IRSTI 28.23.37

Optical character recognition with neural networks

A. Shalakhmetov, University of International Business,
Almaty, Republic of Kazakhstan, e-mail: aidar.shalakhmetov@gmail.com
S. Aubakirov, University of International Business,
Almaty, Republic of Kazakhstan, e-mail: aubakirov.sanzhar@gmail.com

XXI century is the age of global automation and digitization. There is high demand for optical recognition software, including character recognition. There are different approaches in solution optical recognition problem. Some of them based on classical feature extraction methods. Other based on machine learning algorithms. In this work, we observed related works in machine learning field and propose the plan for further research. The work relies on two research studies that describe basics and fundamentals of machine learning. These research include various experiments in this field. We conducted several experiments to get acquainted with methods and techniques and to identify key features that are affecting on optical character recognition process. We analyzed two main architectural structures: multilayer perceptron and convolutional neural network. In conclusion, we identified key points of machine learning techniques and composed our own strategy for further researches. The result shows the difference in performance of different convolutional neural network models under the same conditions. Further work will cover researches and experiments on performance of several architectures. In addition, we observed latest tools, software programs and environments for the most convenient way to organize implementation process.
Keywords: OCR, neural network, convolutional neural networks.

Оптическое распознавание символов с помощью нейронных сетей

Шалахметов А.З., Международный Университет Бизнеса,
г. Алматы, Республика Казахстан, e-mail: aidar.shalakhmetov@gmail.com
Аубакиров С.С., Международный Университет Бизнеса,
г. Алматы, Республика Казахстан, e-mail: aubakirov.sanzhar@gmail.com

XXI век – это век глобальной автоматизации и оцифровки данных. В наше время имеется огромный спрос на системы оптического распознавания, включая системы распознавания символов. В сфере оптического распознавания используются различные подходы в решении поставленных задач. Некоторые из них основываются на классических методах выделения характерных признаков. Некоторые базируются на алгоритмах машинного обучения. В данной работе рассматриваются исследования в сфере машинного обучения и предложения для последующих исследований. Данная статья основывается на двух публикациях, которые описывают основы машинного обучения. Мы поставили несколько аналогичных экспериментов для ознакомления с методами и техниками данного подхода, а также для определения основных принципов, которые влияют на процесс оптического распознавания. Мы проанализировали две основные архитектуры: многослойный перцептрон и сверточные нейронные сети. В заключении, мы ознакомились с основами алгоритмов машинного обучения и составили стратегию для дальнейших исследований. Полученный результат отражает разницу в производительности между разными моделями сверточных нейронных сетей при одинаковых условиях. Последующие работы будут содержать исследования и эксперименты различных архитектур. В дополнении, мы рассмотрели различные утилиты, программное обеспечение и среды для создания оптимального процесса реализации системы.
Ключевые слова: OCR, нейронные сети, сверточные нейронные сети.

Символдарды нейрон желілерінің көмегімен оптикалық тану

Шалахметов А.З., Халықаралық бизнес университеті,
Алматы қ., Қазақстан, e-mail: aidar.shalakhmetov@gmail.com
Аубакиров С.С., Халықаралық бизнес университеті,
Алматы қ., Қазақстан, e-mail: aubakirov.sanzhar@gmail.com

Жиырма бірінші ғасыр – мәліметтерді жахандық автоматизация мен цифралау ғасыры. қазіргі уақытта оптикалық тану жүйелеріне, оның ішінде символдарды тану жүйелері үлкен сұранысқа ие. Оптикалық тану саласында қойылған мақсаттарға жету үшін түрлі амалдарды қолданады. Ол амалдардың кейбірі классикалық ерекше бөлу әдісінде негізделген. Басқалары машиналық үйрету алгоритмдарында негізделеді. Бұл әдісте машиналық үйрету саласының зерттемелері қарастырылады да, келешектегі зерттеу жұмыстарына ұсыныстар қарастырылады. Берілген мақала, машиналық үйретудің негізін сипаттайтын екі басылымға қарап негізделген. Біз берілген әдістің амалы және техникасымен танысу үшін көрсетілген тәжірибелерді қайта орындадық. Сонымен қатар оптикалық тануға әсерін тигізетін негізгі принциптарды анықтамақ болдық. Біз басты екі архитектураны талдадық: көп қатпарлы перцептон мен үйірткілі нейрон желілері. қортындысында біз машиналық үйрету алгоритмдерінің негізімен танысып, келешектегі зерттеулер стратегиясын құрдық. Келесі жұмыстарымызда зерттемелер мен түрлі архитектуралардың тәжірибелерін табуға болады. Нәтиже бірдей шарттарда конвективті нейрондық желілердің әртүрлі модельдері арасындағы әнімділіктің айырмашылығын көрсетеді. Оған қоса біз түрлі утилиталар, бағдарламалық жабдықтау мен жүйені жүзеге асыру процессін жасайтын салаларды қарастырдық. Түйінді сөздер: символдарды оптикалық тану, нейронды желілер, үйірткілі нейрон желілері.

Түйін сөздер: OCR, нейрондық желілер, конвективті нейрондық желілер

1 Introduction

Nowadays there is growing demand for the software system to recognize characters in a computer system when information is scanned through paper documents. This paper presents proposal work in the field of Optical Character Recognition that describes chosen methods of character recognition. In our days technological development growth increasing exponentially in many fields. Machine learning is one of the rapidly increasing and most interesting fields. An approach, that provides machine learning, can be applied to solution of almost any problem. Machine learning and deep learning[1] techniques allows to process a large amount of unstructured data, perform tasks much faster, unlike any person who might spend dozens of man-hours. Main advantage of these techniques is the ability to generalize and abstract data[2]. First prototype of neural network was made in 1943[3], which was conceived as a perception element. With the appearance of high definition cameras, big amount of different data in different areas, and computation capacities, development of machine learning is increasing faster and faster. Nowadays there are large amount of variations of neural networks as well as areas of usage[4]. The architecture and type of neural networks, as well as the approach, is selected depending on the main problem. Neural networks can be divided by their purposes on these categories namely: image recognitions, decision-making, clustering, prediction making, approximation, associative memory, data analysis and optimization[5]. Neural networks can have various shapes and sizes. Usually these shapes, sizes, and architecture in general, are chosen according to the problem. For instance, convolutional neural network greatly deals with the problem of object recognition due to saving the spatial structure of data. Neural networks have ability to compute almost any function. No matter what the function, it is guaranteed to be a neural network so that for every possible input value, there is some close approximation of output value from the network due to universal approximation theorem. Universal approximation theorem states that a a single hidden layer feed-forward neural network can approximate continuous functions on compact subsets of real numbers. Neural networks take their basis from biology. The whole mechanism is copied from the per-

formance of a brain. Similarly, as the signals flow between neurons by synapses, the data flows between mathematical elements, also called neurons. First try of creation of mathematical model, which simulates brain processes, was made by Warren McCulloch and Walter Pitts in 1943. Since then neural networks become more complex and can perform various tasks. Most of the problems, which can be solved by neural networks falls under recognition, prediction, classification, approximation and data analysis problems[6]. For different problem there are different types and architectures of the neural network - with supervised and unsupervised training, fixed and dynamic weights, analog and digital input data, and with different propagation algorithms[7]. Basic neural networks are the mathematical models, which is built to compute mathematical operations in different variations. Therefore, neural network contains several main elements - nodes, which are called neurons, weights, activation functions. Neural networks usually are built step-by-step by grouping all elements in so called layers. Weights are represented as links between each neuron of each layer. Each weight has its own value, which represents "relation" between neurons in the form of certain value. This value is passed to the target neuron in activation phase. Activation function is a mathematical function, which takes sum of all weights as an arguments and, by applying certain function, computes the output. So neuron gathers all described elements in itself. Basically neuron decides, will it pass processed input data further or not. This approach is successfully introducing to various fields. One of the fields of interest is computer vision[8], [9] and artificial perception of the handwritten text. Using neural networks in optical character recognition and text recognition systems can make big contribution in data digitization.

2 Related works

The book, on which this work is partially based, is written by by Michael Nielsen and called "Neural network and Deep learning"[10]. This online book describes basic principles of neural networks and its implementation on Python. Convolutional neural networks were described in paper entitled "Gradient-Based Learning Applied to Document Recognition"[11] by Y. LeCun, L. Bottou, Y. Bengio and P. Haffner. The paper describes the process they used to achieve up to a 99.1% accuracy on the MNIST dataset, using both a 3-layer convolutional network and a 5-layer network that failed to outperform the former. Another great example of appliance of machine learning techniques to text recognition problem is described in paper "Recognizing Handwritten Digits and Characters" by Vishnu Sundaresan and Jasper Lin, 2015[12]. Paper describes appliance of convolutional neural networks to text recognition problem.

2.1 "Neural network and Deep learning", Michael Nielsen

The research was made on multilayer perceptron with MNIST dataset[13]. Neural networks consist of layers which contains some amount of neurons respectively[14]. These layers can be categorized as input layers, hidden layers and output layers. Input layers contains input neurons[15]. Those neurons do not process data with activation functions, they just pass them further. Those neurons can be also called "entry points" of neural networks. Neurons of hidden layers gather data from each neuron of input layers, sum outputs of applied activation functions on weights, and pass it further (or do not pass at all). Most of all computations are performed in these layers. And more model has hidden layers, more powerful the model

becomes, but it also takes more resources for computations in the other hand. Neurons of output layers do the same operations as neurons of hidden layers, but they have the only output weight (or "link") due to absence of next layer. The output value of this type of neuron can be interpreted as the prediction and should be considered in the context. Figure 1 shows four-layer neural network. Activation function serves as the firing mechanism of the output.

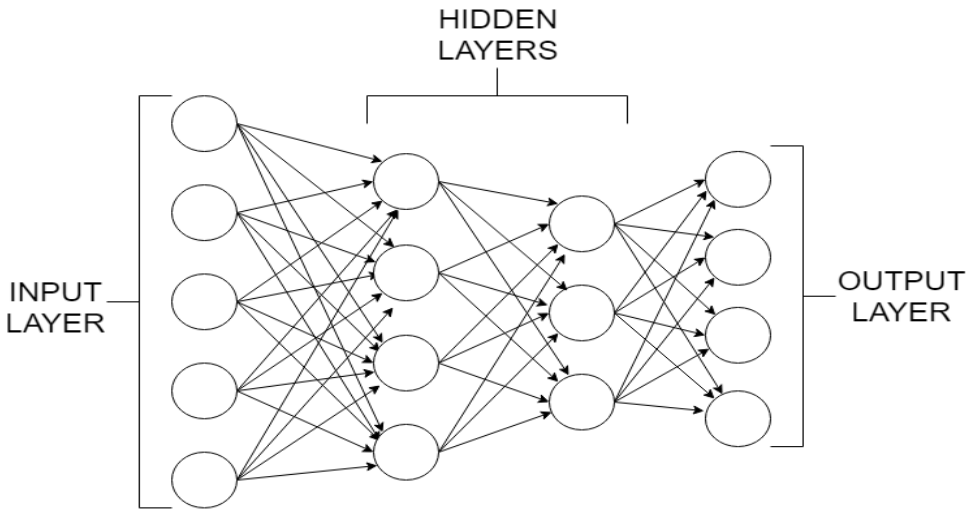

Figure 1: Layers of neural network

In other words, activation function shows whether is neuron activated or not. Activation function can be almost any function, which fits the mathematical model. It is obvious that activation functions have different behavior in the same domain, which affects the output. Usually sigmoid[16], hyperbolic tangent, Rectified Linear Unit and Softmax functions are used as activation functions in neural networks. Activation functions applied to the model based on assumed result. Each element of the neural network imitates a corresponding component of a brain cell. Neuron or perceptron imitates a neural cell itself, weights stand for axons, dendrites and synapses combined, and activation function can be interpreted as neurotransmitter. In reality organic brain is more complex and neural network should not be considered as exact digital analogy to organic brain. When the architecture of the model is defined and weights are randomly set, it can take input data and compute output value. Most likely the output would be different as it is expected, i.e. predict incorrectly. In order to get right predictions with high accuracy, the model must be trained. During the supervised training model takes input data, computes it and generates output value, which is compared to the actual result. The difference between the output and the actual result can be determined by different functions called loss functions. The choice of the loss function directly effects on the performance of the neural network. Loss functions can be divided into two groups - regressive loss functions and classification loss functions. Mean Absolute Error, Mean Square Error or Arctan could be used as regressive loss functions. Cross-Entropy loss, Hinge loss or logistic loss could be used in solution of classification problems. The main idea of the training process is changing the values of the weights and aspiring to reduce an error. One of the methods for defining weight such

that would reduce the loss function is taking the derivative of that loss function with respect to every parameter. This method comes down to finding such weights (or parameters in case of the function), that would minimize the outcome of the loss function. When derivative is found for each parameter, these parameters are being tuned according to the derivative. This process is called backpropagation. An initial experiment, described in this work, was performed on multilayer perceptron with MNIST dataset. MNIST stands for "Modified National Institute of Standards and Technology". This dataset contains over 60,000 samples of handwritten digits. All samples are normalized, smoothed, resized to 28x28 pixels and converted into grayscale images. The model consists of 1 input layer, 1 hidden layer and 1 output layer. All neurons of each layer are connected to all neurons of next layer, except of output layer. An MLP architecture is shown on figure 2. Model in this experiment[17] is created with following architecture: input layer consists of 728 neurons, each for every pixel of 28x28 image[18]. Every neuron gets pixel intensity as input value. Sigmoid is applied as an activation function. Second layer has 30 neurons. Activation function is also sigmoid. Each neuron gathers all values from input connections, applies activation function and produces an output value. Output layer have 1 neuron for each classification class. It means that the output layer has 10 neurons for digits from 0 to 9. Activation function in last layer is softmax. After all, input data is passed forward and computed, model calculates loss with cross-entropy function. The last step of the iteration is backpropagation. This iteration shows the process of learning for one sample. One run with all samples is called epoch. To increase accuracy, training process can be repeated with several epochs. With this configuration Michael Nielsen has achieved accuracy of 96%.

2.2 "Recognizing Handwritten Digits and Characters", V. Sundaresan and J. Lin

Another interesting research in this field was made in 2015 year by Vishnu Sundaresan and Justin Lin from Stanford University. Authors used convolutional neural network for digit and character recognition. Convolutional matrices are the main elements of this type of architecture. These matrices, also known as filters or kernels, compute dot product of its entries and the same sized input. In other words filter overlays an image, slides over it and searches for specific pattern. Each kernel learns to recognize a specific pattern like line, curve or corner. The figure 2 shows the convolution process. This approach allows to save spatial structure of the data. The result is stored in a so called feature map. Sometimes feature maps are compressed to reduce the number of parameters. This procedure is known as pooling. The figure 3 shows the CNN architecture. Authors used both MNIST and Chars74K[20] datasets in their experiments. They have built their own model along with LeNet and AlexNet. Figure 4 shows the architecture of custom model: With this configuration V. Sundaresan and J. Lin has achieved accuracy of 98% on MNIST dataset and 71% on Chars74K dataset comparing to the lower result of other networks.

3 Methods

3.1 Datasets

We conducted experiments with MNIST and Chars74K as datasets. MNIST dataset is a large set of thousands of handwritten digit samples. This dataset is used for basic calibration and

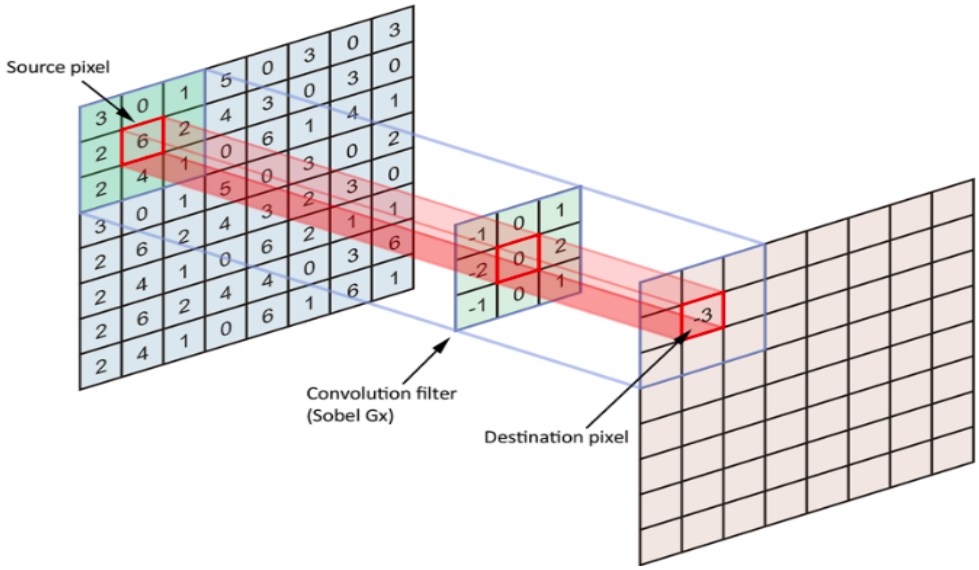

Figure 2: Convolution

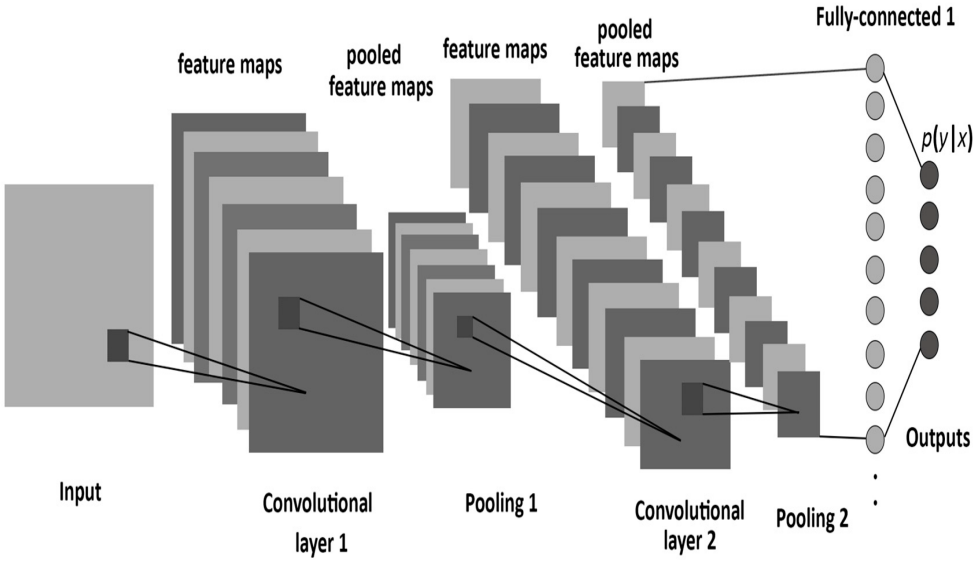

Figure 3: Architecture of Convolutional Neural Network[19]

tuning of different methods for image recognition. Each sample is preprocessed and unified. Preprocessing translates images into one common format. Unified format of images is the requirement for convolutional neural networks. In other words, architecture of convolutional neural networks based on structure of input data. Preprocessing also helps to generalize image features and, therefore, it directly affects on performance of convolutional neural networks[21], [22], [23]. The most common parameters of images are height, width, pixel intensity and number of channels. Typically, images have one or three color channels for grayscale or RGB respectively. Basing on these parameters, we can define architecture of neural network. To

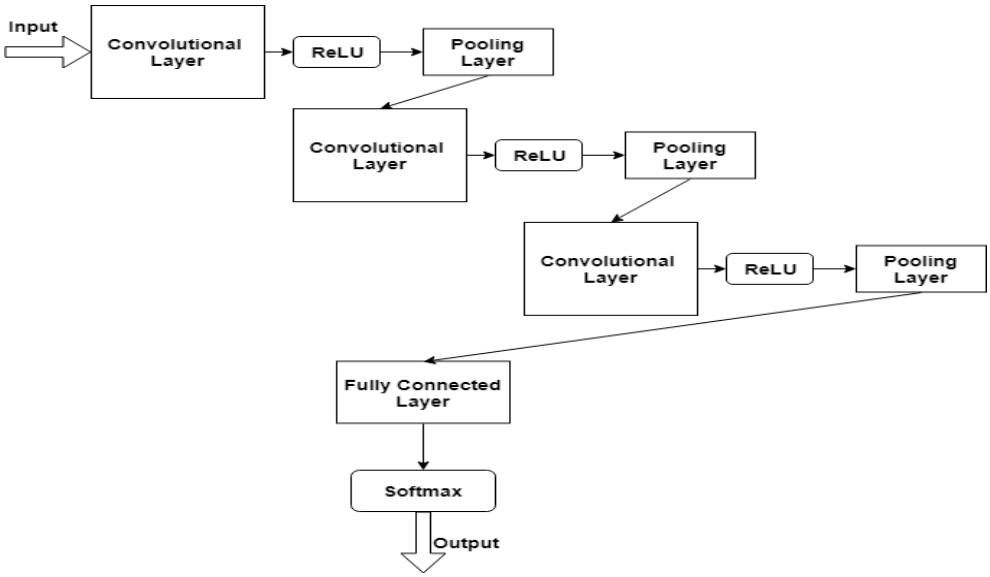

Figure 4: Pipeline of CNN

Table 1: Aggregation of Test Accuracies

	MNIST	Chars74K
LeNet	99%	45.3%
AlexNet	Na	63.4%
Custom network	98.1%	71.7%

increase accuracy of the model, we should perform further preprocessing. Usually it consists of resizing and rescaling, normalization and mean distribution of pixels of images. Convolutional neural network models are designed with fixed number of input nodes and kernels. Each node takes one pixel of image as input. Each kernel is intended for one of the channels. Therefore, for RGB input image should be defined three input kernels for each channel. First of all it is must be ensured that images have the same size and ratio. If some of the samples are not fitting to the specified size, they must be cropped and rescaled. Image should be cropped with the region of interest in center. Afterward image should be rescaled into defined size. There are plenty techniques to perform these transformations. Normalization of input data leads to similar distribution. One of the normalization techniques is to subtract from pixel mean value of all pixels and divide it by the result of the standard deviation. These actions lead to faster convergence during backpropagation. Another step of preprocessing is the data augmentation. This method transforms samples from dataset in different ways, such as rotations and scaling. Due to this method dataset is increases in dozens and helps to prevent recognition of unwanted features. MNIST is the dataset of handwritten digits, which contains about 60 000 of training data and 10 000 validation data. All of the samples are normalized, centered and resized to 28x28 pixels. Images in MNIST dataset grayscale, i.e. the have only

one channel representing gray tone intensity. Chars74K is the set of typed letters. Data in this dataset is raw, that means it was not preprocessed. Samples in dataset has different sizes, not normalized and augmented. All images in Chars74K in RGB color model. Chars74K contains around 7 000 samples, which we divided into training, validation and test sets in proportion of 60%, 20%, 20% respectively.

3.2 Neural network models

We conducted few experiments with two different convolutional neural networks. LeNet and AlexNet architectures was chosen for research to repeat the experiment, described in paper by Vishnu Sundaresan and Jasper Lin. Firstly LeNet architecture was introduced by Yann LeCun in 1989[24] and last modified version was introduced in 1998. LeNet model consist of two convolutional and three consecutive fully connected layers. There also pooling layers after each convolutional layer. AlexNet model was designed by Alex Krizhevsky and Ilya Sutskever in 2012[25]. This model contains five convolutional layers and 3 fully connected layers. Both models use ReLU loss for each layer. We expect the AlexNet model will show better results than LeNet model due to the depth of the model.

3.3 Tools and utilities

We used latest tools and utilities as an environment for experiments. We used latest version of PyTorch – machine-learning framework, along with Numpy, Pandas and Matplotlib frameworks, provide transparent and simple modeling process. Google provides free platform for research purposes. Google Colaboratory platform provides with free preconfigured Python environment and one GPU (in latest versions TPU – tensor-processing unit).

4 Results

4.1 Experiments

Firstly, we conducted experiments on MNIST dataset. We designed LeNet model with 2 convolutional layers and 3 fully-connected layers. We have chosen Cross-Entropy Loss as a loss function and Adam algorithm as an optimizer. We trained model 20 epochs long with batch size of 128 samples per batch. After conducting experiments, we got 99% accuracy – model correctly predicted 9904 sample out of 10 000. Same experiment we tried with AlexNet model with 5 convolutional layers and 3 fully-connected layers. Cross-Entropy Loss and Adam optimizer also had been chosen for this experiment. As the result we got 98.9% accuracy – model predicted 9896 samples out of 10 000. Concluding the experiment we can state that LeNet and AlexNet models showed almost the same result. We conducted experiments with Chars74K dataset. The dataset was not normalized. We rescaled images to the size of 128 pixels. We designed LeNet and AlexNet models with same architecture as in previous experiments. We used different learning rates increasing each by ten times per experiment. First experiment was made on LeNet model. As it is shown on table 2, different learning rates affect on results. With higher learning rate the model have not learned. Optimizer could not get to the global minimum during backpropagation. Learning rates with values equaled to 1e-3 and 1e-4 shown almost similar results – 74% and 71% respectively. Figure below shows loss

Table 2: LeNet learning results

Experiment №	Learning rate	Learning time	Accuracy	Loss
1	1e-2	0:35:40.46	8% (118/1541)	0.0320
2	1e-3	0:35:46.23	74% (1138/1541)	0.0093
3	1e-4	0:34:47.73	71% (1093/1541)	0.0100

reduction during training process. As you can see, the optimizer with learning rate equaled to 1e-3 optimizes weights faster than with learning rate equaled to 1e-4. Optimizer with 1e-2 learning rate could not minimize the loss at all.

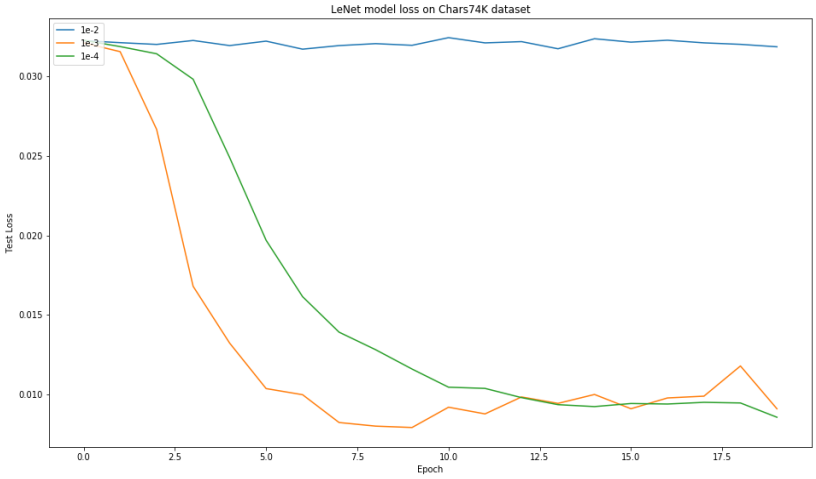

Figure 5: LeNet model loss

As it is shown, on figure 6, most accurate predictions were made while the optimizer was training the model with learning rate equaled to 1e-3. We can state that learning rate in the second experiment is the most optimal, because learning rate in the third experiment takes more time to learn to get same results. On the same dataset, we also trained AlexNet model. The result shown in table 3 below.

AlexNet had reached maximal accuracy with learning rate equaled to 1e-3 – 78% with 1197 correct predictions out of 1541. The third experiment showed much less accurate predictions than the experiment on LeNet model with the same learning rate. That could happen due to greater amount of convolutions in AlexNet model than in LeNet and it takes more computations to reach the same result. Graphical visualization of loss and accuracy is shown

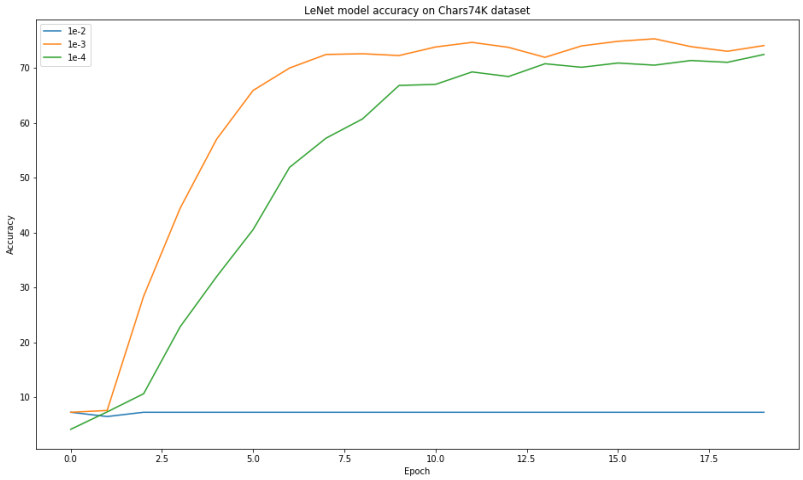

Figure 6: LeNet model accuracy

Table 3: AlexNet learning results

Experiment №	Learning rate	Learning time	Accuracy	Loss
1	1e-2	0:40:15.10	7% (107/1541)	0.0323
2	1e-3	0:41:06.41	78% (1197/1541)	0.0079
3	1e-4	0:41:45.23	64% (990/1541)	0.0108

on the figure 7 and 8.

Comparing two models within the same conditions, we can state that AlexNet model showed slightly better result than LeNet model. AlexNet model is more powerful than LeNet, and the result can be improved by preprocessing of dataset, but it takes more time to train this network. The figure below describes loss decrease of each model over learning epochs. LeNet started to decrease loss right from the beginning, while AlexNet model started to learn after ninth epoch. At the end of twentieth epoch, AlexNet showed slightly better results than LeNet.

Same result could be observed of accuracy comparison graphic. AlexNet started to improve prediction accuracy right from eights epoch and reached its best result at the end of twenties epoch, meanwhile LeNet showed steady growth in accuracy, but could not beat the AlexNet at the end.

Comparing implementations of AlexNet and LeNet models, it is noticeable that there are great difference in total amount of trainable parameters.

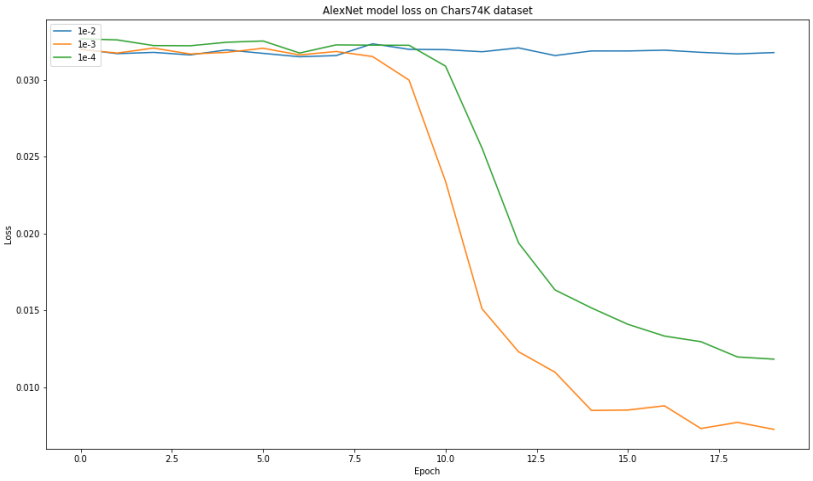

Figure 7: AlexNet model loss

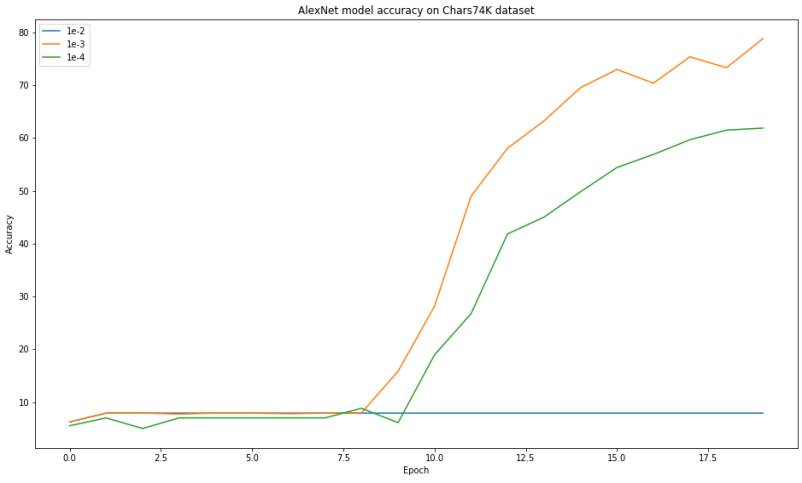

Figure 8: AlexNet model accuracy

Implementation of LeNet model contains 5 times more trainable parameters than AlexNet model. Transition from convolutional layer to fully-connected layer generate large number of trainable parameters. Basing on these numbers, AlexNet model uses 8 MB of memory to feedforward the signal and LeNet uses about 45 MB of memory to pass the same signal.

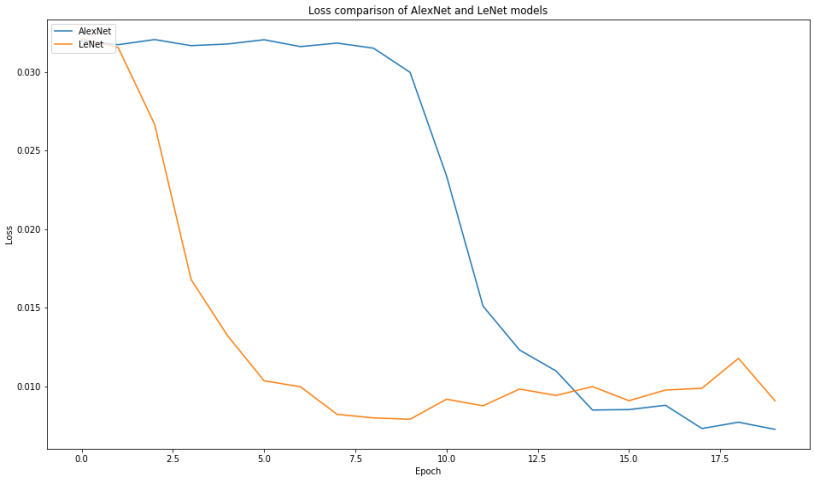

Figure 9: Loss comparison

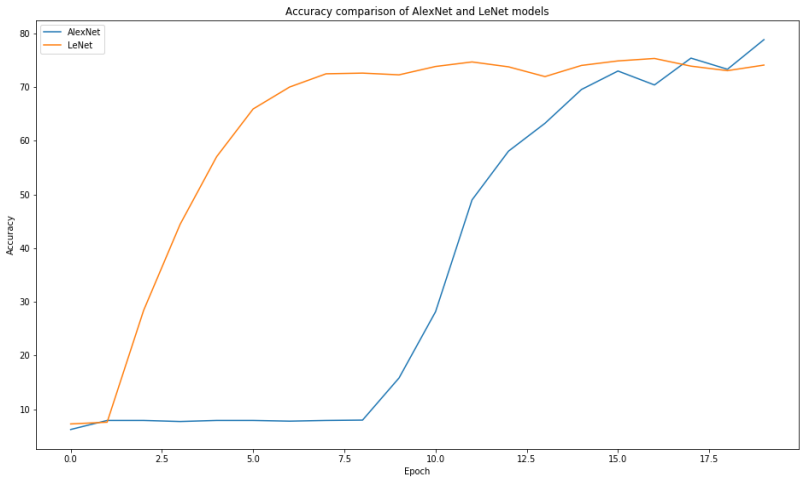

Figure 10: Accuracy comparison

5 Conclusion

In conclusion, we can state that it is obvious that convolutional neural networks could be used as solution of optical recognition problem. It is reasonably to study these fields due to the steady development of object recognition and machine learning in general. In this work, we observed related works in optical character recognition with neural networks area. Also, we conducted experiments with implementations of AlexNet and LeNet models on Chars74K dataset. As the result of experiments, implementation of AlexNet model has shown greater

Table 4: Aggregation of Test Accuracies

Model	Total number of parameters	Memory usage
LeNet	11,412,382	45 MB
AlexNet	2,021,918	8 MB

prediction accuracy than LeNet model. In the other hand AlexNet took longer time to learn and reach the same result as LeNet model. At the first sight, it could seem that implementation of AlexNet model has more trainable parameters than LeNet due to the greater number of layers in architecture. However, convolutional layers generate less parameters than fully connected layers. Taking in account all results of experiments, we can conclude that AlexNet model showed better result in general, than LeNet model. AlexNet could be used in optical character recognition and text recognition systems. Data from Chars74K dataset, which was used in experiments, was not preprocessed and augmented. Preprocessing of data is very important step; it can highly increase accuracy. In further researches, we should consider using different models and combination of architectures, such as recurrent neural networks and combination of convolutional and recurrent networks.

References

[1] Yann LeCun, Yoshua Bengio and Geoffrey Hinton, "Deep learning", Nature (2015): 436–444.

[2] Behnam Neyshabur et al., "Exploring Generalization in Deep Learning", *accessed* October 14, 2018, https://papers.nips.cc/paper/7176-exploring-generalization-in-deep-learning.pdf.

[3] Warren S. McCulloch and Walter H. Pitts, "A logical calculus of the ideas immanent in nervous activity", Bulletin of Mathematical Biophysics (Springer US, 1943): 115–133.

[4] Vidushi Sharma, Sachin Rai and Anurag Dev, "A Comprehensive Study of Artificial Neural Networks", *International Journal of Advanced Research in Computer Science and Software Engineering* 10 (2012): 278-284.

[5] Jayesh B. Ahire, "Real world Applications of Artificial Neural Networks", *accessed* October 14, 2018, https://medium.com/@jayeshbahire/real-world-applications-of-artificial-neural-networks-a6a6bc17ad6a

[6] Sumit Das et al., "Applications of Artificial Intelligence in Machine Learning: Review and Prospect", *International Journal of Computer Applications* 115 (2015): 31-41.

[7] Sonali B. Maind and Wankar Priyanka, "Research Paper on Basic of Artificial Neural Network", *International Journal on Recent and Innovation Trends in Computing and Communication* 2 (2014), *accessed* November 5, 2018, http://www.ijritcc.org/download/Research%20Paper%20on%20Basic%20of%20Artificial%20Neural%20Network.pdf.

[8] Alex Krizhevsky, Ilya Sutskever and Geoffery E. Hinton, "Imagenet classification with deep convolutional neural networks", Advances in neural information processing systems 25 (2012), *accessed* November 5, 2018, https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.

[9] Christian Szegedy et al., "Rethinking the inception architecture for computer vision", *Paper presented at the 29th IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, Nevada, June 26 – July 1*, 2016.

[10] Michael Nielsen, "Neural network and Deep learning", *accessed* October 20, 2018, http://neuralnetworksanddeeplearning.com/index.html.

[11] Yann LeCun et al., "Gradient-based learning applied to document recognition", *Proceedings of the IEEE* 86 (1998):2278-2324.

[12] Vishnu Sundaresan and Jasper Lin, "Recognizing Handwritten Digits and Characters", *accessed* October 20, 2018, http://cs231n.stanford.edu/reports/2015/pdfs/vishnu_final.pdf.

[13] Michael Nielsen, "Neural Network and Deep Learning: Learning with gradient descent", *accessed* October 21, 2018, http://neuralnetworksanddeeplearning.com/chap1.html.

[14] Michael Nielsen, "Neural Network and Deep Learning: The architecture of neural networks", *accessed* October 21, 2018, http://neuralnetworksanddeeplearning.com/chap1.html.

[15] Michael Nielsen, "Neural Network and Deep Learning: Perceptron", *accessed* October 21, 2018, http://neuralnetworksanddeeplearning.com/chap1.html.

[16] Michael Nielsen, "Neural Network and Deep Learning: Sigmoid function", *accessed* October 21, 2018, http://neuralnetworksanddeeplearning.com/chap1.html.

[17] Michael Nielsen, "Neural Network and Deep Learning: Implementing our network to classify digits", *accessed* October 21, 2018, http://neuralnetworksanddeeplearning.com/chap1.html.

[18] Michael Nielsen, "Neural Network and Deep Learning: A simple network to classify handwritten digits", *accessed* October 21, 2018, http://neuralnetworksanddeeplearning.com/chap1.html.

[19] Saleh Albelwi and Ausif Mahmood, "A Framework for Designing the Architectures of Deep Convolutional Neural Networks", Entropy 19(2017): 242-262.

[20] Amit Choudhary, "A Review of Various Character Segmentation Techniques for Cursive Handwritten Words Recognition *International Journal of Information & Computation Technology* 4 (2014): 559-564.

[21] Shuang Wu et al., "L1-Norm Batch Normalization for Efficient Training of Deep Neural Networks *IEEE Transactions on Neural Networks and Learning Systems* (2018), *accessed* November 28, 2018, doi:10.1109/TNNLS.2018.2876179.

[22] Sergey Ioffe and Christian Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift", *Paper presented at the 32nd International Conference on Machine Learning, Lille, France, July 06 – 11*, 2015.

[23] Yann LeCun et al., "Efficient backprop, Neural Networks: Tricks of the Trade", *second edition* (Springer US, 1998): 9-48.

[24] Yann LeCun et al., "A handwritten digit recognition: Applications of neural net chips and automatic learning", Neurocomputing (Springer US, 2005): 303-318.

[25] Alex Krizhevsky, Ilya Sutskever and Geoffery E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks", *accessed* November 20, 2018, https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf