Handwritten Digits Recognition Report

Abstract—**This work attempts to investigate the performance of different deep-learning approaches in a multiclass machine vision problem. Models were trained on an adapted version of the Modified National Institute of Standards and Technology (MNIST) database with the purpose of predicting the maximum of three handwritten digits in a single image. This research mainly contrasts two different approaches, using preprocessed data as well as raw image data to train several deep learning models. Further this study investigates the performance of different well established models, such as VGGNet and ResNet. In summary, we could achieve an accuracy of 96,13 % with a ResNet18 and showed that extensive preprocessing is not necessarily required for these models.**

I. Introduction

The constantly increasing number of digital images, provided in online communities, as well as the increasing need of process automation in industry leads to an increasing need and importance of machine vision systems. The classification of handwritten digits can be important in automatic document processing, and mainly contributes to the development of human-computer interfaces. This work contrasts two different approaches to identify the maximum if three handwritten digits in one image and finally compares the performance of different well established models. In the first approach, we trained a convolutional neural network on the original MNIST dataset [1] and preprocessed the provided dataset (see Fig.1) to correspond to the MNIST data style. We then predicted each number individually and numerically identified the maximum. However, the graphical circumstances, gradual backgrounds and background noise led to errors during the digit identification and thus disturbed the models predictions. This leads us to our second approach. Instead of preprocessing, we only normalized the provided dataset to train several well established models, such as VGG Net and ResNet. In summary, these models turned out to be very efficient in the identification of the important image parts (i.e. the largest digit) without any preprocessing and partitioning. Based on this, we trained different models and identified ResNet18 to be the most efficient and accurate in this problem.

II. Related Work

Since the Modified National Institute of Standards and Technology (MNIST) dataset has been created in 1998 [2], it became a widely used dataset and has been used to compare a huge variety of different learning models performance on handwritten digit recognition. Different supervised learning models, such as K-nearest neighbors [3], Support Vector Machine [4] , and Neural Networks [5] were able to achieve over 99% test accuracy. However, the competition is now led and dominated by Convolutional Neural Networks. Convolutional Neural Networks have been widely used and succeed in large-scale image, as well as video recognition [5]. In 2012, Ciresan et al. [6] combined several deep neural networks to a multi-column deep convolutional neural network and achieved the, until now, the highest test accuracy of 99,77%.

VGGNet [7] and **ResNet** [8] are two famous networks which have optimal performance on image classification, due to their deep architecture. We decided to apply these two models in our second approach and choose the one performs better after several tests as our final model.

1. **Very Deep Convolutional Networks(VGG)** was invented within the Visual Geometry Group [7], [9] and won the localization task in the ILSVRC in 2014. VGGs architecture mainly consists of a relatively high number of convolutional layers (dependent on the Version of VGG, i.e. VGG-16 with 16 convolutional layers or VGG-19 with 19). VGG uses max pooling between the convolutional layers and a ReLU activation function for all hidden layers [7]. By using multiple small filters(3x3) instead of single large filter(eg. 7x7), it obtains a large receptive field, increases the depth of the network and reduces the number of parameters. This helps the model to converge faster and reduces the risk of overfitting. Thus, we decided to use an adapted version of VGGNet to solve this similar image recognition problem.

2. **The Residual Networks(ResNet)** won the the ILSVRC in 2015 and was published in 2016 by He et al. [8]. The ResNet provides a unique and very deep architecture that can contain up to 125 layers. Usual neural networks of this depth faced problems like vanishing or exploding gradients. To account for this problem, the ResNet core feature are so called shortcut- or skip connection. Instead of using a stack of layers to directly fit a desired underlying mapping, the author introduced a residual mapping for fitting, using these skip connections. The new mapping allows short connections to skip one or more layers [8]. Using this strategy, the deep residual nets support a very deep architecture, become easy to optimize and gain accuracy even the network depth is large.

III. Dataset

In this project, an adapted version of the MNIST dataset was provided. In this section we briefly describe the original, as well as the adapted dataset, as both were used in this project.The original MNIST dataset contains of 60,000 training examples, and 10,000 test examples of handwritten 28x28 pixel binary images. Each training image contains a single digit in the range from 0 to 9 and is labeled respectively.

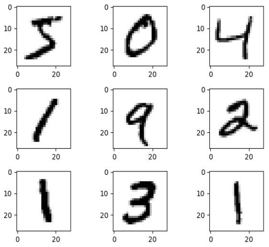


Fig. 1 Examples of the original MNIST dataset

In this particular project, an adapted version of the MNIST dataset, containing 50,000 training images and 10,000 testing images was provided. In contrast to the original dataset, each 128x128 pixel image in the modified version contains 3 digits as well as an individual grayscale background. The three digits per image, ranging from 0 to 9, are randomly rotated and replaced in each sample. Each training image is labeled with its maximal single-digit numeric value, forming 10 classes.

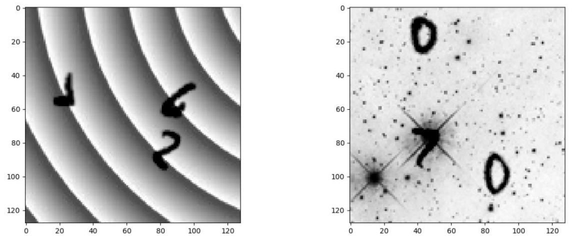


Fig. 2 Examples of the provided adapted MNIST dataset

IV. Proposed Approach

Data preprocessing

As the input data is an adapted version of the MNIST dataset, we have tried in our first approach to preprocess the dataset to transform each 128x128 pixel grayscale image into three (i.e. each input image contained three digits) 28x28 binary images. As all digits had an identical color value of 255 in the center , we used the OpenCV package to apply a threshold to eliminate 90-95% of the pixels. To further eliminate small pixel clusters and smoothen the image, we further applied a kernel of 3x3 pixels. After this step, digits were identified among other contours by size. The correspondent surrounding rectangle was then adapted to the standard MNIST size of 28x28 pixel.

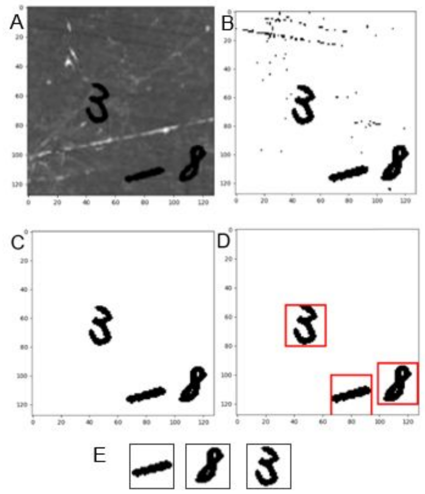


Fig. 3 Example of applied preprocessing and transformation steps

Selected models

As this project targets a machine vision problem, we have decided to work with different types of convolutional neural networks. In our first approach we have trained a convolutional neural network with 2 convolutional layers on the raw MNIST dataset. Our Model (shown in Figure4) contained four layers: two convolutional layers with a kernel size of 4 and stride size of 1, one dropout layer, and two linear layers (see Fig.4). All layers used the Rectifier (ReLu) activation function. The last two layers were fully connected layers, which again reduced the complexity and finally returned a value for the 10-class digit classification problem, using a Softmax activation function. The model was trained on the original MNIST dataset.

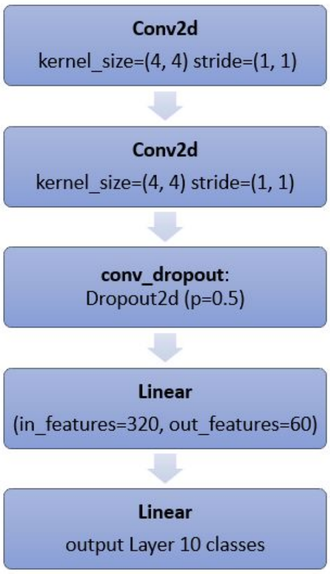

Conv2d
kernel_size=(4, 4) stride=(1, 1)
Conv2d
kernel_size=(4, 4) stride=(1, 1)
conv_dropout:
Dropout2d (p=0.5)
Linear
(in_features=320, out_features=60)
Linear
output Layer 10 classes

Fig. 4 Schema of the MNIST classifier

Second Approach

While we have tried to split our project into separate problems in the first approach (identify digits, label digits, select maximum), this approach uses complex pretrained models to process all problems at once. We decided for this project, to build on pretrained models, such as ResNet and VGG Net. On this approach, we were using the Package FastAI and Pytorch which provides a wide variety of pretrained models.

V. Results

First Approach

In the first approach we applied several preprocessing steps to the provided data to generate uniform pictures, containing one digit at a time (see Figure 3). We then trained the earlier described convolutional neural network on the original MNIST data. With this model, we were able to predict the original MNIST test set with an accuracy of 99%. However, due to the variable background and noise (as an example see Fig. 1, right), the preprocessed data contained misidentified digit areas and prevented therefore the model from accurate predictions. As the training set was not used for training, we have used is for cross validation. However, the convolutional neural network (see Fig. 4) could exceed 53% accuracy. As digits were turned randomly in the provided dataset, we fed the rotated single-digit images into the classifier and kept the prediction with the highest certainty. However, no improvement could be achieved. We thus decided to use more complex models, which do not need preprocessed data input. This leads to our second approach.

Second Approach

In the second approach, we have trained ResNet and VGG Net on the raw data. In particular, we mainly used ResNet18 and VGG16. Our best accuracy was achieved with a ResNet18. We also tried deeper recurrent network architectures, such as ResNet 34 (see Figure 5). However, this deeper architecture models tend to overfit after only a few epochs and mostly reached a maximal validation accuracy of 88% (f.e. see Fig.5). For the ResNet 18, the optimal test accuracy was achieved after 18 epochs. This solution was submitted on Kaggle and reached 96.13% test accuracy.The VGG net did not reach a higher accuracy in our approach. Table 1 summarizes the different models performance.

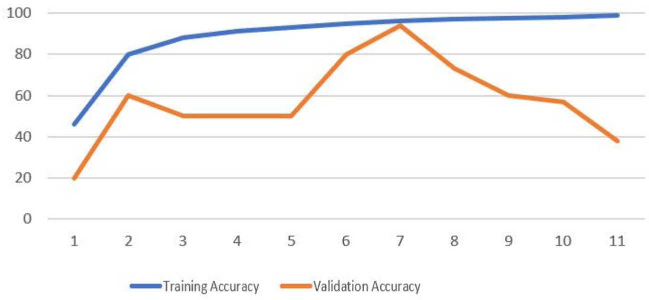
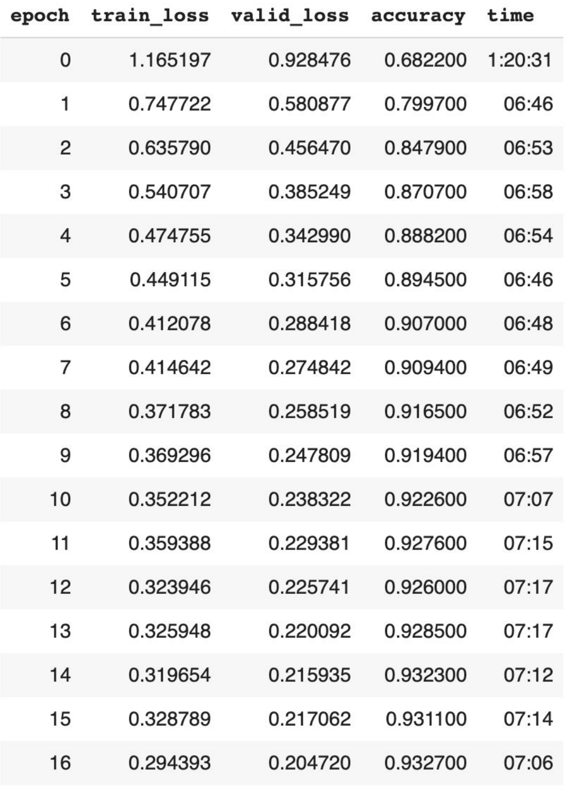


Fig. 5 The Training and Validation Accuracy of ResNet 34

Neural Network	Learning Rate	Validation Accuracy	Test Accuracy	Average Runtime per Epoch
VGG16	SGD	0.9526	0.9412	6.20 min
VGG16	0.0005	0.9327	0.9216	7.15 min
ResNet 18	0.001	0.9698	0.9613	5.20 min
ResNet 34	0.001	0.8764	-	7.30 min

Table. 1 Results of networks with learning rate, validation and test accuracy, and average runtime per epoch*

*In Table 1, VGG16 with SGD optimizer has its learning rate gradually changes according to the current performance. The predictions of ResNet 34 is not submitted to Kaggle to gain test accuracy because of its poor performance on validation data.



epoch	train_loss	valid_loss	accuracy	time
0	1.165197	0.928476	0.682200	1:20:31
1	0.747722	0.580877	0.799700	06:46
2	0.635790	0.456470	0.847900	06:53
3	0.540707	0.385249	0.870700	06:58
4	0.474755	0.342990	0.888200	06:54
5	0.449115	0.315756	0.894500	06:46
6	0.412078	0.288418	0.907000	06:48
7	0.414642	0.274842	0.909400	06:49
8	0.371783	0.258519	0.916500	06:52
9	0.369296	0.247809	0.919400	06:57
10	0.352212	0.238322	0.922600	07:07
11	0.359388	0.229381	0.927600	07:15
12	0.323946	0.225741	0.926000	07:17
13	0.325948	0.220092	0.928500	07:17
14	0.319654	0.215935	0.932300	07:12
15	0.328789	0.217062	0.931100	07:14
16	0.294393	0.204720	0.932700	07:06

Fig. 6 Example of VGG16 learning epochs*

* VGG16 with learning rate 0.0005 has its training loss and validation loss decreasing rapidly during the first 10 epochs. However, its accuracy encountered a divergence after the 11th epoch as the selected learning rate was too large. Without applying an adaptive optimization algorithm, its accuracy didn't improve significantly during the rest of the training progress and stabilized at 0.9327.

VI. Discussion and Conclusion

This work mainly contrasted two approaches on a machine vision problem with the aim to identify a maximal digit in an image containing three handwritten digits. The first approach attempted to separate this problem into subproblems: digit localization, digit classification and evaluation of the maximum. Only the second problem (digit classification) was processed by a convolutional neural network. However, even if the test accuracy on the original MNIST dataset was 99%, the performance for this particular task did not go beyond 53%. The reason for this, is mainly defined by the preprocessing errors and mallocalized digits, due to the presence of many tuning parameters, such as smoothing kernel size, binarization threshold and size threshold. A future approach could be to design a neural network as a feature detector to identify the digits and to feed the output into the existing convolutional neural network. This could be achieved by the creation of a new dataset of MNIST digits and identical shaped images of random noise.

In summary, image morphological operations like binarization , erosion and dilations do not perform well to extract digit regions on the provided dataset. Therefore, a simple convolutional neural network, trained on the MNIST dataset can not be used in this project. Thus, all three subproblems must be assigned to a single complex model. We have compared ResNet18 ResNet34 and VGG16. Overall, these models performed well as their validation accuracies are above 92% except ResNet 34. Our final Model, Resnet18, achieved 96.13% test accuracy on Kaggle.

VII. Statement of Contributions

All group members worked on all tasks together and contributed equally.

References

[1] "MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges." [Online]. Available: http://yann.lecun.com/exdb/mnist/. [Accessed: 13-Nov-2019].

[2] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[3] S. Belongie, J. Malik, and J. Puzicha, "Shape Matching and Object Recognition Using Shape Contexts," *IEEE Trans Pattern Anal Mach Intell*, vol. 24, no. 4, pp. 509–522, Apr. 2002.

[4] D. Decoste and B. Schölkopf, "Training Invariant Support Vector Machines," *Mach. Learn.*, vol. 46, no. 1, pp. 161–190, Jan. 2002.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, *ImageNet: A large-scale hierarchical image database*.

[6] D. Cireşan, U. Meier, and J. Schmidhuber, "Multi-column Deep Neural Networks for Image Classification," *ArXiv12022745 Cs*, Feb. 2012.

[7] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *ArXiv14091556 Cs*, Apr. 2015.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[9] "Visual Geometry Group - University of Oxford." [Online]. Available: http://www.robots.ox.ac.uk/~vgg/. [Accessed: 13-Nov-2019].