning of an image pixel by pixel overlaying with kernel window and computation of new values of convolved image with $(w-k+1)\times(h-k+1)$ size. Different kernel K values allow to apply the variety of specific filters like sharpening, blurring, edge detection etc. [9]. Additionally, convolution process may utilize other special parameters like stride or padding [10].

978-1-5386-2874-4/18/$31.00 ©2018 IEEE

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

Maxpooling is a popular downsampling approach that applies some aggregation (maximization operator mostly) to image parts to leave only the most valuable values. A typical size of such parts is 2×2, so each non-overlapping 2×2 part of image is replaced by a maximum of all values in this part.

Dense layers contain a combination of fully-connected neurons and usually used at the end of CNN structure to gather and generalize features after convolution and maxpooling layers. Training of dense layers is slower compared to other types, so only one or two such layers seems reasonable to use.

Dropout layers are used to prevent overfitting and speed up training process and the idea is to set zero values to some quantity of random input neurons. Exactly half of such inputs are dropped for this paper during training of a network.

C. Activation Function and Optimizer

A lot of different activation functions exist, but here only two types are used. Neurons in internal layers use rectifier activation according to

$$f(x) = max(0, x),$$

where x is the weighted biased input from the previous neuron. Such activation is a good choice to make training faster because of simple gradient and somewhat more effective due to zero reaction to negative inputs.

Last dense layers use sigmoid activation

$$f(x) = \frac{1}{1+e^{-x}}$$

to produce output in the range between 0 and 1.

Default options $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$ of Adam [11] optimizer were used for modeling. Stochastic gradient descent was tested too but Adam optimizer outperformed it easily both from speed and performance points of view.

III. Results Of Modelling

Usage of some shallow CNN architectures was tested on popular datasets defined below. Effective solutions for datasets A-C were found relatively easily, a lot of models were tested for each dataset, especially on those given in D section.

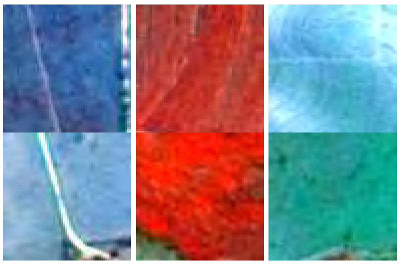
The modeling methodology was different dependently of original dataset structure, k-fold cross-validation was used for those which are already split to folds. Accordingly, if the dataset is split just on “Train” and “Test” part they were used for training and testing as a whole.

A. Brazilian Coffee Scenes

Brazilian Coffee Scenes dataset was proposed in [4, 12, 13] and it consists of 2876 RGB-NIR (Near-InfraRed) images of coffee and noncoffee plantations size of 64x64 pixels. Data was split by creators to 4 folds with 600 images each and the 5th fold with 476 images. All folds are balanced with coffee and noncoffee samples (50% each). We are focused on usage of only RGB channels and near-infrared channel was ignored.

This dataset is very challenging because of high interclass variance, different colorization of coffee regions and presence of distortions like shadows. Table I shows examples of coffee and noncoffee images (first and second row respectively) as well as samples of images that look pretty similar but belong to different classes (“coffee”-labeled images are above, similar noncoffee images are below).

TABLE I. Examples of Brazilian Coffee Scenes Dataset Images

Class label	**Sample images**
Coffee	
Noncoffee	
Coffee Vs Noncoffee	

Some known results of this dataset recognition based on different technologies are gathered in Table II. Best results are related to CaffeNet, that includes 5 convolution layers, 3 maxpooling and 2 dense layers, or GoogLeNet, that contains 22 layers.

TABLE II. Brazilian Coffee Scenes Dataset Known Results

Paper & method	**Results**
Border-Interior Pixel Classification (BIC) [2, 14]	87.03% ± 1.17%
Fine-tuned CaffeNet [4]	94.45% ± 1.20%
Quaternion Orthogonal Matching Pursuit Q-OMP [15]	90.75% ± 0.67%
Architecture II (LQPCANet – Linear Quaternion Principal Component Analysis) + GoogleLeNet [1]	88.46%
GoogLeNet [6, 16]	91.83%
Multiple lAyeR feaTure mAtching(MARTA) generative adversarial networks (GANs) [17]	88.36%

Let’s look at recognition accuracy of shallow CNN that is shown in Fig. 1. It contains 2 convolution layers, 2 maxpooling and 2 dense layers. Additionally, 2 dropout layers were used in between to reduce overfitting possibility.

Results of recognition were gathered with 5-fold cross-validation strategy. 10 independent training and recognition experiments were performed for each fold, results were averaged. The common score was obtained by averaging of all folds results. We were able to get 86.64% of correct recognition (with a maximum value of 89.67% and minimum 78.83%). Training time for separate fold was about 2 minutes

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

(software/hardware description that was used is available in section IV), 21 epochs were performed during training, each image was resized down to 32x32 pixels. Training of the whole fold was done in 32 image batches.

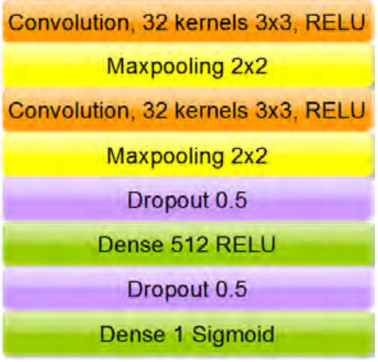
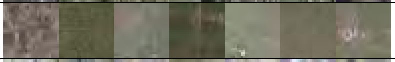


Fig. 1. CNN architecture to process Brazilian Coffee Scenes dataset.

B. *SAT-4 and SAT-6*

SAT-4 and SAT-6 airborne datasets were presented in [18, 19] and contain huge amount of different RGB-NIR aerial images with “barren land”, “trees”, “grassland” and “other” classes for SAT-4 and “barren land”, “trees”, “grassland”, “roads”, “buildings” and “water” classes for SAT-6. Each image has 28x28 pixels size and only one corresponding label, both datasets are challenging because of the huge amount of training and test images – 400000 and 100000 for SAT-4 and 324000 and 81000 respectively. Again, near-infrared channel was not used in this paper. Examples of SAT-6 images are shown in Table III.

TABLE III. EXAMPLES OF SAT-6 DATASETS IMAGES

Class label	Sample images
Building	
Barren land	
Trees	
Grassland	
Road	
Water	

Some results of this dataset processing based on different techniques are available in Table IV. As one can see very high results were achieved for both datasets.

TABLE IV. SAT-4 AND SAT-6 DATASETS KNOWN RESULTS

Paper & method	Results (SAT-4)	Results (SAT-6)
DeepSat [19]	97.95%	93.9%
SatCNN [20]	99.65%	99.54%
DropBand [21]	99.997%	99.994%
AlexNet, VGG [22]	99.98%	99.98%

Huge size of training and test set did not allow fitting them in memory at once even for shallow CNN that is presented in Fig. 2, so partial training process was introduced. Let’s denote with S such amount of images that is possible to load to memory and train, so full image dataset (training or testing) size of N is split to N/S parts. Current weights of CNN are saved after training of each part, that allows to free memory, and restored before training of next part. Training of each part S was performed in 128 image batches and 30% of images were chosen randomly as validation set.

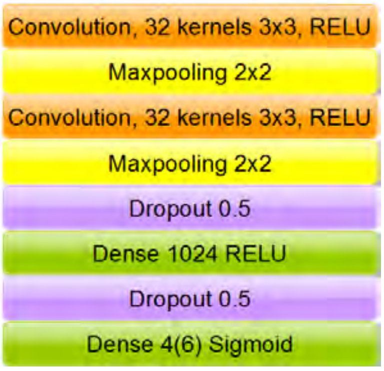


Fig. 2. CNN architecture to process SAT-4 and SAT-6 datasets.

Best modeling results we achieved are presented in Table V for SAT-4 and in Table VI for SAT-6 respectively. Each experiment was performed 5 times with averaging of scores and timings.

TABLE V. SAT-4 RECOGNITION RESULTS

Size of images	Epochs per part	Size of part S	Recognition rate	Training time for the whole dataset
20x20	1	50000	94.81%	10 min.
32x32	15	2000	97.87%	3 hrs. 20 min.
32x32	10	4000	97.99%	3 hrs. 20 min.

It is possible to see that S value should be chosen properly because neither too small nor too big values don’t allow to get best results. Also, it may be noticed that upscaling of images is preferable than downscaling. Correct recognition rate is high and comparable to DeepSat [19] investigation but is not high enough to be comparable directly to state-of-the-art approaches.

TABLE VI. SAT-6 RECOGNITION RESULTS

Size of images	Epochs per part	Size of part S	Recognition rate	Training time for the whole dataset
20x20	1	40500	96.15%	7 min.
32x32	15	2000	97.86%	2 hrs. 40 min.
32x32	10	4000	98.34%	1 hr. 40 min.

C. *MNIST*

MNIST [23, 24] is the other famous dataset of handwritten digit images that contains 60000 train images size of 28x28 and 10000 of test ones. The architecture of CNN is presented in Fig. 3, it allows to achieve average

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

98.52% of correct recognition rate with 10 epochs of training per part with size $S = 5000$ and resizing of input images to 32x32. Order of train files was randomized before each of 5 experiments. Training of the whole dataset in this setup took about 1 hr. and 15 min. As earlier, 30% of samples in each part were used as validation data, changing of weights during learning was done in 128 batches.

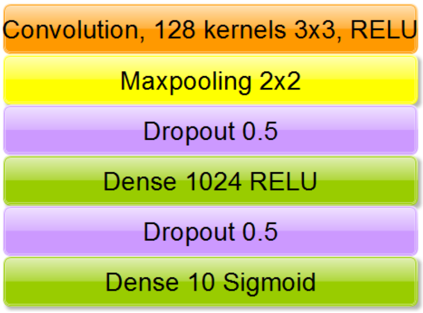


Fig. 3. CNN architecture to process MNIST dataset.

D. UC Merced Land Use and CIFAR

UC Merced Land Use dataset was introduced in [25] and contains 2100 aerial images of size 256×256 pixels that are split into 21 classes: agricultural, airplane, baseball diamond, beach, buildings, chaparral, dense residential, forest, freeway, golf course, harbor, intersection, medium residential, mobile home park, overpass, parking lot, river, runway, sparse residential, storage tanks and tennis courts. UC Merced Land Use dataset is very popular [4, 26 - 28] with high result reported above 99%.

CNN, which is presented in Fig.4, was used to perform 5-fold cross validation recognition. Dataset was split to 5 folds with 420 images in each with balanced amount of every class representatives. 3 folds were used for training, another one for validation and the last one for testing. Training was performed 50 times with 30 epochs each time, the average correct recognition rate is 85.96% (minimum value is 81.86%, maximum one is 88.54%). Full training time was about the hour, weights for 64 images were updated simultaneously.

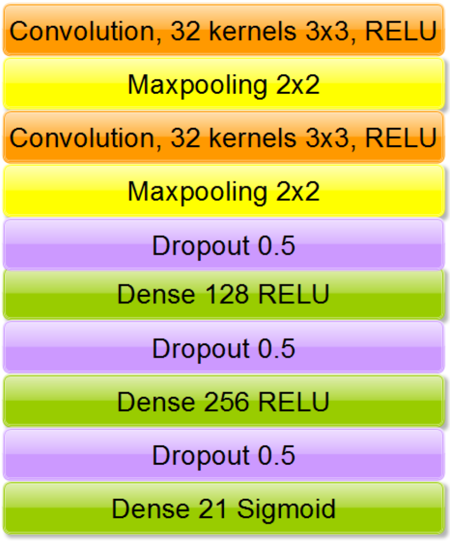


Fig. 4. CNN architecture to process UC Merced Land Use dataset.

Similar recognition rate (85.67%, minimum is 81.9%, the maximum is 89.52%) may be reached with lower CNN, e.g. network shown in Fig.1 with corresponding amount of output neurons, or even with less training iterations or epochs. But it is much harder to achieve better accuracy with such shallow architectures for this dataset.

CIFAR-10 and CIFAR-100 [29, 30] are datasets of tiny (32x32) color images representing 10 and 100 non-overlapping classes respectively, each dataset has 50000 of train and 10000 of test images. Best recognition results reported are over 96% for CIFAR-10 and over 75% for CIFAR-100 [31].

Applying CNN, that is shown in Fig. 4, allows getting 71.7% of correct recognition rate with training time about 4 hrs. We set up the size of the part to be $S = 2000$ and performed 10 epochs per part during training and 20 iterations. Increasing this quantity up to 30 with part size modification $S = 4000$ allows to improve recognition rate up to 74% with training time about of 6 hrs.

IV. TECHICAL NOTES

Results and all reported timings were achieved with Keras [32] deep learning library using default Theano backend, Python 2.7 programming language and without GPU-optimization. Hardware included a personal computer with Intel Core i7 4x 3.60GHz processor and 16GB RAM.

V. CONCLUSIONS

Resolving of practical pattern recognition (classification) problem using CNN seems to be related to the complex structure of network usually but it is possible to get similar results using only shallow networks like we presented on Fig.1 – Fig. 4. Not every shallow model is successful and not every problem may be resolved using this approach though, this was confirmed by testing of shallow models with different options.

Looking at datasets we successfully applied presented approach on (Brazilian coffee, SAT, MNIST) we may guess about the requirement for a problem to be resolved with shallow CNN. Samples of all these datasets have more common image information like color or shape on the same background in case of MNIST images. This information is mostly retained after downscaling to small size, besides that, all images are small initially.

Ways to recognize such datasets as CIFAR and UC Merced Land Use effectively enough with shallow networks were not found. Looks like CIFAR samples have important features which shallow CNN are unable to catch, whilst UC Merced Land Use images have 256x256 size and most details seem to be lost after downscaling.

Partial training approach we used in paper should be investigated deeply as that's unclear for now how training of each separate part influences other parts and the whole model.

REFERENCES

[1] J. Wang, C. Luo, H. Huang, H. Zhao and S. Wang, "Transferring Pre-Trained Deep CNNs for Remote Scene Classification with General Features Learned from Linear PCA Network," Remote Sens. 2017, 9(3), 225; doi:10.3390/rs9030225.

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua

[2] K. Nogueira, W. O. Miranda and J. A. Dos Santos, “Improving spatial feature representation from aerial scenes by using convolutional networks,” in: 28th IEEE SIBGRAPI Conference on Graphics, Patterns and Images, pp. 289–296, 2015.

[3] A. Krizhevsky, I. Sutskever and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Neural Information Processing Systems, pp. 1106–1114, 2012.

[4] K. Nogueira, O. A. B. Penatti and J. A. Dos Santos, “Towards Better Exploiting Convolutional Neural Networks for Remote Sensing Scene Classification,” [Online]. Available: https://arxiv.org/pdf/1602.01517.pdf [June 02, 2018].

[5] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” [Online]. Available: https://arxiv.org/pdf/1409.1556.pdf. [March 15, 2017].

[6] C. Szegedy, W. Liu, Y. Jia et al., “Going deeper with convolutions,” [Online]. Available: https://arxiv.org/pdf/1409.4842.pdf. [March 10, 2017].

[7] K. He, X. Zhang, S. Ren and J. Sun, “Deep Residual Learning for Image Recognition,” [Online]. Available: https://arxiv.org/pdf/1512.03385v1.pdf. [May 12, 2017].

[8] A. Deshpande, “The 9 Deep Learning Papers You Need To Know About (Understanding CNNs Part 3),” [Online]. Available: https://adeshpande3.github.io/adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html. [May 15, 2017].

[9] V. Powell, “Image Kernels,” [Online]. Available: http://setosa.io/ev/image-kernels/. [June 02, 2017].

[10] P. Veličković, “Deep learning for complete beginners: convolutional neural networks with keras,” [Online]. Available: https://cambridgespark.com/content/tutorials/convolutional-neural-networks-with-keras/index.html. [June 02, 2017].

[11] D. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” [Online]. Available: https://arxiv.org/abs/1412.6980v8.pdf. [July 20, 2017].

[12] Brazilian Coffee Scenes Dataset [Online]. Available: http://www.patreo.dcc.ufmg.br/downloads/brazilian-coffee-dataset/. [May 17, 2017].

[13] O. A. B. Penatti, K. Nogueira and J. A. Dos Santos, “Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?,” in IEEE Computer Vision and Pattern Recognition Workshops, pp. 44–51, 2015.

[14] R. de O. Stehling, M. A. Nascimento and A. X. Falcao, “A compact and efficient image retrieval approach based on border/interior pixel classification,” In Eleventh International Conference on Information and Knowledge Management (CIKM'02), pp.102–109, 2002.

[15] V. Risojevic and Z. Babic, “Unsupervised Quaternion Feature Learning for Remote Sensing Image Classification,” [Online]. Available: http://dsp.etfbl.net/aerial/unsupervised_final.pdf. [June 15, 2017].

[16] M. Castelluccio, G. Poggi, C. Sansone and L. Verdoliva, “Land Use Classification in Remote Sensing Images by Convolutional Neural Networks,” [Online]. Available: https://arxiv.org/pdf/1508.00092.pdf. [June 17, 2017].

[17] DaoYu Lin, “Deep Unsupervised Representation Learning for Remote Sensing Images,” [Online]. Available: https://arxiv.org/pdf/1612.08879.pdf. [June 20, 2017].

[18] SAT-4 and SAT-6 airborne datasets [Online]. Available: http://csc.lsu.edu/~saikat/deepsat/. [June 20, 2017].

[19] S. Basu, S. Ganguly, S. Mukhopadhyay, R. DiBiano, M. Karki and R. Nemani, “DeepSat – A Learning framework for Satellite Imagery,” [Online]. Available: http://bit.csc.lsu.edu/~saikat/publications/sigproc-sp.pdf. [June 20, 2017].

[20] Y. Zhong, F. Fei, Y. Liu, B. Zhao, H. Jiao and L. Zhang, “SatCNN: satellite image dataset classification using agile convolutional neural networks,” Remote Sensing Letters, 8:2, 136-145, DOI: 10.1080/2150704X.2016.1235299. [Online]. Available: http://dx.doi.org/10.1080/2150704X.2016.1235299. [June 23, 2017].

[21] N. Yang, H. Tang, H. Sun and X. Yang, “DropBand: a convolutional neural network with data augmentation for scene classification of VHR satellite images,” [Online]. Available: http://proceedings.utwente.nl/403/1/Yang-DropBand-91.pdf. [June 25, 2017].

[22] M. Papadomanolaki, M. Vakalopoulou, S. Zagoruyko and K. Karantzalos, “Benchmarking deep learning frameworks for the classification of very high resolution satellite multispectral data,” in ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume III-7, XXIII SPRS Congress, Prague, Czech Republic, 12–19 July 2016.

[23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, 86(11):2278-2324, November 1998.

[24] Y. LeCun, C. Cortes and C.J.C. Burges, “THE MNIST DATABASE of handwritten digits,” [Online]. Available: http://yann.lecun.com/exdb/mnist/. [July 10, 2017].

[25] Y. Yang and S. Newsam, “Bag-of-visual-words and spatial extensions for land-use classification,” in: 18th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, pp. 270– 279, November 02 – 05, 2010.

[26] M. Castelluccio, G. Poggi, C. Sansone and L.Verdoliva, “Land Use Classification in Remote Sensing Images by Convolutional Neural Networks,” [Online]. Available: https://pdfs.semanticscholar.org/4191/fe93bfd883740a881e6a60e54b371c2f241d.pdf. [July 26, 2017].

[27] F. P. S. Luus, B. P. Salmon, F. van den Bergh and B. T. J. Maharaj, “Multiview Deep Learning for Land-Use Classification,” in IEEE Geoscience and Remote Sensing Letters, Vol. 12, pp. 2448 – 2452, 2015.

[28] Y. Zhong, F. Fei and L. Zhang, “Large patch convolutional neural networks for the scene classification of high spatial resolution imagery,” J. Appl. Remote Sens. 10(2), 025006 (2016), doi: 10.1117/1.JRS.10.025006.

[29] The CIFAR-10 dataset [Online]. Available: https://www.cs.toronto.edu/~kriz/cifar.html. [July 27, 2017].

[30] A. Krizhevsky, “Learning Multiple Layers of Features from Tiny Images,” [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf. [July 27, 2017].

[31] Classification datasets results, [Online]. Available: http://rodrigob.github.io/are_we_there_yet/build/classification_datasets_results.html. [July 27, 2017].

[32] F. Chollet, “Keras,” [Online]. Available: https://github.com/fchollet/keras. [July 30, 2017].

Lviv Polytechnic National University Institutional Repository http://ena.lp.edu.ua