2017 9th International Conference on Knowledge and Systems Engineering(KSE)

Advertisement Image Classification Using Convolutional Neural Network

An Tien Vo[1] , Hai Son Tran[2], Thai Hoang Le[3]
[1]Department of Computer Science, University of Information Technology - VNUHCM, Ho Chi Minh City, Vietnam
anvt.10@grad.uit.edu.vn
[2]Department of Informatics Technology, HCM University of Education, Ho Chi Minh City, Vietnam
haits@hcmup.edu.vn
[3]Department of Computer Science, University of Science - VNUHCM, Ho Chi Minh City, Vietnam
lhthai@fit.hcmus.edu.vn

***Abstract*— Image classification is critical and significant research problems in computer vision applications such as facial expression classification, satellite image classification, plant (fruits, flowers, leaf…) classification base on images. This paper proposes the image classification model applied for identifying the display of the online advertisement. The proposal model uses Convolutional Neural Network with two parameters (*n, m*) where n is a number of layers and m is number of filters in Conv layer. The proposed model is called nLmF-CNN. The suitable values of parameters *(n, m)* for advertisement image classification are identified by experiments. The input data of the proposed model are online captured images. The processing components of nLmF-CNN are developed as deep neural networks using ConvNetJs library. The output of the proposed model is YES/NO. YES means that the advertisements display clearly. NO means that the advertisements do not display or not clear. The experimental results 86% in our normalizing dataset showed the feasibility of a proposed model nLmF-CNN.**

Keywords- Image Classification; Convolutional Neural Network (CNN); Advertisement Image Classification using CNN.

I. INTRODUCTION

In recent years, deep learning is hot research topic, and convolutional neural network is the popular model for image recognition, image classification, image clustering. CNN [1] has been established as a state-of-the-art result both image recognition and classification.

The purpose of image classification is to identify the labels (or the percentage of belongs to labels) for the given images. In general, there are some classification methods such as K-Mean, Nearest Neighbor Classifier, K- Nearest Neighbor (K-NN), Support Vector Machine (SVM), Convolutional Neural Network (CNN). K-Mean and K-NN [2,3] are the classic methods used for classification. Both K-Mean and K-NN are simple, easy to understand and give an accepted result.

SVM [4] has three approaches: supervised, unsupervised, semi-supervised. Supervised SVM is constructed using a labeled data (the training set), unsupervised SVM used unlabeled data, semi-supervised SVM is constructed using a mixture of labeled data (the training set) and unlabeled data (the working set). Recently, CNN [1] constitute one such class of models and suitable for digital image classification. CNN has many connections and parameters so that they are easier to train, while their theoretically-best performance needs to improve.

Therefore, we will apply CNN for online advertisement image classification. This is a real problem come from advertisement service providers. They want to check whether advertisements display clearly in their online environments in order to match with dealing invoices or not.

The remainder of this paper is organized as follows: Section II devotes to study of related work. Section III provides a detailed exposition of our proposal model CNN which has been compiled for classification with advertisement images. Section IV contains a discussion of the experiments and evaluations of our implement model for advertisement image classification problems. Conclusion and future work are given in the final section.

II. BACKGROUND AND RELATED WORK

A. *Image Classification with Convolutional Neural Network (CNN)*

ConvNetJS [5] is an open library which implements Deep Convolutional Neural Network. It is written by JavaScript and HTML with many support features for researchers. It is lightweight, effective, and able to use CPU instead of GPU, and also reach out the real-time target.

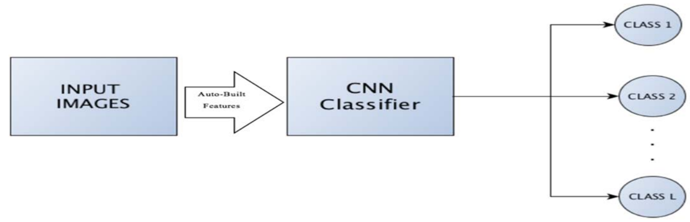


Fig. 1. CNN Images Classifier.

Figure 1 shows an overview of our proposed CNN model for advertisement image classification. The input data are online captured images. The processing component use CNN techniques. The output is in {YES, NO}.

According to Figure 1, an advertisement image classification system based on CNN includes 4 steps:

Step 1: Input - The URL address of website need to check advertisement will be displayed.

Step 2: Image capture –

Step 2.1: Displaying HTML response to the browser.

Step 2.2: Capturing screen display of the browser.

Step 2.3: Saving images.

Step 3: Using CNN to classify images.

Step 4: Output –The classification conclusion.

B. *Normalize Dataset using with ConvNetJS*

The collected data is 9.49 GB, collected from 06/2016 to 10/2016 by capturing the display of TWC News on multiple browsers such as Chrome, Firefox, IE on cross-platforms (Windows, Linux, macOS).

There are many positions to place the advertisement on the website. We can put them on the left, on the right, on the top, and/or on the bottom.

Fig. 2. Position advertisement display in website.

Figure 2 shows an example of advertisements position on the website. The advertisements display clearly or are unclear in the yellow rectangle. For example, figure 2 has two clear advertisements and one unclear advertisement. Figure 2a is an example which the advertisement is put on the right. Figure 2b is an example which we put the advertisement on the bottom. Figure 2c is an example which the advertisement does not appear.

The dataset is organized into 5 folders month by month from June to October as below. Each folder has capture images as PNG or GIF and logs information as TXT format.

TABLE I. Dataset Description.

Folder Name	# Images	Size
Capture Image 201606	633	1.3 GB
Capture Image 201607	1335	3.1 GB
Capture Image 201608	2003	4.6 GB
Capture Image 201609	4785	189.5 MB
Capture Image 201610	5770	309 MB

Table I shows information (number of images and total images size) of 5 image folders in the dataset. The first folder captured in June-2016 has 633 images. Number images of folders increase from 633 to 5770 corresponding to the capturing time from June to October. The size is not depending on the capturing time. It depends on saving format images. The sizes of the last two folders (189.5 and 309 MB) are smaller than others (1.3, 3.1 and 4.6 GB) because they use GIF format instead of PNG or JPEG. The benchmark dataset often has only one format, but this is raw data collected from the software project. In the early stage, capturing images are stored as PNG or JPEG. After that, they are stored as GIF in other to reduce storage space.

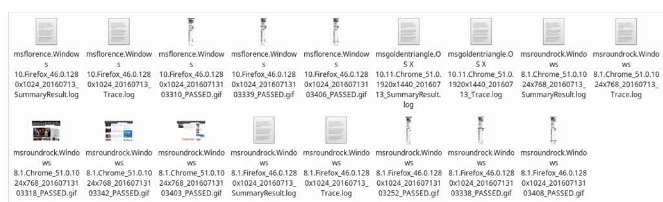


Fig. 3. Sample details of folder July 201607

Figure 3 shows some sample images and logs file in the folder 201607. In this folder have some log files and some capture images cross operating systems with multiple browsers such as Windows, Linux, macOS with some common browsers IE, Firefox, Chrome.

Due to the mixed raw dataset, we must normalize raw data to be standard input 32x32 of CNN model as a pre-processing component. Firstly, pre-processing component removes all log files. Secondly, it resizes all image to 32x32. Finally, making images to binary files in order to create batch files is the standard input for CNN system customized by ConvNetJS.

III. CNN MODEL FOR ADVERTISEMENT CLASSIFICATION

The overall of CNN model for advertisement images shows in the below diagram with four blocks (input, capturing, classification, output).

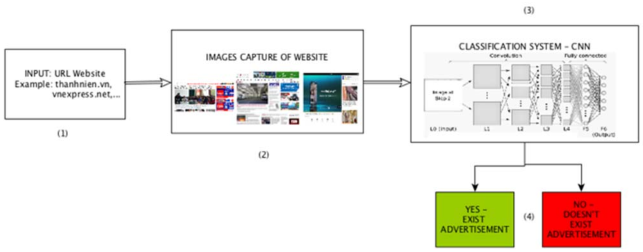


Fig. 4. The full system for advertisement image classification using CNN Model.

Figure 4 shows full system for advertisement image classification. The input of the system is URL of the website which needs to check the display of the advertisement. Then the system captures the web pages as screenshots and saves them as images with PNG format. After that, these images are resized to the same standard size (32x32 pixels) in order to be easily processed by CNN. The output of the system is one of two conclusions {YES, NO}.

YES means that the advertisements display clearly. "Clear" means that users are able to see the content of advertisements. For example, please ref to the figure 2a and 2b.

NO means that the advertisements do not display or not clear. In the first case, "Not clear" means that the content of advertisements does not display. In the second case, "Not clear" means that the content of advertisements displays but users are unable to identify the content of advertisements.

The main problem is to find a suitable number of layer (nL) and a number of the filter (mF) in each conv layer of CNN model for advertisement image classification. Therefore, we call adapted CNN model for advertisement image classification nLmF-CNN. nL, mF parameters are identified by experiments.

Where:

n is the number of layers Convolution (conv) and pool.

$$n \in N^{+}, n > 0$$

m is number filters in layer conv.

$$m = 2^{k}, k \geq 0$$

For example, *2L2F*-CNN means that CNN model has 2 layers conv and pool and in each conv layer use 2 filters. One more example, *3L4F*–CNN means that CNN model has 3 layers conv and pool, in each conv layer use 4 filters.

This paper uses ConvNetJS to develop the nLmF-CNN system. More details see the following figure below.

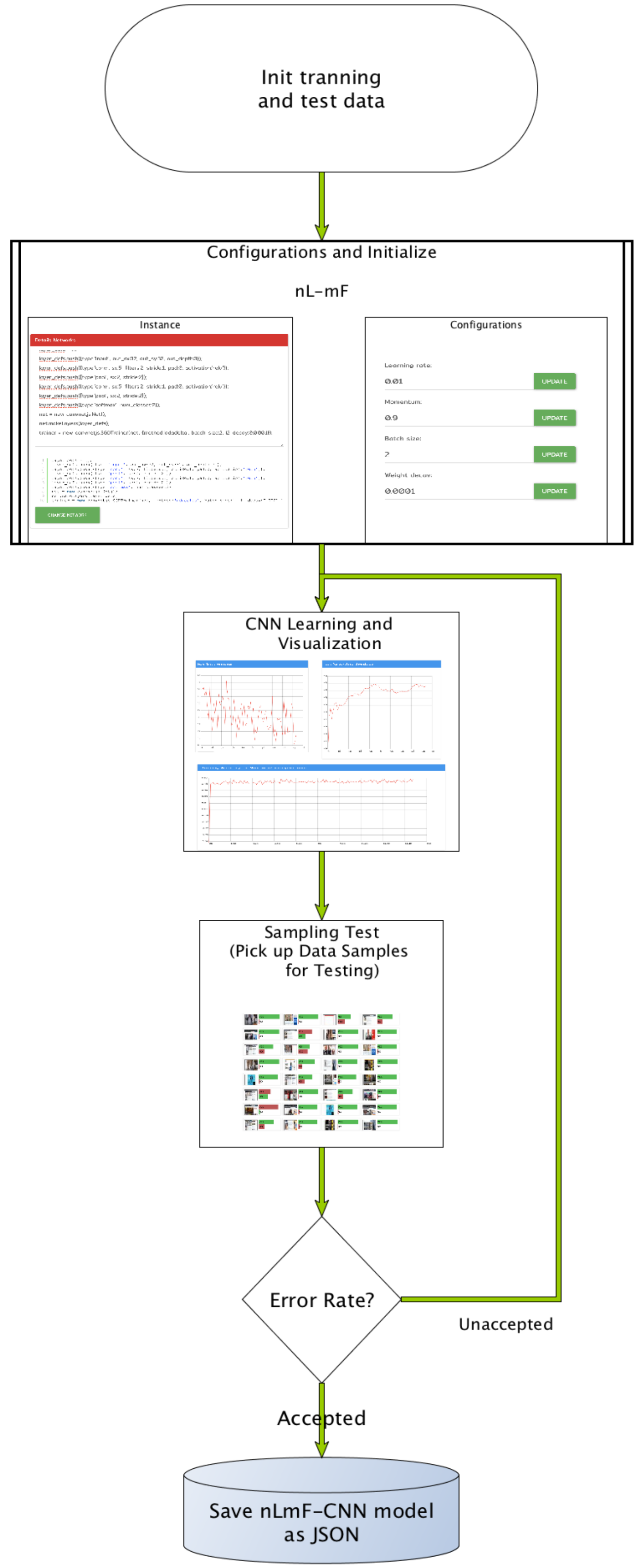


Fig. 5. The processing diagram of the nLmF-CNN model.

Figure 5 shows the architect of the nLmF-CNN system. In this picture have many components, with three main processing components: configurations and initialize, CNN Learning and Visualization and pick up Sample Test.

Configurations and Initialize component has two inputs: instance and configurations. Firstly, the instance describes the number of layers, names of layers, the number of filters on each conv layer, the classification method, size of input images, the kernel of convolution layers. Lastly, the configuration describes some parameters for the model such as learning-rate, mini-batch, weight decay, momentum.

CNN Learning and Visualization component of nLmF-CNN describes features of images in many layers such as input, conv, relu, pool, fc, and softmax layers. It visualizes the current result of learning status.

Pick up Sample Test component calculates the classification error rate. This component selects random samples instead of all samples to test. It is helpful for machine learning with CPU to reach out real-time processing without GPU. The real-time processing is one of software requirements of the advertisement image classification system.

A. *Convolutional Neural Network Visualization*

Fig. 6. Input Visualization

The left part of figure 6 shows that the size of input image 32x32. The right part of figure 6 shows the input image with activations.

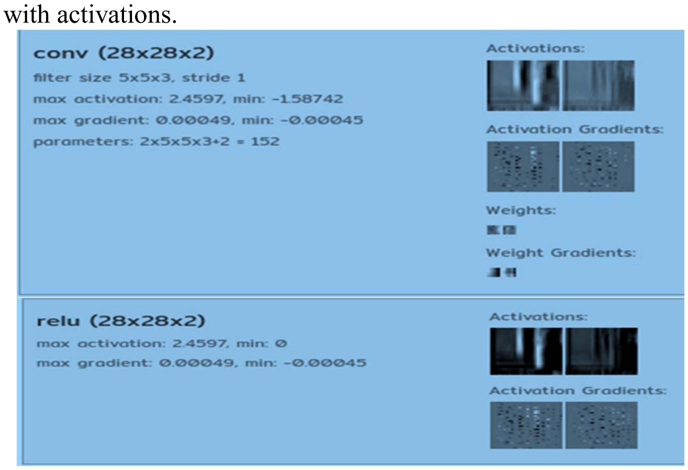


Fig. 7. Convolution Layer Visualization

Figure 7 shows data processing of convolution layer and relu layer. Using the input size 32x32 like figure 7 with kernel 5x5, the conv layer size is 28x28 pixels. The relu layer using max method uses kernel 5x5 to get features which present in the right part of figure 7.

Fig. 8. Visualization of the pool layer with the gradient function.

According to figure 8, the data processing of pool layer creates the output with size 14x14 and visualizes the results in the right part. The output data of conv, relu and pool layers are used as the input of the next layer (the second conv and pool)

The output data of second conv, relu and pool layers are used as the input of the next (the third) layer. The output data of the last layers are used as the input of softmax layer.

Softmax layer uses a classification technique in order to give the final classification conclusion. The classification output has two classes: YES/NO or clear/not clear.

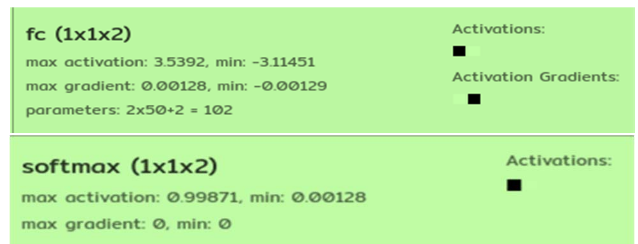


Fig. 9. Softmax layer visualization

Figure 9 shows that the upper part is Fully Connection (fc) Layer and the lower part is softmax layer of the nLmF-CNN model.

B. *Sampling Test*

The aim of sampling test is to improve the performance of training phase. It picks up some sample test data to calculate accuracy. In our experiment environment, we set 200 to the number of sampling test images.

Fig. 10. A part of output results.

Figure 10 shows a part of output results. Each image in test data gets value YES or NO. YES is green and NO is red.

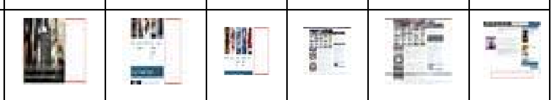
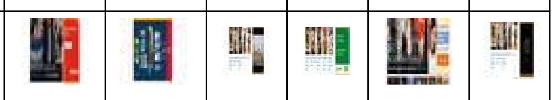
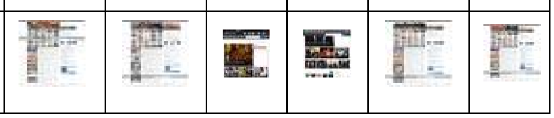
In general, there are able training methods for both nLmF-CNN and traditional CNN models such as Stochastic gradient descent (sgd)[6], Momentum, adagrad[7], adadelta[3]. Adagrad and Adadelta (an Adagrad improvement method) automatically adapt to the learning rate. That is result why in this paper we suggest using Adadelta method.

IV. EXPERIMENTAL RESULT AND DISCUSSION

For our experiments, we use k-fold cross-validation with k=2. It means that the train and the test data are the same sizes. We have transferred raw data to standard input data for the nLmF-CNN model. The standard input consists 2 batches train and test. Train and test batches have the same 1000 online capturing images with size 32x32. Train and test batches also have 205 NO label images which do not include advertisements and 795 YES label images which include at least one advertisement.

This research uses Xubuntu 16.04 LTS operating system using Nginx Server. All the following experiments were performed on a normal laptop, Dell Inspiron N5010, CPU: Core I5 460M - 2,53 GHz, Ram: 6GB, SSD: 240GB. For both accuracy and computation time of training performance, we use the same dataset and the same environments.

TABLE II. Some Images in Train and Test Batches

Test – 'Yes' Label	
Test – 'No' Label	
Train – 'Yes' Label	
Train – 'No' Label	

According to Karpathy Andrej [2], we fix some variables in the nLmF-CNN model in experiments like bellow:

Learning Rate set at 0.01

Momentum set at 0.9

Weight decay set at 0.0001

Method for training is Adadelta [3]

Filter mask size set 5x5

Therefore, we need to identify the best number of layers and number of filters depending on the context of the problem. It means that we need to identify the most suitable parameters (n, m).

$n \in \{1,2,3,4\}$, because the output of advertisement classification is only two classes {YES, NO}.

$m = 2^k$, with k=1..6 so that $m \in \{2,4,8,16,32,64\}$.

The results in figure 11 were computed on the advertisement dataset for the nLmF-CNN model. A pair parameter (n, m) is chosen by the max accuracy.

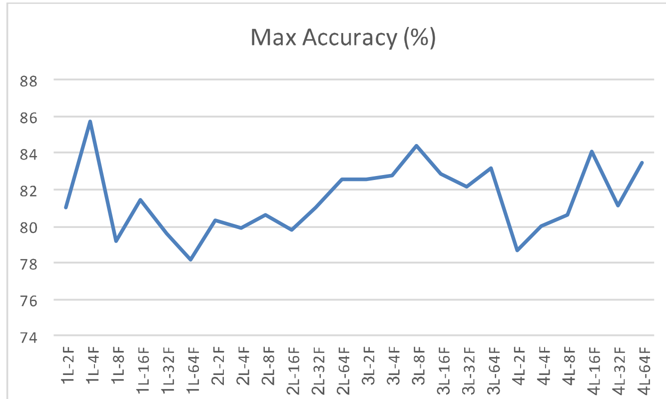


Fig. 11. The max – accuracy results of nLmF-CNN model with different parameters (n, m) in online advertisement images dataset.

Figure 11 shows max accuracy for each couple n and m in *nL-mF CNN* model. In vertical axis is name of *nL-mF*: start with 1L-2F, 1L-4F, 1L-8F, 1L-32F, 2L-2F, 2L-4F, 2L-8F, 2L-16F, 2L-32F, 2L-64F, 3L-2F, 3L-4F, 3L-8F, 3L-16F, 3L-32F, 3L-32F, 4L-2F, 4L-4F, 4L-8F, 4L-16F, 4L-32F, 4L-64F. In horizontal axis shows number for max accuracy using percent (%) unit, with lowest value is 74 and highest value is 88.

According to the results in the figure 11, the best max-accuracy is 1L-4F with 85.74%, the worst max-accuracy is 4L-2F with 78.66%, the second max-accuracy is 3L-8F with 84,38%. Thus, the best parameter to get the highest accuracy is (n=1, m=4).

TABLE III. The Confusion Accuracy Matrix of nLmF-CNN.

m filter(s) / n layer(s)	**2**	**4**	**8**	**16**	**32**	**64**
1	81.01	**85.74**	79.17	81.47	79.64	78.19
2	80.36	79.9	80.63	79.84	81.03	**82.6**
3	82.56	82.74	**84.38**	82.84	82.2	83.21
4	78.66	80	80.64	**84.07**	81.1	83.53

Table III shows preference results of the comparison of multiple (n, m) values. A table cell with a yellow background gets the best accuracy result per layer. A table cell with red color content in table III is a compact result to find the best accuracy in per filter. When m filter increases from 2 to 64, n down (3->1, 4->3) and up (1->3, 3->4, 3->4) continuously.

Using one layer, four filters get the best accuracy. Using two layers, 64 filters get the best accuracy. While using three layers, 8 filters get the best accuracy. Finally, using four layers, 16 filters get the best accuracy. The dependence of n and m shows as the graph bellow.

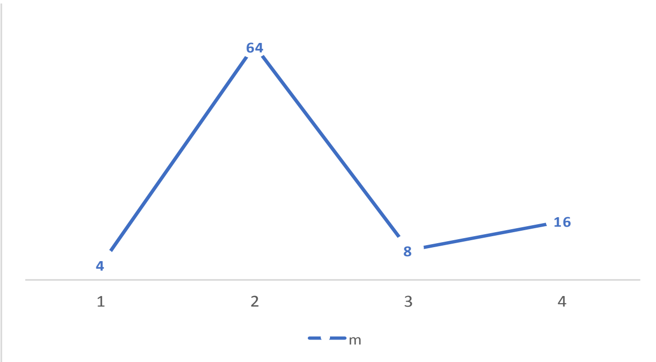


Fig. 12. The dependence of n and m.

The results in figure 13 shows the average comparative using training time (seconds).

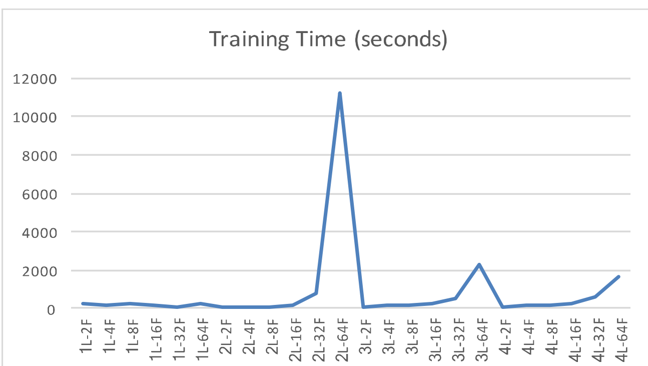


Fig. 13. The Comparative of average training time (seconds) on our dataset.

Figure 13 shows training time for 1000 samples with each couple n and m in the nLm-CNN model. In vertical axis, (n, m) starts with 1L-2F and ends with 4L-64F. The horizontal axis shows training time (seconds unit) up to 12000.

According to figure 13, the best training time is 4L-2F with 50.25 seconds for 1000 samples, the worst training time is 2L-64F with 11232.24 seconds for 1000 samples, the second training time is 3T-2F with 57.1 seconds for 1000 samples and the 1T-4F got training time in 140.64 seconds for 1000 samples.

Basically, the nLmF-CNN model for classifying advertisement images has the best the accuracy more than 86% and the training time for 1000 samples is only 140.64 seconds. It satisfies the accuracy and performance requirements of the advertisement image classification module.

V. CONCLUSION

In this research, we proposed an nLmF-CNN model improved from traditional CNN model and developed from ConvNetJS. Where n is a number of layers Convolution (conv) and pool, and $m=2^k$ is number filters in layer conv.

We applied and found the suitable architecture of the nLmF-CNN model for advertisement image classification, which is two classes (YES, NO) images classification problem. The accuracy results 85.74% showed the feasibility of the proposed model. The training time is acceptable with CPU system instead of GPU.

The open challenge of using our proposed model is how to identify the suitable pair parameter (n, m) for the specific dataset. In this research, we identify (n, m) by experiments.

REFERENCES

[1] A. Krizhevsky, I. Sutskever and G.E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, pp. 1097-1105, 2012.

[2] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang and Y. Gong, "Locality-constrained linear coding for image classification," In Computer Vision and Pattern Recognition (CVPR), IEEE Conference on, 2010, pp. 3360-3367.

[3] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," In Proceedings of the fourteenth international conference on artificial intelligence and statistics, 2011, pp. 215-223.

[4] K. P. Bennett, and A. Demiriz, "Semi-supervised support vector machines," In Advances in Neural Information processing systems, 1999, pp. 368-374.

[5] A. Karpathy, "ConvNetJS: Deep learning in your browser," http://cs. stanford. edu/people/karpathy/convnetjs, 2014.

[6] L. Bottou, "Stochastic gradient learning in neural networks," Proceedings of Neuro-Nımes, 1991, vol. 91, no. 8.

[7] M.D. Zeiler, "ADADELTA: an adaptive learning rate method," arXiv preprint arXiv:1212.5701, 2012.

[8] H.S. Tran, T.H. Le, and T.T. Nguyen, "Facial expression classification using artificial neural network and k-nearest neighbor," International Journal of Information Technology and Computer Science (IJITCS), 2015, vol. 7, no. 3, pp. 27-32.

[9] Y. Lecun, L.Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, 1998, vol. 86, no. 11, pp. 2278-2324.

[10] H.S. Tran, T.H. Le and T.T. Nguyen, "The Degree of Skin Burns Images Recognition using Convolutional Neural Network." Indian Journal of Science and Technology, 2016, vol. 9, no. 45.

[11] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "System and method for generating training cases for image classification," U.S. Patent, 2016, no. 9.

[12] A.T. Vo, T.Q. Le, H.S. Tran, and T.H. Le, "A Fusion TLD and CMT Model for Motion Object Tracking," The International Conference on Information, System and Convergence Applications, July 13-16 in Chiang Mai, Thailand, ISSN 2383-479X, 2016, vol.3 no.1, pp. 60-63.

[13] D. D. Le, T. D. Mai, S. I. Satoh, T. D. Ngo, and D. A. Duong, "Efficient Large Scale Image Classification via Prediction Score Decomposition," In European Conference on Computer Vision, Springer International Publishing, 2016, pp. 770-785

[14] N.V. Do, and H.D. Nguyen. "A Reasoning Method on Computational Network and Its Applications." In Proceedings of the International MultiConference of Engineers and Computer Scientists, 2011, vol. 1.