A comparison study between MLP and Convolutional Neural Network models for character recognition

Syrine Ben Driss, Mahmoud Soua, Rostom Kachouri, Mohamed Akil

A comparison study between MLP and Convolutional Neural Network models for character recognition

S. Ben Driss, M. Soua, R. Kachouri, and M. Akil

ESIEE Paris, IGM, A3SI, 2 Bd Blaise Pascal, BP 99, 93162 Noisy-Le-Grand, France

ABSTRACT

Optical Character Recognition (OCR) systems have been designed to operate on text contained in scanned documents and images. They include text detection and character recognition in which characters are described then classified. In the classification step, characters are identified according to their features or template descriptions. Then, a given classifier is employed to identify characters. In this context, we have proposed the unified character descriptor (UCD) to represent characters based on their features. Then, matching was employed to ensure the classification. This recognition scheme performs a good OCR Accuracy on homogeneous scanned documents, however it cannot discriminate characters with high font variation and distortion.[3] To improve recognition, classifiers based on neural networks can be used. The multilayer perceptron (MLP) ensures high recognition accuracy when performing a robust training. Moreover, the convolutional neural network (CNN), is gaining nowadays a lot of popularity for its high performance. Furthermore, both CNN and MLP may suffer from the large amount of computation in the training phase. In this paper, we establish a comparison between MLP and CNN. We provide MLP with the UCD descriptor and the appropriate network configuration. For CNN, we employ the convolutional network designed for handwritten and machine-printed character recognition (Lenet-5) and we adapt it to support 62 classes, including both digits and characters. In addition, GPU parallelization is studied to speed up both of MLP and CNN classifiers. Based on our experimentations, we demonstrate that the used real-time CNN is 2x more relevant than MLP when classifying characters.

Keywords: Real-time character recognition, Neural Networks, Deep Learning, GPU

1. INTRODUCTION

Optical character recognition (OCR) systems were introduced to carry the text and solve many problems like retrieving words from documents,[1] transforming document image to editable text,[4] etc. Most of OCR systems are based on text detection and character recognition.[1,4] The recognition phase proceeds at first by a description, then characters are assigned into predefined classes based on number of attributes provided by a descriptor. Recently, the unified character descriptor (UCD) was proposed to represent characters based on their features.[14] Applied to homogeneous scanned documents, it gives a high OCR accuracy when using matching as a classification technique. However, the simple matching is not efficient when applied to characters with high font variation and distortion. To overcome this problem, classification based on neuronal techniques are employed.[5,12] The most popular classifiers include multilayer perceptron (MLP)[12] and radial bases function (RBF).[5]

A class of machine learning techniques named deep learning was developed mainly since 2006, where many layers of non-linear information processing stages or hierarchical architectures are exploited. Deep learning performance is better than the existing classification methods[30] . In this context, the convolutional neural network (CNN)[4] is able to learn the feature vector directly from the training character image without any hand-crafting to determine the feature vector. Some frameworks enable efficient development and implementation of deep learning methods in aim to optimise training or deployment on CPUs. Available frameworks include, caffe,[7] deeplearning4J,[8] TensorFlow,[9] Theano,[10] Torch,[11] etc. Deep Neural Network (DNN) models are known to be very slow.[21] Indeed, training with large data sets may take several days or weeks. That is why, several frameworks are famous thanks to their strong GPU background[2] which offer speed improvement.[21] In this context, it is reported that the top available frameworks are caffe, theano and torch.[2] Their speed and accuracy were widely studied in the litterature.[2,21]

In this work, we compare at first MLP classification based on UCD description against CNN for character recognition in a set of characters in the chars74k dataset.[20] For CNN, we employ the convolutional network designed for handwritten and machine-printed character recognition (Lenet-5[6]) and we adapt it to support 62 classes, including both digits and characters. In addition, GPU parallelization is studied to speed up both of MLP and CNN classifiers. Based on our experimentations, we demonstrate that the used real-time CNN is 2x more relevant than MLP when classifying characters. In addition, we compared several GPU-based CNN architectures when performing on the chars74k dataset.[20] These architectures are namely : lenet,[6] lenet-5[6] and SPnet.[18] The rest of the paper is organized as follows : Section 2 gives an overview on the multilayer perceptron neural networks. Section 3 introduces the convolutional neural network and gives a set of networks used in character recognition context. Finally experiments are given in Section 4 followed by conclusion in Section 5.

2. MULTILAYER PERCEPTRON BASED ON CHARACTER FEATURE DESCRIPTION

We consider the character description based on the unified character description (UCD).[14] The UCD feature vector is extracted from each character and feeded to the decision stage. The advantage of UCD is to employ a sufficient number of characteristics that helps to discriminate characters efficiently. These characteristics are computed the unification of vertical and horizontal character projections.[14] As shown in figure 1, the employed features are character segment number (NBS), character segment position (TMB, LMR), character barycentre coordinate (Bx, By) and character ratio (R).

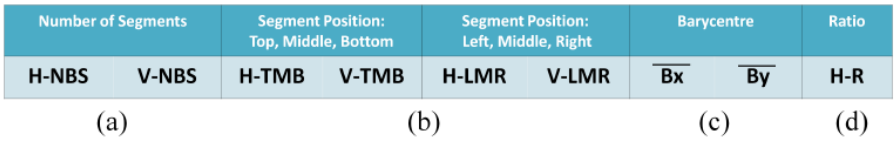

Figure 1. The Unified Character Descriptor (UCD):[14] a. Character Segment Number Feature (NBS), b. Character Segment Position Feature (TMB, LMR), c. Character Barycentre Feature (Bx, By), d. Character Ratio Feature (R).[14]

The matching technique was employed as the decision stage to recognize characters based on the UCD descriptor.[14] For this six templates categorized ionto two classes within serif and sans serif type-faces. Each category includes three sub-templates consisting of three scale ranges : small, medium and large. A high character recognition accuracy was reported on multiscale and multifont characters.[14] However, this accuracy decreases when dealing with higher font variation because the matching technique is limited when dealing only with six templates. To improve this result, we explore the neuronal network approach instead of the matching one. More specifically, we focus on multilayer perceptron networks that are robust decision methods, widely used in character recognition tasks.[30]

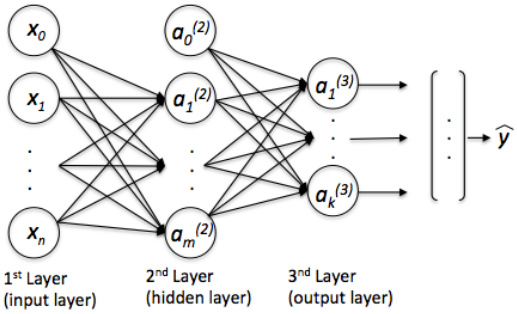

Figure 2. Multilayer perceptron architecture.

Actually, the multilayer perceptron is a feed-forward layered network of artificial neurons, where the data circulates in one way, from the input layer to the output layer. As shown in figure 2 MLP is composed of three layers. The input layer, the output layer and the hidden layer. The input layer contains the inputs features of the network. The first hidden layer receives the weighted inputs from the input layer and sends data from the previous layer to the next one. The use of additional layers makes the perceptron able to solve nonlinear classification problems. The output layer contains the classification result. Several algorithms are used for the learning step of MLP. The common supervised learning technique is called back propagation[15] (figure 3) . It consists of four stages: initializing weights, feed forward, back propagation of errors and weight update.

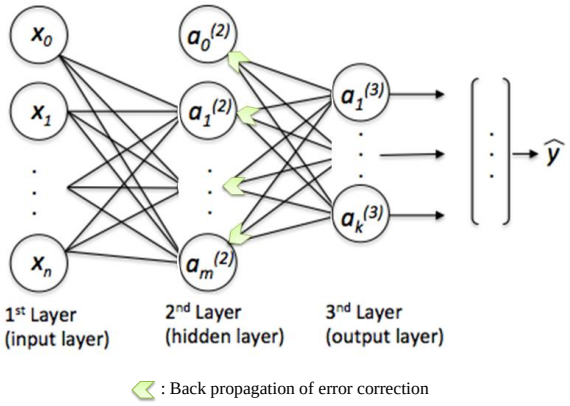

Figure 3. Multilayer perceptron : Back propagation.

Recently, deep learning has been successfully adopted in various areas such as computer vision, automatic speech recognition, natural language processing, optical character recognition, etc. Deep learning models have several variants such as Autoencoders,[25] Deep Belief Network,[26] Deep Boltzmann Machines,[27] Convolutional Neural Networks[28] and Recurrent Neural Networks.[29] As a variant of the standard neural network (MLP), the convolutional neural network are a new emerging deep learning architecture. Following we focus on CNN architectures for character recognition applications.

3. CONVOLUTIONAL NEURAL NETWORK FOR CHARACTER RECOGNITION

CNNs are a derivative of standard Multilayer Perceptron (MLP) neural networks optimized for two-dimensional pattern recognition problems such as Optical Character Recognition (OCR) or face detection.[6] Instead of using fully connected hidden layers as described in the preceding section, the CNN introduces a special network structure, which consists of alternating named convolution and subsamlping layers.

Feature maps generated by convolution layers, contain neurons that take their synaptic inputs from a local receptive field. The weights of neurons within the same feature map are shared. This represents ones of the characteristics of convolutional neural networks. It allows to have replicated units sharing the same configuration, thereby features can be detected regardless of their position in the visual field. Moreover, the fact that weights are shared increases learning efficiency by reducing the number of parameters being learnt. In order to have a data reduction, a sub-sampling operation called pooling is performed. This data reduction operation is applied to the predecessor convolution result by a local averaging over a predefined window. It partitions the input image into a set of non-overlapping windows and then for each sub-region outputs the maximum value. This step is important because it helps to eliminate non-maximal values and to provide a form an invariant translation. The output layers ensures the classification of the input character. In these layers all neurons are fully connected and have a unique set of weights so they can detect complex features and perform classification.

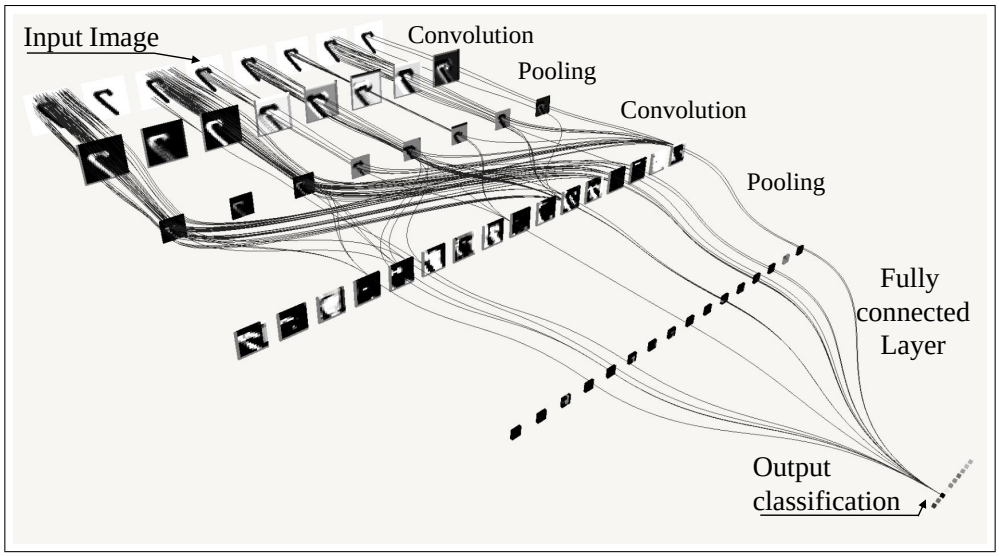

Figure 4. CNN architecture (classification stage, generated from the topological CNN visualisation tool[19]).

Following, we present the well-known Lenet[6] and Lenet-5[6] neural networks, followed by the recently proposed SPnet one.[18] The networks figure 5, figure 7 and figure 5 are generated using the caffe framework.

3.1 Lenet

Lenet[6] network is reported as one of the moost famous convolution networks. it includes three convolution layers : conv1 and conv2 with kernel size 5×5. The sub-sampling operation can be seen in layers pool1 and pool2 with kernel size 2×2. ip1 and ip2 are the output layers for the classification.

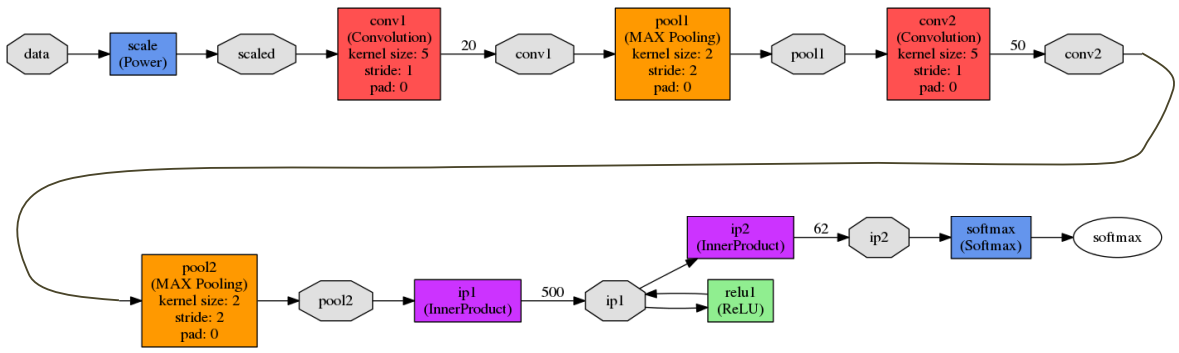

Figure 5. Lenet architecture.

3.2 Lenet-5

LeNet-5[6] is a convolutional network designed for handwritten and machine-printed character recognition. It includes feature maps conv1, conv2 and con3 with kernel size 5×5. Two layers : pool1 and pool2 (with kernel size 2×2) are dedicated to sub-sampling. ip1 and ip2 are the output layers.

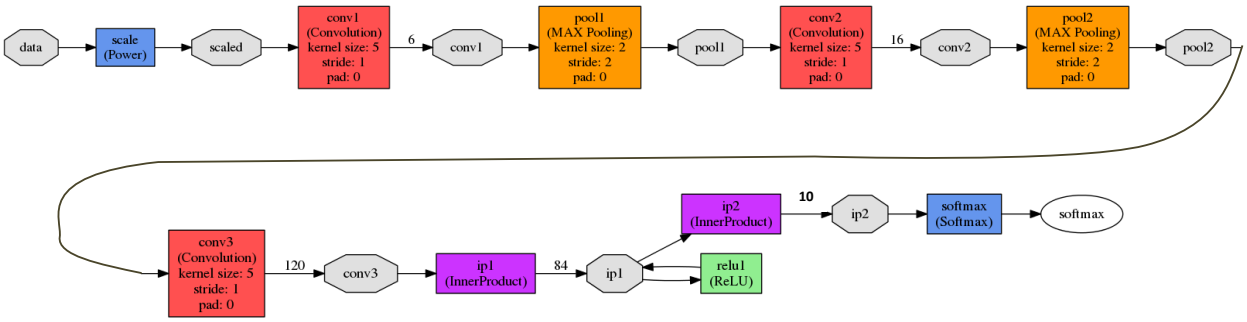

Figure 6. Lenet-5 architecture.

3.3 SPnet

The SPnet[18] is a recently proposed CNN network. Feature maps conv1, conv2 and conv3 with kernel size 5×5 contain neurons that take their synaptic inputs from a local receptive field. The sub-sampling operation are given with three layers : pool1, pool2 and pool3 with kernel size 3×3. ip1 and ip2 are the output layers that ensures the classification.

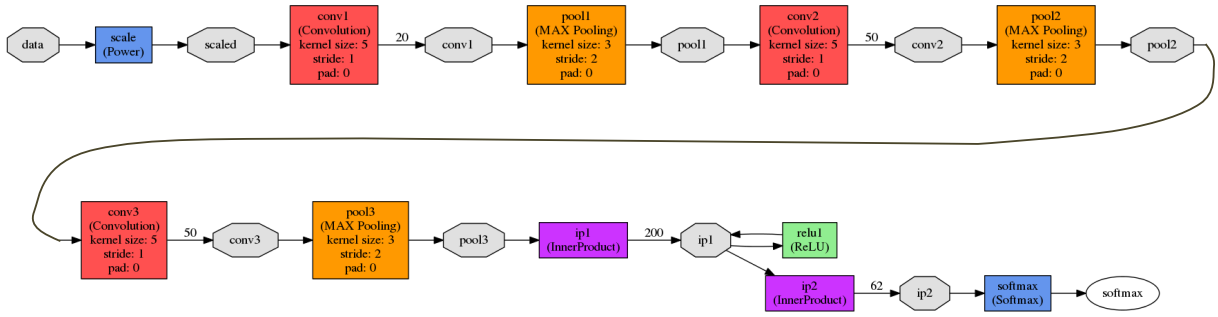

Figure 7. SPnet architecture.

We note that, the biggest drawback of all these three CNN architectures is the long training time. Indeed, since CNN training is very compute and data intensive, training with large data sets may take several days or weeks. The huge number of floating point operations and relatively low data transfer in every training step makes this task well suited for Graphic processing units.

4. REAL TIME CONVOLUTIONAL NEURAL NETWORK

In this section, we introduce the graphic processing unit (GPU) generic architecture. Then, we give an overview of existing frameworks supporting GPUs for CNN architectures deployment.

4.1 Parallel Computing on GPUs

Modern GPUs have evolved from pure graphics rendering machines into massively parallel architecture, recently peaking at 1 TFLOPS. They give a higher computational throughput at relatively low cost compared to CPUs. The CUDA (Compute Unified Device Architecture) programming paradigm allows the development of parallel applications for graphics cards. Computations on the GPU are initiated through kernel functions which essentially are C-functions being executed N times in N parallel threads. Semantically, threads are organized in 1, 2 or 3 dimensional groups, called blocks, as shown in Figure 8.

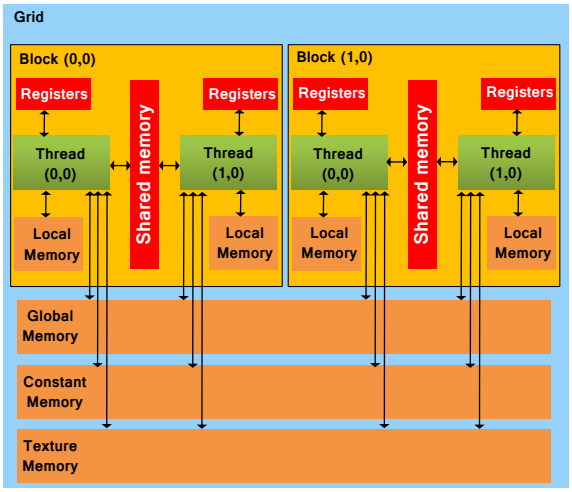

Figure 8. Graphic Processing Unit architecture

Each block is scheduled to run separately from all others on one multiprocessor. They can be executed in arbitrary order simultaneously or in sequence. Each thread has a small private local memory space. In addition, all threads within the same block can communicate with each other through the low-latency shared memory. The much larger global memory has a higher latency and can be accessed by the CPU, thus being the only communication channel between CPU and GPU. Multiple memory accesses can be coalesced into one memory transaction if consecutive threads access data elements from the same memory segment. Following such specific access patterns can dramatically improve the memory bandwidth and is essential for optimizing the application performance.

4.2 The caffe learning framework

Many of deep learning framesworks are very fast in training deep networks thanks to their GPU backend. The graphic processing unit is employed to accelerate the training process and this had led to joint developpment of software libraries such as CuDNN. It is reported that the top three deep learning frameworks are : caffe, theano and torch.[2] In the rest of this paper, we focus on the caffe framework because it enables fast prototyping.

Caffe is a deep learning tool developed by the Berkeley Vision and Learning Center.[2] It is developed in C++ with expression, speed, and modularity in mind which uses CUDA for GPU computation and has commandline, Python, and Matlab interfaces for training and deployment purposes. It separates the definition of the network architecture from actual implementation allowing to conveniently and quickly explore different architectures and layers on either CPU or GPU.
Several types of layers and loss functions are already implemented which can be configured in the form of arbitrary directed acyclic graphs in a configuration file. Caffe supports various layers such as convolution, fully connected and pooling layers, etc. The convolution operation can be computed using either a native implementation (by dense matrix multiplications using Blas) or faster using Nvidia cuDNN.

5. EXPERIMENTATION

In this section we present firstly the employed materials. Then, we make the comparison between MLP and CNN for character recognition. The difference between them is that MLP classifies characters based on the UCD description vector however CNN uses the whole character image. Finally we compare several state of the art CNN architecture namely Lenet,[6] Lenet-5[6] and SPnet[18] based on learning, classification accuracy and execution time on GPU.

5.1 Materials

Our experiments were conducted on PC with and intel Core i5-7200U CPU performing at 2.5 GHz and solid state drive (SSD). In addition, we employ an NVIDIA GeFore 940Mx GPU. This one running in our system consists of 4 multiprocessors with 128 stream processors each, resulting in a total of 512 cores. Each multiprocessor contains 48 KB of on-chip shared memory as well as 65536 registers. The GPU-wide 2048 MB of global memory can be accessed with a maximum bandwidth of 16.02 GB per second. We use the training system DIGITS dev 5.1[17] to create datasets of training and validation images, train a model on the dataset, and test the model in various ways. To create a model, several standard networks are available: LeNet-5,[6] AlexNet[22] and GoogLeNet.[23] In this evaluation, we are using the chars74k dataset[20] for character recognition. This dataset is composed from different subdata-sets, we are going to use only the one containing numbers and English characters figure 9; It contains 62 classes. For our experiments, we divide this dataset into 34658 training images, 12586 validation images (20%) and 15748 test images (25%).

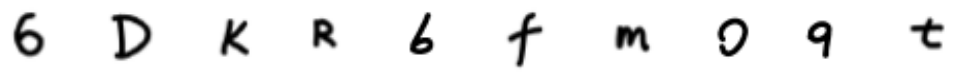

Figure 9. Samples of Chars74k dataset.[20]

5.2 MLP configuration

As shown in figure 10, we use three layers for our MLP (input, hidden and output one). Actually, it has been proven by George Cybanko[24] through the universal approximation theorem that a feed forward network with a single hidden layer, containing a finite number of neurons and with a non-linear activation function is able to approximate a various of training objects with a small error.

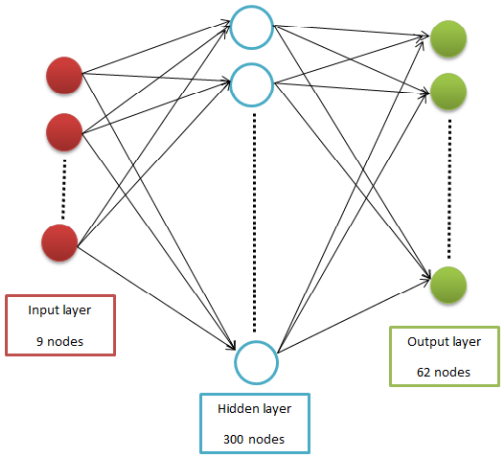

Figure 10. The employed multilayer perceptron architecture.

The input layer is composed of 9 nodes, which takes the description vector as input. Then the hidden layer with 300 nodes and finally the output layer with 62 nodes which are the number of classes: [a.. z], [A.. Z], [0.. 9]. The MLP is trained by the backpropagation algorithm.

5.3 Character recognition accuracy

Lenet-5 network was designed only for digits. It contains only 10 outputs. We used the same Lenet-5 architecture as with a slight modification in the output layers given that we have 62 classes (figure 11) instead of 10. The first layers of the network (C1,S1,C2,S2,C3) are trainable feature extactor that have specific constraints such as local connectivity and weight sharing. Convolutional layers apply a convolutional kernel (5x5) and the sub-sampling ones apply a kernel (2x2). The classification layers in the output are fully connected MLPs. These layers use the extracted local features to perform classification of the input image.

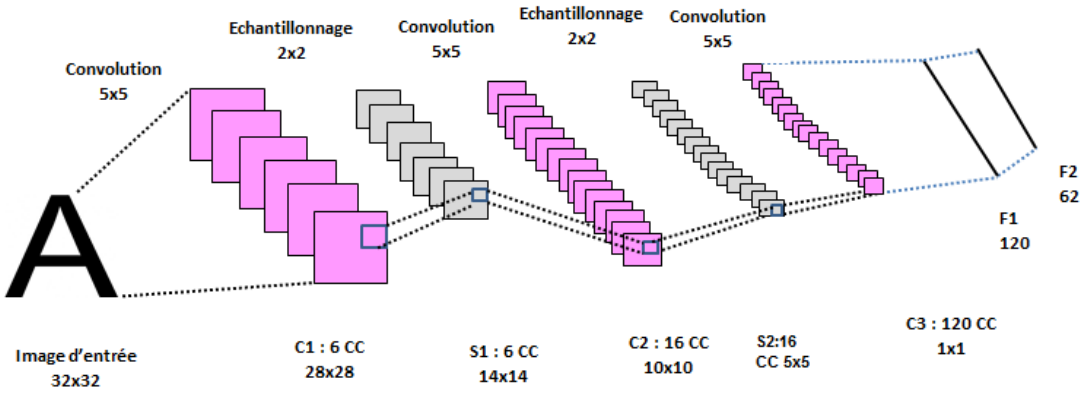

Figure 11. Lenet-5 adapted for 62 classes.

Table 1 shows the performance of learning and classification of MLP and several convolution neural networks namely, Lenet (figure 5), Lenet-5 (figure 11) and SPnet[18](figure 7) on the chars74k dataset.[20] We can see that CNN outperforms MLP on 1.35x on learning and 2x on classification stages. We can see the recent SPNet gives the best accuracy result. This advantage would be more significant when training more complex models with larger data.

Table 1. Accuracy evaluation on GPU of MLP and several CNN models for character recognition on the Chars74k dataset.[20]

	Learning rate (%)	Classification rate (%)
MLP	70.72	43.4
caffe-cnn (Lenet)	88	88.39
caffe-cnn (Lenet-5)	86.23	85.53
caffe-cnn (SPnet)	**89.90**	**90.56**

Figure 12 illustrates the accuracy of compared CNN models during the learning (train) process. For each model (Lenet, Lenet-5 and SPnet), we give ine output from the training set (loss-train) and two outputs from the validation set (loss-val and accuracy-val). Like shown in table 1, we can see that the accuracy reaches $\simeq$90% for the three models. The loss is a function consisting in a summation of the error for each example the train or validation. Indeed, the lower the loss is, the better is the model. The loss-train is low wich mean that the model is trained well. The loss-validation is higher because the number of images used in validation is higher.

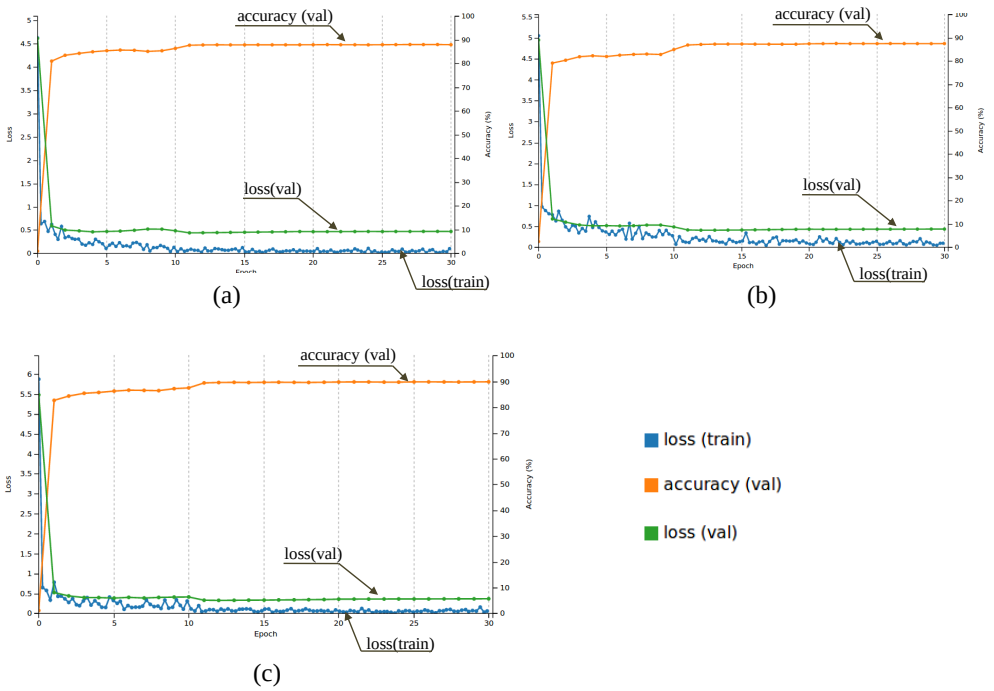

Figure 12. Comparison of training accuracy CNN networks : a. Lenet, b. Lenet-5 and c. SPnet.

5.4 Real time evaluation

We evaluate the learning and classification time on GPU on the chars74k dataset.[20] Table 2 shows the total processing time for learning and classification on all characters of the dataset using batch size of 64.

Table 2. Real time evaluation on GPU of character recognition on the Chars74k dataset.[20]

	Learning (s)	Classification (s) (25% of test data)	Classification (ms) (per image)
caffe-cnn (Lenet)	**120**	**10**	**0.635**
caffe-cnn (Lenet-5)	**100**	**10**	**0.635**
caffe-cnn (SPnet)	556	19	1.2

We report that learning and classification execution time of Lenet and Lenet-5 is lower than that of the SPnet. This is because SPnet model is processing 60×60 images however Lenet and Lenet-5 are processing respectively 28×28 and 32×32 images.

6. CONCLUSION

In OCR systems text is extracted then, characters are described and classified. When coupling with matching technique, a simple matching cannot discriminate characters with high font variation and distortion. To improve recognition, classifiers based on neural networks the multilayer perceptron (MLP) are used ensures high recognition accuracy when performing a robust training. Moreover, the convolutional neural network (CNN), is gaining nowadays a lot of popularity for its high performance. Furthermore, both CNN and MLP may suffer from the large amount of computation in the training phase.

In this paper, we compared MLP and CNN for character recognition. We provide MLP with the UCD descriptor and the appropriate network configuration. For CNN, we employ the convolutional network designed for handwritten and machine-printed character recognition (Lenet-5) and we adapt it to support 62 classes, including both digits and characters. In addition, GPU parallelization is studied to speed up both of MLP and CNN classifiers. Based on our experimentations, we demonstrate that the used real-time CNN is 2x more relevant than MLP when classifying characters. In addition, we evaluated three convolutional neural netwroks namely Lenet, Lenet-5 and SPnet using the Caffe framework.

REFERENCES

[1] X., Xu Z., An P., Liu Q., Lu Y. Advances on Digital Television and Wireless Multimedia Communications. Communications in Computer and Information Science, vol 331. Springer

[2] S.Bahrampour and al., Comparative study of caffe, neon, theano, and torch for deep learning, ICLR 2016

[3] G. J. Alred and C. T. Brusaw and W. E. Oliu, Handbook of Technical Writing, St. Martin's, New York, 2015

[4] Durjoy Sen Maitra, Ujjwal Bhattacharya and Swapan K.Parui.,CNN Based Common Approach to Handwritten Character Recognition of Multiple Scripts,Document Analysis and Recognition (ICDAR), 2015 13th International Conference on., November ,2015

[5] F.Schwenker and al., Three learning phases for radial-basis-function networks, Pergamon., 2000

[6] Y. Lecun,L. Bottou, Y. Bengio and P. Haffner, Gradient-Based Learning Applied to Document Recognition., Proceedings of the IEEE vol.86,no.11,pp.2278-2324., 1998

[7] J. Yangqing and al., Caffe: Convolutional Architecture for Fast Feature Embedding, arXiv preprint arXiv:1408.5093, 2014.

[8] Deeplearning4j Development Team. Deeplearning4j: Open-source distributed deep learning for the JVM, Apache Software Foundation License 2.0. http://deeplearning4j.org

[9] M. Abadi and al., TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[10] Theano Development Team, Theano: A Python framework for fast computation of mathematical expressions, arXiv e-prints, Vol: abs/1605.02688, 2016, http://arxiv.org/abs/1605.02688

[11] R. Collobert, S. Bengio, J. Marithoz, Torch: a modular machine learning software library". 30 October 2002. Retrieved 24 April 2014.

[12] M. Pariseau,Le perceptron multicouche et son algorithme de rtropropagation des erreurs", 2004

[13] L. C. Perelman and J. Paradis and E. Barrett, Mayfield Handbook of Technical and Scientific Writing, Mountain View, Mayfield, April, 1997

[14] M.soua, R.Kachouri and M.Akil, Efficient multiscale and multifont optical character recognition system based on robust feature description., Image Processing Theory, Tools and Applications (IPTA), 2015

[15] D. E. Rumelhart,G. E. Hinton and R. J. Williams, Parallel distributed processing: explorations in the microstructure of cognition, vol. 1, MIT Press Cambridge, MA, USA, 318-362,1986

[16] Steven M. Beitzel, Eric C. Jensen, David A. Grossman, Retrieving OCR Text: A Survey of Current Approaches, Information Retrieval Laboratory Department of Computer Science, 2003

[17] L.Yeager and al., DIGITS: the Deep learning GPU Training System, ICML 2015 AutoML Workshop, 2015.

[18] Dae-Gun Ko, Su-Han Song, Ki-Min Kang, and Seong-Wook Han, Convolutional Neural Networks for Character-level Classification, IEEE Transactions on Smart Processing and Computing, vol. 6, no. 1, 2017

[19] Topological Visualisation of a Convolutional Neural Network, Project by Terence Broad, MNIST Digit Classification, http://terencebroad.com/convnetvis/vis.html

[20] T. E. de Campos, B. R. Babu and M. Varma. Character recognition in natural images. In Proceedings of the International Conference on Computer Vision Theory and Applications (VISAPP), 2009.

[21] Vasilache and al. Fast convolutional nets with fbfft: A gpu performance evaluation. arXiv:1412.7580, 2014.

[22] A.Krizhevsky and al,ImageNet Classification with Deep Convolutional Neural Networks, Advances in Neural Information Processing Systems 25,1097-1105,2012, Curran Associates, Inc., http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[23] C.Szegedy and al., Going Deeper with Convolutions, Computer Vision and Pattern Recognition, 2014

[24] G.Cybenko. Approximation by superpositions of sigmoidal function. Mathematics of control, signal and systems, 2(4) : 303-314, 1989.

[25] P. Vincent and al., Extracting and composing robust features with denoising autoencoders, in Proceedings of the 25th international conference on Machine learning. ACM, 2008, pp. 1096-1103.

[26] G. E. Hinton, S. Osindero, and Y.-W. Teh, A fast learning algorithm for deep belief nets,Neural computation, vol. 18, no. 7, pp. 1527.1554, 2006.

[27] R. Salakhutdinov and G. E. Hinton, Deep boltzmann machines. in AISTATS, vol. 1, 2009, p. 3.

[28] P. Sermanet, S. Chintala, and Y. LeCun, Convolutional neural networks applied to house numbers digit classification, in Pattern Recognition (ICPR), 2012 21st International Conference on. IEEE, 2012, pp. 3288.3291.

[29] K.-i. Funahashi and Y. Nakamura, Approximation of dynamical systems by continuous time recurrent neural networks, Neural networks, vol. 6, no. 6, pp. 801806, 1993.

[30] X.Hu and al. Deep-Learning-Based Classification for DTM Extraction from ALS Point Cloud, Remote sensing, 2016

[31] J.Hocking, M.Puttkammer, Optical character recognition for South African languages, in: Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference, 2016