Accepted manuscript: Tusień, E., Wilke, A., Woźna, J., Czarnul, P. (2022). Food Classification from Images Using a Neural Network Based Approach with NVIDIA Volta and Pascal GPUs. In: Saeed, K., Dvorský, J. (eds) Computer Information Systems and Industrial Management. CISIM 2022. Lecture Notes in Computer Science, vol 13293. Springer, Cham. https://doi.org/10.1007/978-3-031-10539-5_20

Food classification from images using a neural network based approach with NVIDIA Volta and Pascal GPUs

Ewa Tusień, Aleksandra Wilke, Joanna Woźna,
Pawel Czarnul[0000−0002−4918−9196]

Faculty of Electronics, Telecommunications and Informatics
Gdansk University of Technology, Narutowicza 11/12, 80-233 Poland
pczarnul@eti.pg.edu.pl

Abstract. In the paper we investigate the problem of food classification from images, for the Food-101 dataset extended with 31 additional food classes from Polish cuisine. We adopted transfer learning and firstly measured training times for models such as MobileNet, MobileNetV2, ResNet50, ResNet50V2, ResNet101, ResNet101V2, InceptionV3, InceptionResNetV2, Xception, NasNetMobile and DenseNet, for systems with NVIDIA Tesla V100 (Volta) and NVIDIA GTX 1060 (Pascal) GPUs. We presented inference times corresponding to training the various considered network models, both using a desktop NVIDIA GTX 1060 GPU and an Intel i7-7000 CPU. Subsequently, we investigated the InceptionV3 model in more detail, best in the preliminary tests, regarding the impact of both learning rates (including both various fixed and variable rates) as well as batch sizes on the accuracy of classification, along with training times for various batch sizes. This allowed to identify better learning rate configurations as well as classification performance versus training time.

Keywords: deep neural networks, food classification, GPUs, inference, neural network training

1 Introduction

Topics related to food have become very important, especially in the context of globalization, when we have opportunities to visit many places that considerably differ in terms of cuisine. Food recognition, which is the subject of this paper along with modern deep learning based algorithms, is of interest to many people as it helps to identify what we eat. Development of an effective classifier as well as assessment of inference performance is important in the context of wide adoption of mobile devices. Mobile devices are widely used, therefore these constitute the best medium for reaching future users for such applications.

In this article, we discuss the problem of image recognition using a convolutional neural network approach. Convolutional neural networks are a type of deep neural networks which are mostly applied to the problem of image classification, as shown in [10]. The contribution of this paper is as follows:

1. Initial assessment of performance of 11 various network models: MobileNet, MobileNetV2, ResNet50, ResNet50V2, ResNet101, ResNet101V2, Inception V3, InceptionResNetV2, Xception, NasNetMobile and DenseNet for food classification using transfer learning, specifically regarding models' ability to obtain top-1 and top-5 accuracies, in the context of learning time, for two distinct and representative hardware setups: server/workstation NVIDIA Tesla V100 and desktop NVIDIA GTX 1060. Comparison of times gives an indication of what performance we can expect in a datacenter/cloud versus home environment, the latter could be enaged in volunteer systems.
2. Investigation of the performance of all the models for the Food-101 vs a data set of the Food-101 extended with 31 additional Polish food classes, downloaded from the Internet. The latter can be thought of the type of images taken by users with their smartphones on daily basis.
3. Comparison of inference times using all the models for the NVIDIA GTX 1060 GPU and Intel(R) Core(TM) i7-7700 CPU, of interest to end users of food classification applications.
4. Detailed investigation of the impact of various learning rates including variable learning rates as well as batch sizes on both final performance of the best identified model among the tested ones – InceptionV3, including assessment of training times for various settings.

2 Related work and motivations

In this section, progress on algorithms as well as benchmarking of food detection and classification is summarized, especially in the context of accuracy obtained for particular algorithms as well as, what is important, numbers of food categories.

Authors of paper [6] proposed a very practical approach to food image recognition (aimed at recording eating habits) using mobile phones. Specifically, a Multiple Kernel Learning (MKL) method was used for integration of image features such as color, texture as well as SIFT. They obtained the accuracy of 61.34% for 50 types of food.

In paper [7] authors focused on exploration of hyper parameters for accuracy of food recognition using a Convolutional Neural Network (CNN), specifically number of layers, kernels, sizes of kernels and normalization. For a data set with 10 most frequent food items from a 170 000 set of images acquired from FoodLog. Images were scaled to 64x64. Best accuracy obtained was 73.7% while food detection 93.8%.

Authors of [1] used a Random Forest to cluster superpixels of a training set. For classification, superpixels of an input image are scored using component models and a multi-class SVM with spatial pooling is used to predict the final class. The Food-101 data set (101 food categories with 1000 images each) with 750 images of each class are used for training and the remaining 250 for testing. The authors achieved an average accuracy of 50.76% which is better than MLDS and IFV by 8.13% and 11.88% but worse than CNN (56.40%).

In work [9] authors combined features obtained from a pre-trained Deep Convolutional Neural network on a LSVRC 1000-class dataset with Fisher Vectors with HoG and Color patches. For the UEC-FOOD100 100-class food dataset they achieved the top-1 accuracy of 72.26% and the top-5 accuracy of 92.00%. The same authors, in paper [24] extended their previous work and used a fine-tuned DCNN pre-trained with 2000 categories in the ImageNet (with 1000 food categories). They achieved the top-1 accuracy of 78.77% for the UEC-FOOD100 set and 67.57% for the UEC-FOOD256 dataset. They also mentioned the 0.03 second time for food image classification using a GPU (NVIDIA Titan Black).

Authors of [25] used a five-layer CNN for recognition using a 100-class food dataset with about 15000 for accuracy of 80.8% and a fruit dataset with approximately 40000 images (30 kinds) for accuracy of 60.9%. A part of research included by [23], apart from food/non-food classification, was recognizing the type of a food in an image. A dataset called Food-11 with 11 classes and 16643 images was used to train and test a model. A modified CNN GoogLeNet was used: 11 classes learning rate of 0.001 and policy polynomial. The authors obtained the maximum accuracy of 83.5%, the maximum values of F-measure and kappa coefficients of 0.911 and 0.816 respectively. In paper [13] the author used a dataset with 5822 images of ten categories. A bag-of-features (BoF) model together with a support vector machine (SVM) returned accuracy of 56% while a plain five-layer CNN gave accuracy of 74%. Furthermore, data augmentation techniques through geometric transformations allowed to increase the training data size and accuracy to over 90%.

In paper authors [12] presented a CNN based solution for food image recognition. The solution uses two Inception modules (with additional convolutional layers) connected via an additional max pooling layer. The network has 22 layers with parameters. 70% dropout is used in the approach. For the UEC-256 set with 256 categories with a total of 28375 images, the proposed approach allowed to obtain top-1 accuracy of 54.7%, top-5 accuracy of 81.5%. For UEC-100, corresponding results were 76.3% and 94.6% while for Food-101 77.4% and 93.7%. Adding bounding boxes improved top-1 accuracy for UEC-256 to 63.8%.

Authors of [5] used a tuned Inception V3 network architecture for food recognition using the ETH Food-101, UEC FOOD 100 and UEC FOOD 256 data sets, for which they obtained the top-1 accuracies of 88.28%, 81.45% and 76.17% while top-5 accuracies of 96.88%, 97.27% and 92.58%.

In paper [2] authors used UNIMIB2016 food data set collected in a real canteen environment and performed segmentation into 73 food classes. Finally, they used 1010 tray images and 65 classes with partitioning into 70% for training and 30% testing sets. CNN4096 features with the combination of posterior probability strategy (from global and local) returned the best performance of 78.9% (SVM).

In paper [19] authors proposed a new network model Ensemble Net that includes histogram and equalization layer followed by parallel assessment using fine tuned AlexNet, fine tuned GoogLeNet and fine tuned ResNet. Experimental results were performed on two data sets: ETH Food-101 with 1000 images per

class with 101 classes as well as an Indian food database which includes images divided into 50 food classes, each with 100 images. For the former, Ensemble Net reached 72.12% top-1 accuracy and 91.61% top-5 accuracy while for the latter it reached 73.5% top-1 and 94.4% top-5 accuracy, outperforming AlexNet, GoogLeNet and ResNet.

This work is similar to [4] where authors combined a new Turkish cuisine dataset with Food-101 and deep learning was applied for the combined set of 113 classes. Tests were performed for learning rates of 0.1, 0.3 and 0.7 with batch size equal to 100. In that comparison the best accuracy of 62.7% was obtained for the learning rate of 0.3 which was further improved in longer training to 68.2%. Validation cross entropy of around 1.3 was reported while for training around 0.7.

Authors of [15] performed comparison of performance of various models for food and drink recognition. Specifically, they compared four architectures including AlexNet, GoogLeNet, ResNet and NutriNet. Three solvers were tested: SGD, NAG and AdaGrad. For 512x512 images, best test accuracies were obtained by the 512×512 version of ResNet with NAG that achieved the accuracy of 87.96% as compared to the best NutriNet with AdaGrad of 86.72% which turned out to be 1.93% better than its AlexNet version. On the other hand the authors argue that NutriNet si significantly faster (approx. 5x) to train than ResNet. The dataset was divided into training, validation and testing sets proportionally to 70%, 10% and 20%, for a total of 225953 images of 520 food and drink items.

The research shown by authors of [17] confirms that food recognition using neural networks for a small number of categories can result in really high accuracy values. Specifically, for food images taken from personal life archives from life loggers, for a total of 14760 images of just eight different foods, the authors obtained 91.67% for AlexNet and 95.97% for GoogLeNet for test sets.

Authors of paper [20] proposed a Deep Convolutional Neural Network food recognition model, K-foodNet for recognition of Korean food and conducted experimental comparison vs AlexNet, GoogLeNet, VGG-19 and ResNet-18. For a data set with 23 food categories, which was divided into training and test images with 69000 and 23000 images in each set respectively, the proposed model achieved best results obtaining the test accuracy of 91.3% albeit with a noisy loss function. The authors argue that Korean food is reasonably complex to recognize, especially to other national food items. The authors struggled with the problem of too many similar, augmented images.

In paper [18] authors used the Food-41 dataset (4100 images and 41 classes) and partitioned it into parts 60% – training, 20% – validation and 20% – testing after resizing into 640x480 pixel images. They used Keras, GTX 1070 and a proposed CBNet that uses output from auxiliary classifiers (such as ResNet50, VGG19, DenseNet121) and performs fusion for final prediction. Generally, CBNet solutions returns better accuracies than best single models, both for tuning the last layer: CBNet-VD with 89.47% vs VGG19-AVG with 88.82% and the overall network: CBNet-RD with 95.28% vs DenseNet121 with 93.78%.

3 Problem formulation and approach

The main purpose of this research is to investigate a neural network based approach to food classification. We considered 11 models of artificial neural networks: MobileNet, MobileNetV2, ResNet50, ResNet50V2, ResNet101, ResNet101 V2, InceptionV3, InceptionResNetV2, Xception, NasNetMobile and DenseNet. During analysis the following parameters are taken into consideration: prediction accuracy, the time of training a neural network and the time of models' inference. All of the measurements are taken on each of the following hardware: two GPUs: NVIDIA Tesla V100 and NVIDIA GTX 1060; and Intel(R) Core(TM) i7-7700 CPU. The implementations of models are obtained from Keras Applications. The library contains popular deep learning models which are available with pre-trained weights. Each model selected for this research is prepared as follows: import a model from Keras Applications with weights trained on ImageNet, attach classification layers: *GlobalAveragePooling2D*, *Dense* with arguments: *units* - number of classes and *activation* - an element-wise activation function activation (with value *softmax*).

The main assumption adopted for the design of this investigation is the approach to the problem of food recognition based on pictures of dishes. The problem has non-trivial solutions because of the similarity of classes in a dataset. There is a strong conviction that the complexity of this classification problem will convey much more valuable results of tests in comparison with many elementary problems, e.g. a binary classification task.

In machine learning, classification is an example of the common problem of pattern recognition. The popularity of classification problem and significant resources of pre-trained models on various data have a tremendous impact on the decision of applying transfer learning in the presented solution. Transfer learning speeds up training, improves the performance of neural networks and circumvents the need for lots of new data. It is for these reasons that pre-trained models are commonly used for obtaining better results.

4 Training and validation data

The Food-101 dataset – the first public collection of dishes with such a large number of photos - has been chosen as the base dataset for network training. It is owned by the Federal Institute Technology in Zurich (ETHZ). It contains 101 ordered food categories with 1000 images each. Dimensions of a single image from this dataset are not uniform - photos reach range between 512x317 pixels and 512x512 pixels, while the size of the photo is approximately 45 KB. We further created an extended data set containing 31 Polish dishes using script downloading photos based on Google search results and manual selection of suitable photos to use.

The images in the dataset had been pre-processed. A change to the RGB from [0, 255] to range [0, 1] value range was used on the extended dataset. Another important aspect of pre-processing is swapping RGB order values to BRG order.

Table 1 presents the characteristics of the pre-processing used to transform data into various models in the Keras library. Data augmentation was also used on the extended dataset through cropping, padding and horizontal flipping.

Table 1. Pre-processing dedicated to models from Keras library

Model name	Photo size (pixels, pixels)	Order RGB value	Range RGB value	Other transformations
VGG16, VGG19	(224, 224)	BGR	No scaling	Pixel values scaled to an average equal zero
ResNet 50, 101	(224, 224)	BGR	No scaling	Pixel values scaled to an average equal zero
ResNetV2 50, 101	(224, 224)	RGB	[-1, 1]	—
InceptionV3	(299, 299)	RGB	[-1, 1]	—
Xception	(299, 299)	RGB	[-1, 1]	—
InceptionRes NetV2	(299, 299)	RGB	[-1, 1]	—
MobileNet	(224, 224)	RGB	[-1, 1]	—
MobileNetV2	(224, 224)	RGB	[-1, 1]	—
DenseNet 121	(224, 224)	RGB	[0, 1]	Normalization
NASNetMobile	(224, 224)	RGB	[-1, 1]	—

The dataset has been divided into the three parts: training data (70%), test data (15%) and evaluation data (15%).

5 Experimental results

5.1 Preliminary results for various models

During training, all models were tested for top-1 and top-5 accuracy and cross-entropy loss function recommended in classification problems. Cross entropy has been favorably compared to quadratic loss for classification by [3], using the CIFAR 100 dataset. For the optimizer, it was decided to use the Stochastic Gradient Descent (SGD) method. It is a very popular and common algorithm used in various machine learning algorithms. Its popularity is due to the introduction of randomization in the algorithm, which significantly contributed to reducing the number of computational operations. The study used the SGD method with two parameters: learning rate of 0.01 and momentum parameter of 0.9. To prevent over-training of the network, the early stopping method was used. If the loss function on the validation dataset does not receive smaller values for 5 learning periods, then the training process stops. Furthermore, to get the best result that the model can achieve, the evaluation accuracy was checked every epoch, and if the result was better than in the previous model, the model was saved. Thanks to this, at the end of each training process we obtained the best model. After each training an evaluation process started. The evaluation dataset consisted

of unique photos of dishes, which were used neither in the training nor in the validation process.

Firstly, all models were trained on the Food-101 dataset, all of which exceeded 80% top-1 accuracy as indicated in Table 2. The InceptionV3 model obtained top-1 accuracy 87.63%, which renders it as the best of all tested models.

Table 2. Test top-1 and top-5 accuracies for dataset Food-101

Model name	Epochs	top-1	top-5
InceptionV3	20	0.876	0.969
DenseNet	17	0.859	0.973
MobileNet	18	0.844	0.965
Xception	10	0.832	0.965
ResNetV50	15	0.830	0.966
ResNet101V2	18	0.828	0.963
ResNet101	19	0.826	0.963
ResNet50V2	13	0.822	0.954
InceptionResNetV2	10	0.819	0.961
NASNetMobile	18	0.818	0.957
MobileNetV2	27	0.813	0.958

Table 3. Test top-1 and top-5 accuracies for extended dataset

Model name	Epochs	top-1	top-5
InceptionV3	19	0.833	0.954
MobileNetV2	16	0.797	0.942
MobileNet	16	0.793	0.941
ResNet101	20	0.793	0.940
NASNetMobile	17	0.782	0.933
ResNetV50	17	0.775	0.927
ResNet101V2	19	0.764	0.924
Xception	9	0.751	0.922
ResNet50V2	11	0.748	0.921
InceptionResNetV2	8	0.732	0.921
DenseNet	10	0.694	0.886

Then all models were trained on an extended set of Food-101 with 31 additional dish classes. Results are presented in Table 3. As in the previous study, the InceptionV3 model obtained the best accuracy equal to 83%, whereas other models' results have not dropped below 70%. It is noteworthy that for all architectures, the accuracy of top-1 and top-5 has decreased compared to only Food-101.

Figure 1 presents the top-1 accuracy of the Food-101 dataset compared to the trainable number of parameters. The latter is usually used to approximate learning time because it tells us how many parameters need to be calculated and corrected during the learning process.

Table 4 presents the average training time per epoch for each model. The test was carried out on Nvidia's graphic cards GTX 1060 with 6GB memory and Tesla V100 with 16GB memory. Due to the difference in memory size, the batch size for each model for the GTX card is 12 and for Tesla is 32. Besides, the determining factor may be the input size of the model because, for InceptionV3, Xception and InceptionResNetV2 the size is 299x299 pixels, while for the rest of models is 244x244 pixels. These models were designed in this size, thus the difference was accepted in this experiment.

Whereas the number of different factors, the results can only be used to estimate the duration of learning on these machines. This result shows that if on Tesla V100 full training of InceptionV3 takes 20 epochs what overall is 3.5 hours that on GTX 1060 it could take more than 9 hours.

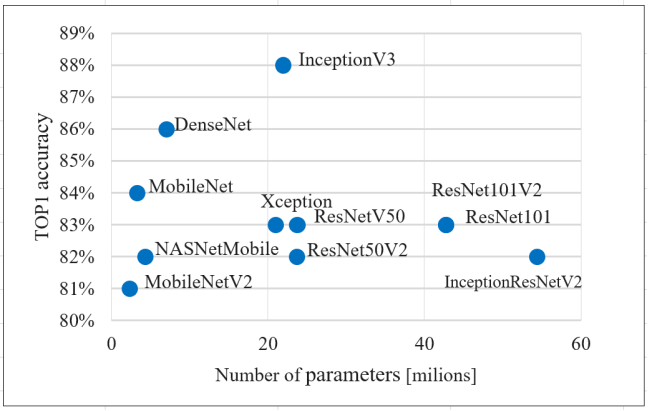

Fig. 1. top-1 accuracy compared to the number of parameters

Table 4. Training time on GTX 1060 and Tesla V100 graphic card

Model name	Average training time per one epoch [s]	
	Tesla V100	GTX 1060
MobileNet	330.06	1095.86
ResNet50V2	351.26	1160.27
MobileNetV2	379.84	806.40
ResNet50	402.86	1367.44
DenseNet	465.09	1376.77
InceptionV3	586.05	1642.05
ResNet101V2	601.94	2024.22
ResNet101	649.88	2242.47
NasNetMoblie	839.12	1379.56
InceptionResNetV2	1236.61	2317.98
Xception	1502.04	3691.54

5.2 Inference times using CPU vs GPU

Following the preliminary tests for various models, we carried out experiments for inference time tests on the CPU and GPU. A desktop class Intel(R) Core(TM) i7-7700 CPU and NVIDIA GTX 1060 with memory 6 GB were used which can be regarded as representative of desktop systems that could be used by typical end users. Two virtual Python environments were created, in which Tensorflow on a CPU was installed on one, and Tensorflow on a GPU on the other. The test was carried out in such a way that one image and one network model were loaded into memory, after which the inference process was started. Time was measured only during the prediction process, 10 times for each model. Results are presented in Table 5 and are very similar between CPU and GPU, not exceeding 11.3% (0.12s) – the largest for InceptionV3. Differences between GPU vs CPU times typically stem from additional CPU-GPU copy and specific GPU kernel configuration like grid size, memory usage etc. and could be further investigated for more details.

Table 5. Inference times using CPU vs GPU

Model name	Average time per picture [s]		Comparison
	GPU Nvidia GTX 1060	CPU Intel Core i7-7700	GPU to CPU
InceptionV3	0.223	0.198	11.24%
NASNetMobile	0.921	0.882	4.17%
DenseNet	0.954	0.936	1.83%
ResNet50V2	0.435	0.436	-0.28%
ResNet50	0.398	0.404	-1.54%
MobileNetV2	0.189	0.192	-1.57%
MobileNet	0.135	0.137	-1.71%
Xception	0.168	0.171	-1.73%
InceptionResNetV2	0.544	0.561	-3.06%
ResNet101V2	0.648	0.681	-5.07%
ResNet101	0.572	0.608	-6.23%

5.3 Training and results for various parameters for InceptionV3 model

Following the preliminary results, we have decided to analyze the InceptionV3 model in more detail, using NVIDIA Tesla V100. Specifically, we present detailed results for various training parameters such as various batch sizes as well as learning rates, including variable learning rates showing how these affect the final performance of the model.

Firstly, we have performed learning for various constant values of the learning rate observing final top-1 accuracy, after 25 epochs, for learning rates 0.2, 0.01 and 0.001 respectively. As a reference at this point, for the learning rate of 0.001 the following precision and recall values were obtained:

- batch size 16: precision 0.972 and recall 0.947 for training and 0.852 and 0.811 for validation,
- batch size 32: precision 0.980 and recall 0.960 for training and 0.857 and 0.805 for validation,
- batch size 64: precision 0.979 and recall 0.954 for training and 0.860 and 0.801 for validation.

Subsequently, we have varied learning rate values along the process:

- 0.02 (epochs 1-7), 0.01 (epochs 8-17), 0.001 (epochs 18-25),
- 0.01 (epochs 1-10), 0.001 (epochs 11-20), 0.0001 (epochs 21-25).

All the results for top-1 and top-5 test accuracies and various batch sizes are shown in Figure 2 and Figure 3 respectively. We can draw the following conclusions based on the presented results:

1. In terms of accuracy in the transfer learning used in this work, we can see that for the constant learning rate 0.01 allows to achieve slightly higher accuracy

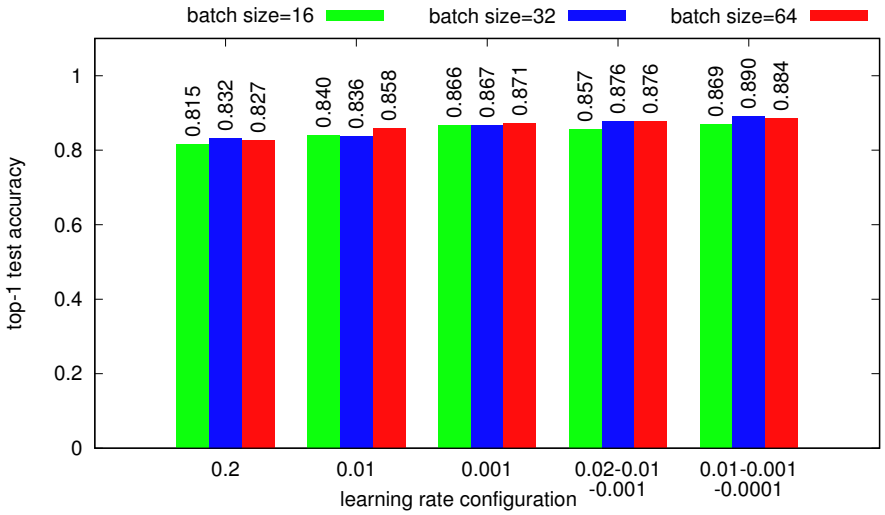

Fig. 2. top-1 test accuracy for InceptionV3 – various learning rate configurations

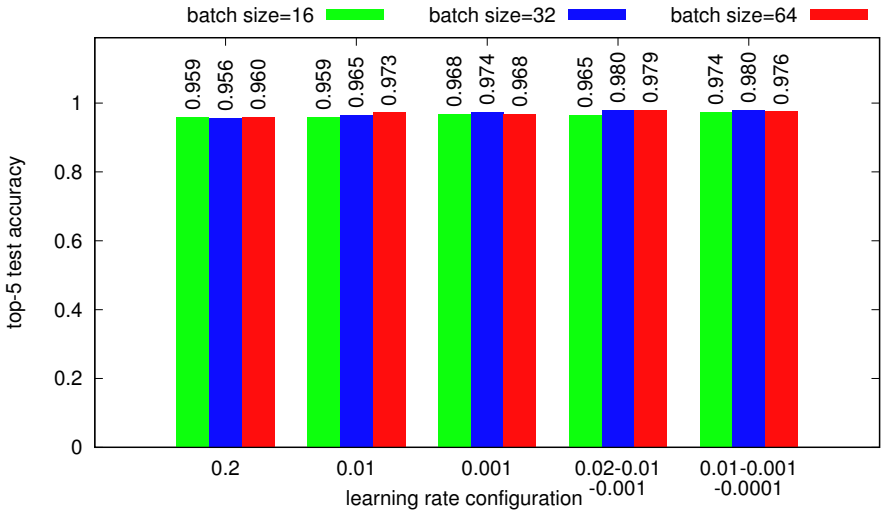

Fig. 3. top-5 test accuracy for InceptionV3 – various learning rate configurations

than 0.02 and the learning rate of 0.001 marginally higher than for 0.01. On the other hand, even better results have been possible with decreasing the learning rate i.e. 0.02-0.01-0.001 gives even better accuracy and marginally best out of the tested sets was obtained by 0.01-0.001-0.0001 for top-1 and very similar ones for the last two configurations for top-5 accuracy, which is very high.

2. In terms of batch size, for the best tested learning rate configuration (0.01-0.001-0.0001), best performance of the model was obtained for batch size 32 with the top-1 accuracy of 0.89, followed by 0.884 for batch size 64 after 25 epochs. top-1 accuracy for training and test as well as corresponding losses for batch size 32 are shown in Figure 4. Corresponding top-1 and top-5 accuracies obtained in various epochs are shown in Figure 5.

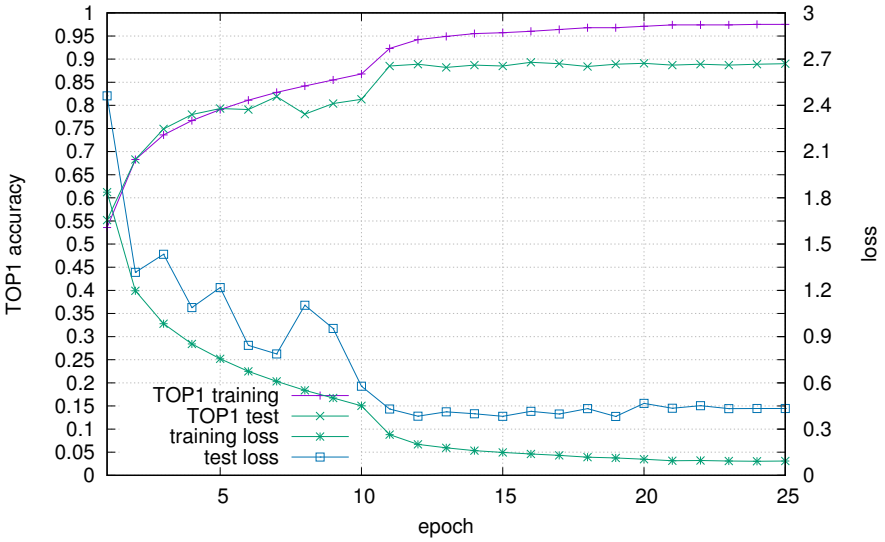

Fig. 4. top-1 accuracy and loss (training and test) vs epoch for InceptionV3 – learning rate=0.01-0.001-0.0001, batch size=32

Batch size is limited due to GPU memory size. We have tested configurations fitting into the given GPU memory size. Our results and observing increasing accuracies up to the tested batch size of 32 are similar to those presented in [14] where, for MNIST and CIFAR-10 and batch sizes 16, 32, 50, 64, 100 and 128 increasing accuracies are observed up to the batch size of 100 and a drop for 128. Furthermore, authors of paper [21] tested various learning rates for batch sizes of 32 and 64 for training a LeNet network for detecting exudate in eye fundus images, achieving visibly better results for batch size 64 and learning rate 0.01. Within this paper we tested even variable learning rates, compared to that approach. Results seen in the charts in this paper are also in line with top-1 accuracy versus batch size for a fixed learning rate of 0.01 shown in [16] with increasing values from 32 through 64, 128 and 256 and visible drop for 512 and 1024 – not observed here due to the fact that such large values were not possible to be tested. Authors of [8] concluded that for CNNs, for larger learning rates larger batch sizes perform better and they recommend small batch sizes for

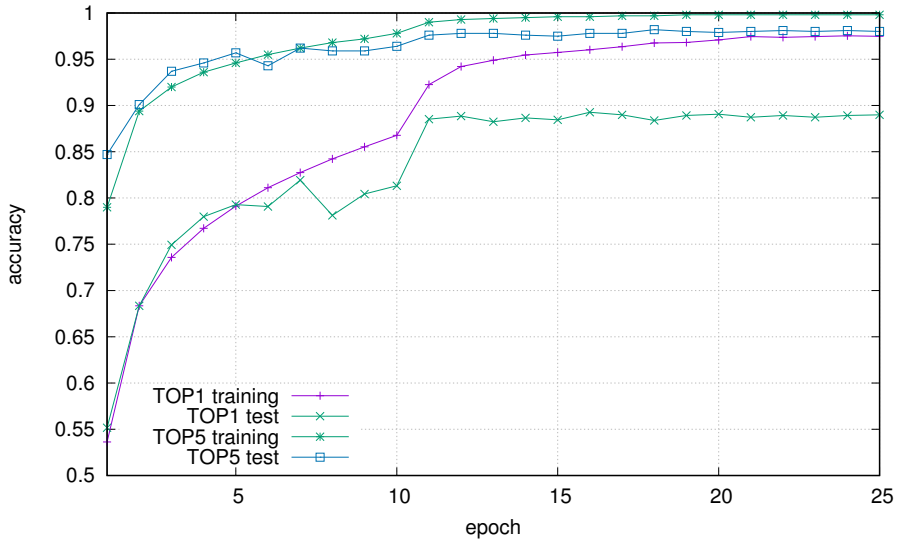

Fig. 5. top-1 and top-5 accuracy (training and test) vs epoch for InceptionV3 – learning rate=0.01-0.001-0.0001, batch size=32

smaller learning rates. In the case of our experiments with fixed learning rate, for 0.01 best accuracy was obtained for batch size 64 out of 16, 32 and 64 while for learning rate 0.001 better results were obtained for batch sizes 32 and 16, compared to 64.

Finally, we present model training times for the best learning rate configuration in Figure 6, showing a considerable reduction of times from batch size 8 to 16 and smaller for 32 and 64. In general, training performance for selected neural networks depends on both batch sizes as well as architectural advancements such as the I/O subsystem, as reported by [22]. Accuracy/training time is best for batch size 64.

6 Conclusions and future work

In the paper, we presented a neural network based approach for classification of food categories from images, both for Food-101 as well as Food-101 extended with 31 additional Polish dishes. Training and comparison was initially performed for several models including MobileNet, MobileNetV2, ResNet50, ResNet50V2, ResNet101, ResNet101V2, InceptionV3, InceptionResNetV2, Xception, NasNetMobile and DenseNet, using both NVIDIA Tesla V100 and GTX 1060 GPUs. Then we analyzed in detail the model giving best results – InceptionV3 and performed detailed assessment of model performance and training times for various learning rate configurations (both constant and variable) and various batch sizes finding the best (out of the tested ones) variable learning rate

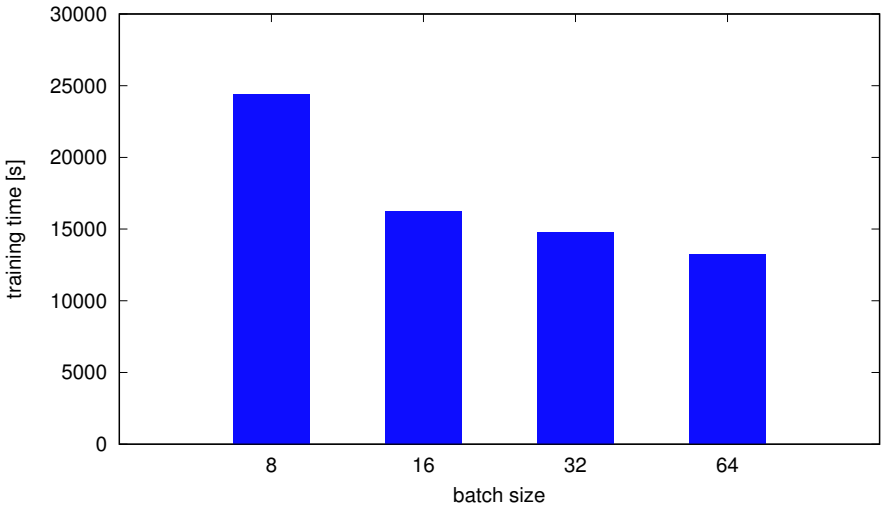

Fig. 6. Training times for best tested learning rate=0.01-0.001-0.0001

configuration 0.01-0.001-0.0001 and batch size 32. Finally, we presented comparison of inference times for Intel i7-7700 CPU and NVIDIA GTX 1060 GPU that are typical of desktop systems used by end users nowadays.

Future work will cover the following areas: incorporation of energy measurements into assessment of performance-energy trade-offs such as presented in [11] as well as extending focus on deployment and inference time measurements for mobile devices.

Acknowledgment

In this work, we used facilities located at the Faculty of Electronics, Telecommunications and Informatics – especially the DGX Station with NVIDIA V100 cards as well as GTX 1060 cards located at the lab of the Department of Computer Architecture of the aforementioned faculty.

References

1. Bossard, L., Guillaumin, M., Van Gool, L.: Food-101 – mining discriminative components with random forests. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. pp. 446–461. Springer International Publishing, Cham (2014)
2. Ciocca, G., Napoletano, P., Schettini, R.: Food recognition: a new dataset, experiments and results. IEEE Journal of Biomedical and Health Informatics **21**(3), 588–598 (2017). https://doi.org/10.1109/JBHI.2016.2636441

3. Demirkaya, A., Chen, J., Oymak, S.: Exploring the role of loss functions in multiclass classification. In: 2020 54th Annual Conference on Information Sciences and Systems (CISS). pp. 1–5 (2020). https://doi.org/10.1109/CISS48834.2020.1570627167
4. Gungor, C., Baltaci, F., Erdem, A., Erdem, E.: Turkish cuisine: A benchmark dataset with turkish meals for food recognition. In: Signal Processing and Communications Applications Conference (SIU) 2017. pp. 1–4. IEEE (2017)
5. Hassannejad, H., Matrella, G., Ciampolini, P., De Munari, I., Mordonini, M., Cagnoni, S.: Food image recognition using very deep convolutional networks. In: Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management. p. 41–49. MADiMa '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2986035.2986042, https://doi.org/10.1145/2986035.2986042
6. Joutou, T., Yanai, K.: A food image recognition system with multiple kernel learning. In: ICIP. pp. 285–288. IEEE (2009), http://dblp.uni-trier.de/db/conf/icip/icip2009.html#JoutouY09
7. Kagaya, H., Aizawa, K., Ogawa, M.: Food detection and recognition using convolutional neural network. In: Proceedings of the 22nd ACM International Conference on Multimedia. p. 1085–1088. MM '14, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2647868.2654970, https://doi.org/10.1145/2647868.2654970
8. Kandel, I., Castelli, M.: The effect of batch size on the generalizability of the convolutional neural networks on a histopathology dataset. ICT Express **6**(4), 312–315 (2020). https://doi.org/https://doi.org/10.1016/j.icte.2020.04.010, https://www.sciencedirect.com/science/article/pii/S2405959519303455
9. Kawano, Y., Yanai, K.: Food image recognition with deep convolutional features. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication. p. 589–593. UbiComp '14 Adjunct, Association for Computing Machinery, New York, NY, USA (2014). https://doi.org/10.1145/2638728.2641339, https://doi.org/10.1145/2638728.2641339
10. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (May 2017). https://doi.org/10.1145/3065386, https://doi.org/10.1145/3065386
11. Krzywaniak, A., Czarnul, P., Proficz, J.: Extended investigation of performance-energy trade-offs under power capping in hpc environments. In: 2019 International Conference on High Performance Computing Simulation (HPCS). pp. 440–447 (2019). https://doi.org/10.1109/HPCS48598.2019.9188149
12. Liu, C., Cao, Y., Luo, Y., Chen, G., Vokkarane, V., Ma, Y.: Deepfood: Deep learning-based food image recognition for computer-aided dietary assessment. In: Chang, C.K., Chiari, L., Cao, Y., Jin, H., Mokhtari, M., Aloulou, H. (eds.) Inclusive Smart Cities and Digital Health. pp. 37–48. Springer International Publishing, Cham (2016)
13. Lu, Y.: Food image recognition by using convolutional neural networks (cnns) (2016)
14. M., R.P.: Impact of Training Set Batch Size on the Performance of Convolutional Neural Networks for Diverse Datasets. Information Technology and Management Science **20**(1), 20–24 (December 2017), https://ideas.repec.org/a/vrs/itmasc/v20y2017i1p20-24n3.html

15. Mezgec, S., Koroušić Seljak, B.: Nutrinet: A deep learning food and drink image recognition system for dietary assessment. Nutrients **9**(7) (2017). https://doi.org/10.3390/nu9070657, https://www.mdpi.com/2072-6643/9/7/657
16. Mishkin, D., Sergievskiy, N., Matas, J.: Systematic evaluation of CNN advances on the imagenet. CoRR **abs/1606.02228** (2016), http://arxiv.org/abs/1606.02228
17. Nguyen, B.T., Dang-Nguyen, D., Tien, D.X., Phat, T.V., Gurrin, C.: A deep learning based food recognition system for lifelog images. In: De Marsico, M., di Baja, G.S., Fred, A.L.N. (eds.) Proceedings of the 7th International Conference on Pattern Recognition Applications and Methods, ICPRAM 2018, Funchal, Madeira - Portugal, January 16-18, 2018. pp. 657–664. SciTePress (2018). https://doi.org/10.5220/0006749006570664, https://doi.org/10.5220/0006749006570664
18. Pan, L., Li, C., Pouyanfar, S., Chen, R., Zhou, Y.: A novel combinational convolutional neural network for automatic food-ingredient classification. Computers, Materials & Continua **62**(2), 731–746 (2020). https://doi.org/10.32604/cmc.2020.06508, http://www.techscience.com/cmc/v62n2/38273
19. Pandey, P., Deepthi, A., Mandal, B., Puhan, N.B.: Foodnet: Recognizing foods using ensemble of deep networks. CoRR **abs/1709.09429** (2017), http://arxiv.org/abs/1709.09429
20. Park, S.J., Palvanov, A., Lee, C.H., Jeong, N., Cho, Y.I., Lee, H.J.: The development of food image detection and recognition model of korean food for mobile dietary management. Nutrition research and practice **13**(6), 521–528 (2019), https://doi.org/10.4162/nrp.2019.13.6.521
21. Perdomo, O., Arevalo, J., González, F.A.: Convolutional network to detect exudates in eye fundus images of diabetic subjects. In: Romero, E., Lepore, N., Brieva, J., Brieva, J., and, I.L. (eds.) 12th International Symposium on Medical Information Processing and Analysis. vol. 10160, pp. 235 – 240. International Society for Optics and Photonics, SPIE (2017). https://doi.org/10.1117/12.2256939, https://doi.org/10.1117/12.2256939
22. Rościszewski, P., Iwański, M., Czarnul, P.: The impact of the ac922 architecture on performance of deep neural network training. In: 2019 International Conference on High Performance Computing Simulation (HPCS). pp. 666–673 (2019). https://doi.org/10.1109/HPCS48598.2019.9188164
23. Singla, A., Yuan, L., Ebrahimi, T.: Food/non-food image classification and food categorization using pre-trained googlenet model. In: Proceedings of the 2nd International Workshop on Multimedia Assisted Dietary Management. p. 3–11. MADiMa '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2986035.2986039, https://doi.org/10.1145/2986035.2986039
24. Yanai, K., Kawano, Y.: Food image recognition using deep convolutional network with pre-training and fine-tuning. In: 2015 IEEE International Conference on Multimedia Expo Workshops (ICMEW). pp. 1–6 (June 2015). https://doi.org/10.1109/ICMEW.2015.7169816
25. Zhang, W., Zhao, D., Gong, W., Li, Z., Lu, Q., Yang, S.: Food image recognition with convolutional neural networks. In: 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom). pp. 690–693 (2015)