

Application of a Scaled MNIST Dataset Blended with Natural Scene Background on ResNet

Marinov, Alexander; Mtetwa, Nhamo; Larijani, Hadi

Published in:

ICBDE'19: Proceedings of the 2019 International Conference on Big Data and Education

DOI:

[10.1145/3322134.3322147](https://doi.org/10.1145/3322134.3322147)

Publication date:

2019

Document Version

Author accepted manuscript

[Link to publication in ResearchOnline](#)

Citation for published version (Harvard):

Marinov, A, Mtetwa, N & Larijani, H 2019, Application of a Scaled MNIST Dataset Blended with Natural Scene Background on ResNet. in *ICBDE'19: Proceedings of the 2019 International Conference on Big Data and Education*. ACM International Conference Proceeding Series, Association for Computing Machinery (ACM), New York, pp. 77-81, 2019 International Conference on Big Data and Education, London, United Kingdom, 30/03/19. <https://doi.org/10.1145/3322134.3322147>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

If you believe that this document breaches copyright please view our takedown policy at <https://edshare.gcu.ac.uk/id/eprint/5179> for details of how to contact us.

Application of a Scaled MNIST Dataset Blended with Natural Scene Background on ResNet

Alexander Marinov

Dept. Cyber Security and Networks
Glasgow Caledonian University
Glasgow, G4 0BA, U.K.
(0044)-203-289-2332

Alexander.Marinov@gcu.ac.uk

Nhamo Mtetwa

Dept. Cyber Security and Networks
Glasgow Caledonian University
Glasgow, G4 0BA, U.K.
(0044)-141-331-3190

Nhamoinesu.Mtetwa@gcu.ac.uk

Hadi Larijani

Dept. Cyber Security and Networks
Glasgow Caledonian University
Glasgow, G4 0BA, U.K.
(0044)-141-331-3190

H.Larijani@gcu.ac.uk

k

ABSTRACT

Deep learning (DL) has gained a lot of popularity in the science and business community. It has been successful in a range of applications, especially in computer vision. This paper presents results from applying scaled MNIST images dataset to a popular implementation of deep learning called ResNet. This is a valuable contribution because in general convolutional networks are not scale invariant. Our objective is to explore the behavior of a residual neural network when trained and evaluated using three different datasets of scaled MNIST images.

CCS Concepts

• Computing methodologies~Neural networks • Computing methodologies~Batch learning

Keywords

feature cnn; machine learning; scaling; mnist; resnet; deep learning

1. INTRODUCTION

Machine learning (ML) is progressing at a fast rate, but one branch of ML, deep learning (DL) has attracted a lot of attention. DL is behind many improvements in computer vision tasks where analysis, understanding and automatic extraction of useful information is performed on images. For example, tasks like:

- image classification – where the classification depends on the statistical probability of the object presence in the image and then can be assigned to a class [1],
- object detection [2] – where a list of coordinates is generated for the object, class label is assigned with an associated probability score.
- scene recognition [3] [4] – a process where knowledge of the objects and the scene is required.
- image restoration [5] – the process allows an image restoration from a low-level resolution to provide better quality of the image.
- semantic segmentation [6] – here the task is to describe regions and structures on the input image. With semantic segmentation an attempt is made to recognize and understand what has been provided via the input at pixel level.

and many more.

Despite all these successes one aspect that is still lacking is the application of deep learning to scaled images. Attempts to address this issue by using variations of data augmentation have been made

by scaling the original data and pushing the data via the training process so the network could learn more variation of the same scaled data [7]. Although the results from applying data augmentation have provided improvement [8], the down side is that it has made the training process longer.

The main contributions of this research are:

- A novel way of creating a new dataset by scaling the MNIST dataset objects and blending it with a specific size of a natural scene background image.
- The second contribution is providing a ResNet-50 based network implementation in keras, with 10 output classes.

The rest of the paper is organized as follows: Section II presents literature review. The description of our implementation pipeline is in section III. The findings from the experiment are in section IV. And the paper concludes in section V.

2. LITERATURE REVIEW

During the rise of deep learning there has been a lot of different experiments which vary based on dataset quantity and quality [9] [10] [11], type of the dataset e.g. natural scene, text, objects etc. [9] [11], type of the deep learning network architecture e.g. convolutional, recurrent etc. [12] [13], initial data preparation and data massaging [14] and the feature extraction approach [15] [16].

2.1 Mnist Dataset

The MNIST database [17] contains a large amount of handwritten decimal digits in different forms. The images in MNIST dataset are normalized and fit into 28x28 pixels bounding box. It contains 70,000 different images which allows the data to be split into three groups as follows 50,000 for training, 10,000 for testing and 10,000 for evaluation.

2.2 Scaling Images

Scaling depends on the type of image which must be scaled. When the image is vectorized the process could use geometric transformation of the graphic primitives with no loss of quality. The second type of images are raster where a new image has to be generated with different numbers of pixels (increase when scaling up and decrease when scaling down).

2.3 Applying Background

Applying a blending algorithm on two images determines how the two layers of a new image will blend together. Usually the default technique is to hide the lower level layer and display only the top layer as a new image. There are many blending approaches: Normal blend mode, Dissolve etc. [18].

2.4 Dataset Formating

Datasets could be stored in two methods: in compressed / non-compressed packages or in a file structure. The package method allows easiness of transfer and it is suitable for small to medium sized datasets. The file structured dataset allows one to store large quantities of data. Another benefit of file stored dataset is that it is easier to load the data on request rather than load the whole package in memory.

2.5 Training

Training is a crucial step in the neural network learning process. It usually consists of optimising the loss function. There have been many attempts to optimise and minimise the loss function. The loss function is a composition of an error and regularization terms. The main purpose of learning is to search for a parameter vector W at which the loss function takes a minimum value. As the loss function is a non-linear function the consequence is that it is difficult to find a training algorithm for achieving minimum value. Some of the algorithms used to find the minimum value of the loss function are: Gradient descent [19], Newton's method [20], Conjugate gradient [21], Quasi Newton [22], Levenberg Marquardt [23].

The choice of which algorithm to use for training depends among other things on the neural network (NN) parameters and instances. In general, the choice of which algorithm to use depends on the size of the NN and the hardware. For NNs with thousands of parameters one of the gradient algorithms is preferred as they consume less memory during training. For NNs with few instances and few hundred parameters applying Levenberg-Marquardt algorithm is most suitable. For any other configurations, one of the Newton's algorithms is most suited.

3. METHOD

The methodology followed in this work is summarised in Figure 1.

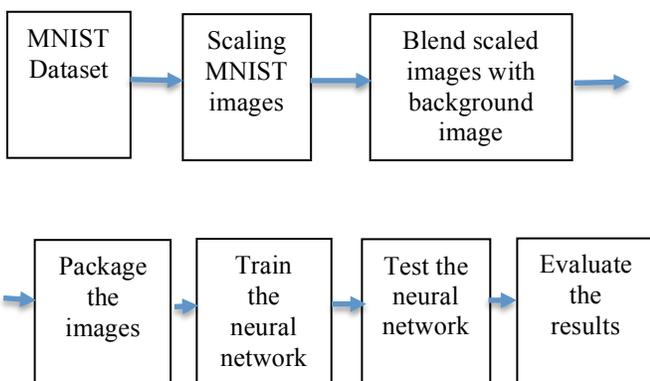


Figure 1. Summarized workflow.

For these experiments we will use the residual approach for the deep learning network (DLN) created by the Microsoft team which allowed DLN to go deeper and extract more features from the input [24]. We will use the implementation provided by the keras framework of ResNet-50.

3.1 Data Preparation

For the purpose of the experiment we are using as base the MNIST dataset images. Three datasets based on three upscaling of the MNIST images and an image background of size 224x224 pixels are prepared. To provide the scaling we use the OpenCV library and its functionality to scale images. To scale the MNIST images we use an extension of cubic interpolation - bicubic interpolation [25] as this process interpolates data points on a two dimensional grid.

The upscaled image and a random natural scene image are blended using OpenCV library. The scaled image is used as a top layer and the other image is used as a background. For the purpose of the experiment a normal blending with 0.5 opacity is applied as shown on figure 2. The list of image manipulation actions is in Table 1.



Figure 1. Overview of the image preparation.

Table 1: List of changes to the images

Action	Explanation
Preparation of MNIST	Extracting the 60000 images and separating them to 50000 images for training and 10000 images for validation. Keeping the original 10000 images for evaluation.
Scale MNIST double	Scaling MNIST images from 28 to 56 pixels – 100%
Scale MNIST 4 times	Scaling MNIST images from 28 to 112 pixels – 200%
Scale MNIST 8 times	Scaling MNIST images from 28 to 224 pixels – 400%
Applying background	Applying background to all three scaled data sets by blending the scaled images and a natural scene image with size 224x224 as a background, and the scaled MNIST instance as a top layer.

For brevity the following naming convention will be adopted:

- The dataset with 100% scaled MNIST images will be called scaledMNIST100pct and the rest as follow

scaledMNIST200pct for 200% scaled MNIST and scaledMNIST400pct for 400% ones.

Algorithm for dataset preparation of scaledMNIST100pct:

1. Extract 50000 images from MNIST dataset out of the 60000 images for training.
2. Apply 100% bicubic scale
3. Blend the image with the background using normal blending
4. Pack the images in an array for training
5. Use remaining 10000 images from MNIST dataset out of the 60000 images for training not used in the step 1.
6. Repeat steps 2-3
7. Pack the images in an array for validation
8. Extract 10000 images from MNIST from the 10000 validation images
9. Repeat steps 2-3
10. Pack the images in array for testing.

The same algorithm is applied for the other two datasets with the relevant scaling of 200% and 400%.

3.2 Training and Evaluation Selection

A Python program has been developed based on the keras framework. The program provides a neural network which will be used to perform training on the scaled images datasets. It extracts the relevant results from the training process and evaluates the trained network with the evaluation data. A readymade template of ResNet-50 is used as an initial neural network which is then customized. In most cases this network is used pretrained with the ImageNet dataset, but for this experiment the scaled MNIST dataset will be used to train the network from scratch. The template ResNet-50 outputs 1000 classes, but our dataset expects only 10 outputs. As shown on figure 2 in order to achieve the desired 10 output classes, we flatten the original output, applied a dropout layer and softmax layer for producing 10 outputs.

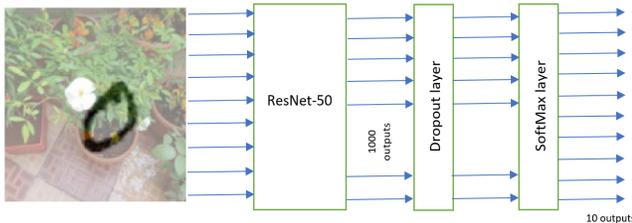


Figure 2. Extended ResNet-50 architecture.

The configuration of the network is as follows:

- all the weights are randomly initialized,
- input shape is adjusted as per image size as follow 224x224x3
- as a learning optimizer Adadelta [26] is used
- the training is configured to 12 epochs with 70 batches. This restriction is purely hardware dependent.

After completing the training of the neural network an evaluation is performed on the testing data which the network has not seen during training.

4. RESULTS

Running the experiment, first training is performed separately for the three data sets. In figures 2 and 3 is shown the results of each epoch's validation metrics for the three different trainings. Figure 2 shows the validation accuracy, and figure 3 shows the validation error during training for the three datasets.

The results show that training with different scaled images yields different results in accuracy and error across the different scaling factors. From figures 2 and 3 it is clear, that the images scaled by 200% has the most accurate score and the least error. The 100% scaled images accuracy and error measures are wavy. The 400% scaled images results are somewhere in the middle.

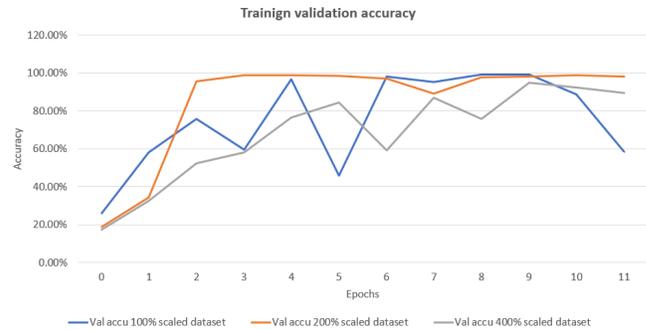


Figure 3. Validation accuracy from the three scaled images dataset training.

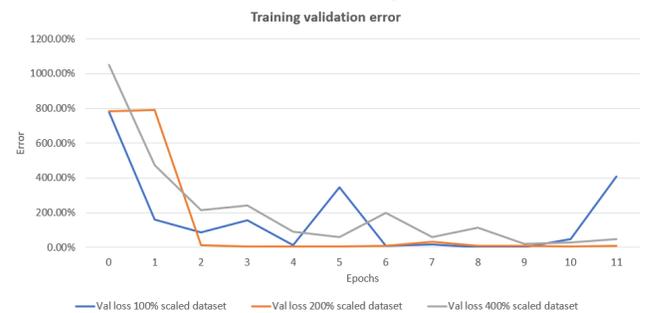


Figure 4. Validation error/loss from the three scaled images dataset training.

After the training has completed, an evaluation is performed per dataset. The evaluation is performed on the 10,000 images which have not been used during training. Table 2 and figure 4 show the accuracy during evaluation for the three datasets, and table 3 and figure 5 show the error during evaluation.

The results from evaluation are similar in comparison to the training results in that the 200% scaled images performed at the best and the 100% scaled images performed at least.

Table 2: accuracy on evaluation of the trained networks

Acc 100% scaled dataset	Acc 200% scaled dataset	Acc 400% scaled dataset
59.91%	98.32%	90.53%

Table 3: loss on evaluation of the trained networks

Loss 100% scaled dataset	Loss 200% scaled dataset	Loss 400% scaled dataset
396.19%	8.10%	45.08%

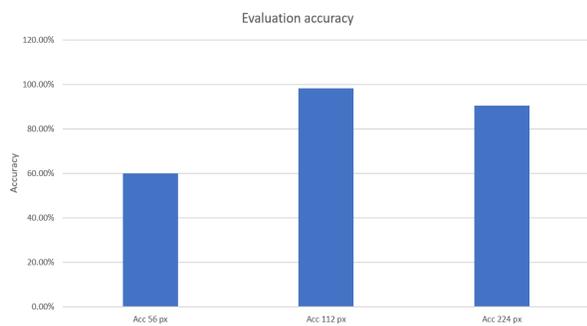


Figure 5. Validation accuracy from the three scaled images dataset evaluation.

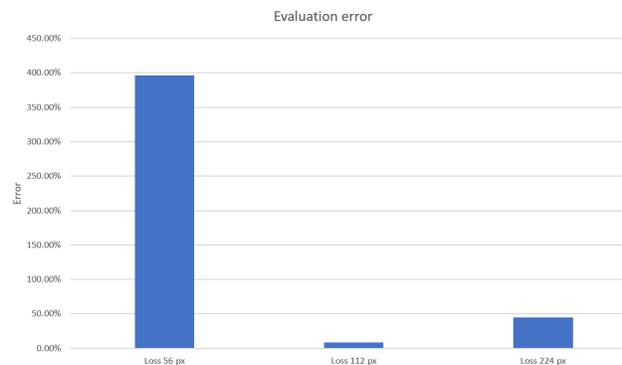


Figure 6. Validation error/loss from the three scaled images dataset evaluation.

5. CONCLUSION

In conclusion, training and evaluating a CNN with different scaled images yields different results. There have been very promising results as shown in the 200% scaled images and very poor results in the 100% scaled images. For us this shows that the CNN performs better at a certain scale and inability of the network to generalize the provided information in other scales. For future work we hope to simulate across more finer grain scaling for the images to discover the optimal proportion of the object to image.

6. ACKNOWLEDGMENTS

Our thanks to MLCLabs and MGCA Ltd to provide their laboratories for our experiments.

7. REFERENCES

- [1] M. S., Large scale object detection., Department of Cybernetics Faculty of Electrical Engineering, Czech Technical University, 2014.
- [2] S. Ren, K. He, R. Girshick and J. Sun, ""Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,"" *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, 2017.
- [3] Bolei Zhou, Agata Lapedriza, Jianxiang Xiao, Antonio Torralba and Aude Oliva, "Learning Deep Features for Scene Recognition using Places Database,"DOI= http://places.csail.mit.edu/places_NIPS14.pdf.
- [4] Luis Herranz, Shuqiang Jiang and Xiangyang Li, "Scene recognition with CNNs: objects, scales and dataset bias,"(21 January 2018). DOI= <https://arxiv.org/pdf/1801.06867.pdf>.
- [5] Kai Zhang, Wangmeng Zuo, Shuhang Gu and Lei Zhang, "Learning Deep CNN Denoiser Prior for Image Restoration," (11 April 2017). DOI= <https://arxiv.org/pdf/1704.03264.pdf>.
- [6] Jonathan Long, Evan Shelhamer, Trevor Darrell, "Fully Convolutional Networks for Semantic Segmentation," DOI= https://people.eecs.berkeley.edu/~jonlong/long_shelhamer_fcnn.pdf.
- [7] L. Yaeger, R. Lyon, and B. Webb., "Effective training of a neural network character classifier for word recognition.," in *In Advances in Neural Information Processing Systems*, 1996, pp. 807-813.
- [8] Luke Taylor, Geoff Nitschke, "Improving Deep Learning using Generic Data Augmentation," 20 August 2017. DOI= <https://arxiv.org/pdf/1708.06020.pdf>.
- [9] Kaiming He, Georgia Gkioxari, Piotr Dollar, Ross Girshick, "Mask R-CNN," 24 January 2018. DOI= <https://arxiv.org/pdf/1703.06870.pdf>.
- [10] Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, Kaiming He, "Aggregated Residual Transformations for Deep Neural Networks," 11 April 2017. DOI=<https://arxiv.org/pdf/1611.05431.pdf>.
- [11] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Ken Nakae, Shin Ishii, "DISTRIBUTIONAL SMOOTHING WITH VIRTUAL ADVERSARIAL TRAINING," 11 June 2016. DOI= <https://arxiv.org/pdf/1507.00677.pdf>.
- [12] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," 2012. DOI=<http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf>.
- [13] Toma's Mikolov, Martin Karafiat, Luka's Burget, Jan "Honza" Cernock, Sanjeev Khudanpur, "Recurrent neural network based language model," 2010. DOI= https://www.isca-speech.org/archive/archive_papers/interspeech_2010/i10_1045.pdf.
- [14] Lean Yu, Shouyang Wang and K. K. Lai., ""An integrated data preparation scheme for neural network data analysis,"" *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 2, pp. 217-230, 2006.
- [15] Jianchang Mao and A. K. Jain, ""Artificial neural networks for feature extraction and multivariate data projection,"" *IEEE Transactions on Neural Networks*, vol. 6, no. 2, pp. 296-317, 1995.
- [16] Setiono R., Liu H., "Feature extraction via Neural networks.," in *Feature Extraction, Construction and Selection.*, Boston, Springer, 1998, pp. 191-204.
- [17] Yann LeCun, Corinna Cortes, Christopher J.C. Burges, "THE MNIST DATABASE of handwritten digits," [Online]. Available: <http://yann.lecun.com/exdb/mnist/>.
- [18] Rik Cabanier, Nikos Andronikos, "Compositing and Blending Level 1," W3C, 13 January 2015. DOI=<https://www.w3.org/TR/compositing/>.
- [19] N. Cui, "Applying Gradient Descent in Convolutional Neural," 2018. DOI= <http://iopscience.iop.org/article/10.1088/1742->

6596/1004/1/012027/pdf.

- [20] Chien-Chih Wang, Kent Loong Tan, Chun-Ting Chen, Yu-Hsiang Lin, S. Sathya Keerthi, Dhruv Mahajan, S. Sundararajan, Chih-Jen Lin, "Distributed Newton Methods for Deep Neural Networks," 1 February 2018. DOI=<https://arxiv.org/pdf/1802.00130.pdf>.
- [21] Ioannis E. Livieris, Panagiotis Pintelas, "A new conjugate gradient algorithm for training neural networks based on a modified secant equation," *Applied Mathematics and Computation*, vol. 221, pp. 491-503, 2013.
- [22] B. ROBITAILLE, B. MARCOS, M. VEILLETTE and G. PAYRE , "MODIFIED QUASI-NEWTON METHODS FOR TRAINING NEURAL NETWORKS," *Computers chem.*, vol. 20, no. 9, pp. 1133-1140, 1993.
- [23] Hao Yu, Bogdan M. Wilamowski, "Levenberg–Marquardt Training," 9 3 2010. DOI=http://www.eng.auburn.edu/~wilambm/pap/2011/K10149_C012.pdf.
- [24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Deep Residual Learning for Image Recognition," 10 December 2015. DOI=<https://arxiv.org/pdf/1512.03385.pdf>.
- [25] Prachi R Rajarapollu, Vijay R Mankar, "Bicubic Interpolation Algorithm Implementation," *IJCST*, vol. 8, no. 2, pp. 23-26, 2017.
- [26] M. D. Zeiler, "ADADELTA: AN ADAPTIVE LEARNING RATE METHOD," 22 December 2012. DOI= <https://arxiv.org/pdf/1212.5701.pdf>.