NU-ResNet: Deep Residual Networks for Thai Food Image Recognition

Chakkrit Termritthikun and Surachet Kanprachar
Department of Electrical and Computer Engineering, Faculty of Engineering,
Naresuan University, Phitsanulok 65000, Thailand.
surachetka@nu.ac.th

***Abstract*—To improve the recognition accuracy of a convolutional neural network, the number of the modules inside the network is normally increased so that the whole network becomes a deeper network. By doing such, it does not always guarantee that the accuracy will be improved. In addition, adding more modules to the network, the required parameter size and processing time are certainly increased. These then result in a significant drawback if such network is utilized in a smartphone in which the computational resources are limited. In this paper, another technique called Identity mapping, which is from the Residual networks, is adopted and added to the network. This technique is applied to the Deep NU-InNet with a depth of 4, 8, and 12 in order to increase the recognition accuracy while the depth is kept constant. Testing this proposed network; that is, NU-ResNet, with THFOOD-50 dataset, which contains various images of 50 Thai famous dishes, the improvement in terms of the recognition accuracy is obtained. With a depth of 4 for NU-ResNet, the achieved Top-1 accuracy and Top-5 accuracy are 83.07% and 97.04%, respectively. The parameter size of the network is only 1.48×10^6, which is quite small for being used with a smartphone application. Moreover, the average processing time per image is 44.60 ms, which can be practically used in an image recognition application. These results show a promising performance of the proposed network to be used with a Thai food image recognition application in a smartphone.**

***Index Terms*—Deep Learning; Food Recognition; Convolutional Neural Network; Residual Networks; Smartphone; Thai Food.**

I. INTRODUCTION

Deep learning [1, 2] is one of the successful algorithms used for pattern recognition and image recognition. With deep learning algorithm, higher recognition accuracy is obtained comparing to those from the preceding techniques used in the past. Additionally, the learning process of deep learning technique is quite suited for large amount of data. This is possibly done by using the artificial neural network (ANN). ANN is applied to different layers; for example, convolutional layer, pooling layer, and fully-connected layer, and so on. When these layers are cascaded in series, a network called a convolutional neural network (CNN) is formed. CNN is one of the promising algorithms used in deep learning. It has been used for analyzing photos so that computers can comprehend and be able to classify the similar photos into the same class. This is wisely done by using supervised learning.

CNN has been continuously developed. In 2015, the Residual network (ResNet) [3] was proposed by Microsoft Research Asia (MSRA). ResNet was the winning in the competitions the ImageNet Large Scale Visual Recognition Challenge (ILSVRC 2015) in the year 2015 [4] and the Common Objects in Context (MSCOCO 2015) in the year 2015 [5]. The idea of Identity mapping was proposed. This is done by inserting shortcut connections in order to add the data from the preceding layer (or from the input) to the data found at the output. By doing this, the data to be used in the following layer will be in the same dimensions; thus, the residual representations of the analyzed data are kept. Additionally, in ResNet, batch normalization (BN) is adopted just like in Inception-v2 [6]. This BN layer is added after convolutional layer and before activation layer. With these two new processes added, the recognition accuracy and the convergence speed are improved. Note that there are 3 different values of the deepness of ResNet; that is, 50, 101, and 152 layers. The one that has won the ILSVRC 2015 is ResNet-152.

NU-InNet [7] utilized the idea of the Inception module from GoogLeNet to adjust and design a new structure of CNN called NU-Inception module to be suited for image recognition of THFOOD-50 dataset, which contains 50 famous kinds of Thai food photos. Moreover, this network was designed in order to use with a smartphone in which the processing time and model size are the key concerned factors. It was found that with NU-InNet, the obtained performance in terms of recognition accuracy is in the same level as that from GoogLeNet while the processing time and the model size is significantly reduced by the factors of 2 and 10, respectively. To further improve the accuracy of the NU-InNet, Deep NU-InNet [8] was proposed. The model size was kept in the range that is suitable for being used with a smartphone. The number of NU-Inception module was increased in each level of the image; that is, 56×56, 28×28, 14×14, 7×7, so that a more detail can be scrutinized. Three levels of depth were used; that is, 4, 8, and 12 (that is, with N = 1, 2, and 3, respectively). With a depth of 4, an increase of 6.54% in terms of the recognition accuracy was obtained.

In this work, both NU-InNet and Deep NU-InNet are further improved in terms of the recognition accuracy. The technique used in ResNet is adopted; that is, a shortcut connection linked between the input and the output of the NU-Inception module is added. The same basic structure used in Deep NU-InNet is kept; that is, BN and MSRA Initialization [9]. The performance in many aspects between this new proposed network is compared to those of NU-InNet and Deep NU-InNet.

The organization of this paper is done as follow. In section 2, the related technology will be shown. The description of the proposed network will be given in section 3. And, in section 4, the testing results from applying this proposed network to THFOOD-50 dataset will be shown and discussed. Also, the comparisons between networks will be given.

Finally, the research work is summarized in the last section.

II. Related Technology

A. Deep NU-InNet [8]

NU-InNet was initially developed by modifying CNN so that it is suitable for using with a smartphone. NU-InNet was then focused on reducing the processing time and model size, which are very important and limited in smartphone applications. In the design, there are two versions of NU-InNet; that is, NU-InNet 1.0 and NU-InNet 1.1, as shown in Figure 1.

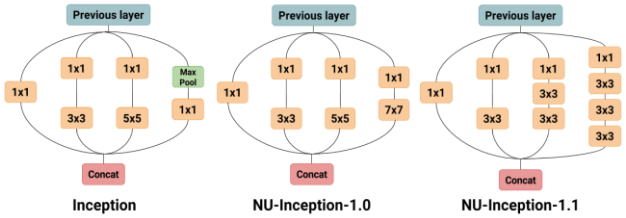

Figure 1: Inception and NU-Inception modules [7]

From Figure 1, for NU-InNet 1.0, the inception module of GoogLeNet was modified. The max pooling and 1×1 convolutional layers were replaced by 1×1 and 7×7 convolutional layers. And, for NU-InNet 1.1, the 5×5 and 7×7 convolutional layers in NU-InNet 1.0 were replaced by two 3×3 convolutional layers and three 3×3 convolutional layers, respectively. After testing these two models with THFOOD-50 data set, it was found that the performance in terms of recognition accuracy, processing time, and model size were superior to those of AlexNet [2] and GoogLeNet [10]. The obtained performance has confirmed that NU-InNet is suited for being used in a smartphone.

In NU-InNet, only one NU-Inception module was used. Applying more NU-Inception modules (that is, 4, 8, and 12 modules) in a model, Deep NU-InNet was developed [8]. The accuracy was meaningfully improved; that is, an increase of 6.54% of accuracy was obtained.

B. Residual Networks [3, 11]

Residual networks or ResNet are the model of CNN developed by MSRA in order to help solving the problem in learning process that affects the recognition accuracy. Such problem normally happens in a deep neural network; for example, the deep convolutional neural network (DCNN) [12]. It was shown that a CNN with 18 layers resulted in a better recognition accuracy than that with 34 layers [3]. However, as the Identity mapping idea (as shown in Figure 2) was added to the model.

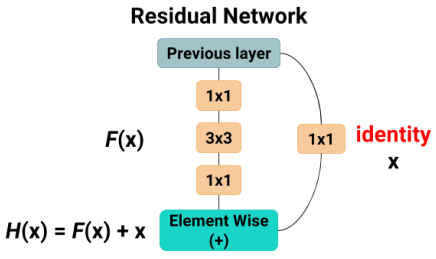

Figure 2: Residual network model

From Figure 2, it is seen that a shortcut connection is inserted and links between the previous layer and the element wise layer. By doing this, the information from the previous layer called x is added to the output from process in the module called F(x); thus, the actual output from the module, H(x), becomes F(x) + x. To have this done properly, the dimension of x has to be adjusted to be identical to the dimension of F(x). After having this done, the accuracy obtained from ResNet-34 is better than that of ResNet-18.

It is seen that the problem of having the deeper network on the accuracy is solved by using the identity mapping. If BN is also used, the training speed or the convergence can be improved since the internal covariate shift problem between the training and testing sets is lessened. Additionally, the accuracy problem from having a deep CNN networks is decreased.

III. Proposed Network Architecture

In this section, the proposed network architecture is explained. As discussed previously that in this work, the identity mapping is applied with NU-InNet 1.0/1.1 in order to improve the recognition accuracy. As shown in Figure 3, the input data is separated into 2 paths. On the first path, the input data is passed through an NU-Inception module, which can be chosen to be either version 1.0 or version 1.1. Then, it will be sent to 1×1 convolutional and BN layers, respectively. For the other path, the input data is sent to the 1×1 convolutional layer and then the BN layer. The dimension of the output from this path is modified to be the same dimension as obtained from the NU-Inception module. Then, the outputs from these two paths are combined at the Addition block; that is, the element wise layer. The output from this block will then be sent to the activation layer, at which in this work, ReLU [13] is adopted.

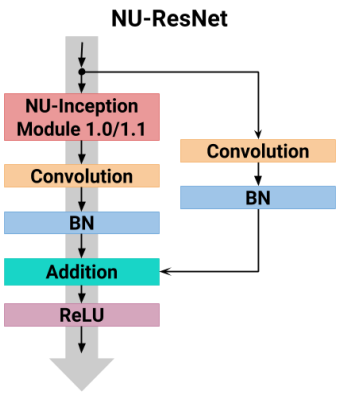

Figure 3: Proposed network architecture

The proposed network is used with NU-InNet [7] by adding a shortcut block to be used with the image level of 12×12 since, at that level, NU-Inception module is applied. Additionally, the proposed network is applied to the Deep NU-InNet [8], which was developed from NU-InNet by adding more NU-Inception modules to each level of image; that is, 56×56, 28×28, 14×14, and 7×7, so that CNN can analyze the considered image in a more detail manner. There are three values of deepness to be studied; that is, 4, 8, and 12 (or N = 1, 2, and 3) as shown in Figure 4. Note that in Figure 4, the deep NU-InNet 1.0 and deep NU-ResNet 1.0 are shown. However, the deep NU-InNet 1.1 and deep NU-ResNet 1.1 will also be studied in this work.

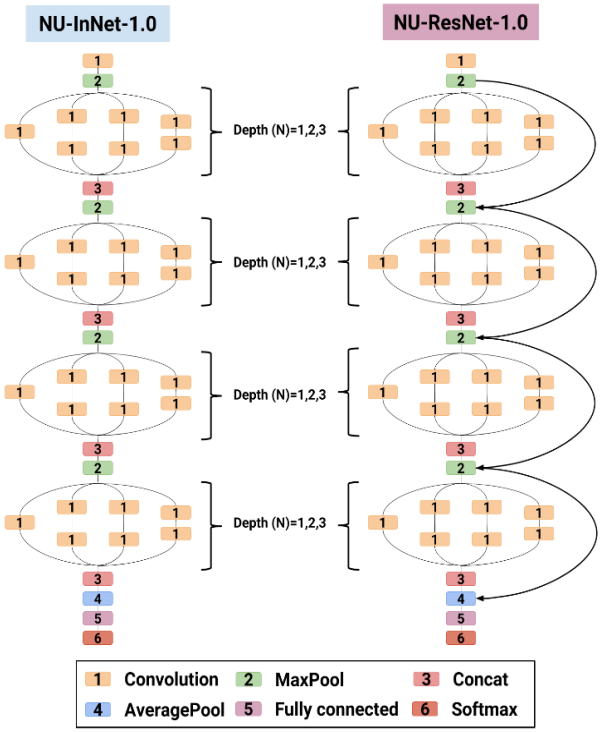

Figure 4: Deep NU-InNet and deep NU-ResNet architectures

IV. Results and Discussion

The proposed network is trained with a high-performance computer consisting of Inter(R) Xeon(R) E5-2683 v3 @2.00GHz 56-core CPU, 64 GB RAM, and NVIDIA Tesla K80 GPU. The operating system is Ubuntu Server 14.04.5 with an installed Caffe [14], which is a deep learning framework for developing CNN with C++. And, the training process is done from scratch with a dataset called THFOOD-50 containing 15,770 images of 50 famous Thai dishes. Some examples of these images are shown in Figure 5.

Figure 5: Examples of Thai food images from THFOOD-50 dataset

The proposed network is tested via 10-fold cross validation by assigning 90% of the dataset to be the training set and the rest 10% of the dataset to be used as the testing set. All images are adjusted the size to be 256×256 pixels. And, the data is pre-processed by center cropping to be 224×224 pixels, horizontal flip, and per-pixel subtraction [2]. The following hyper-parameters are used; that is, Stochastic Gradient Descent (SGD) solver [15], mini-batch size of 64, learning rate starting at 0.1 and reduced in every 25 epochs by 1/10, weight decay of 0.0005, and epoch size of 100.

The performance in terms of the number of parameters, the average forward-backward time, Top-1 accuracy, and Top-5 accuracy, is determined. Additionally, the training speed or the convergence is considered. First, the comparisons between NU-InNets (with versions 1.0 and 1.1) and NU-ResNets (with versions 1.0 and 1.1) are studied. Then, the deep NU-InNets and the deep NU-ResNets for different values of depth are considered.

The results of testing NU-InNets and NU-ResNets with the THFOOD-50 dataset is shown in Table 1. From Table 1, the recognition accuracies; that is, Top-1 and Top-5 accuracies for NU-ResNet 1.0 are 72.13% and 93.90%, respectively. These accuracies are higher than those from NU-InNet 1.0; thus, an improvement in terms of the accuracy is obtained when NU-ResNet is adopted. Similarly, considering NU-InNet 1.1 and NU-ResNet 1.1, it is seen that an improvement in terms of the recognition accuracy is obtained. For example, an increase of approximately 1.5% for Top-1 accuracy is gained. The increase in the recognition accuracies is the result from adding the identity mapping concept the NU-ResNets. Considering the parameter size and the average forward-backward time required, it is seen that NU-ResNets require slightly larger numbers for these two factors. However, the required parameter size and average forward-backward time by NU-ResNets are still in the range that is applicable for being used with a smartphone. Thus, the increase in terms of these factors is still acceptable.

Table 1
Performance of NU-InNets and NU-ResNets: Tested with THFOOF-50 Dataset

Model	Depth	Parameter size (x10^6)	Avg. Forward-Backward Time (ms/image)	Avg. Accuracy (%)	
				Top-1	Top-5
NU-InNet 1.0 [7]	1	0.43	12.54	70.14	93.22
NU-InNet 1.1 [7]	1	0.44	20.03	74.24	94.53
NU-ResNet 1.0	1	0.52	14.01	72.13	93.90
NU-ResNet 1.1	1	0.54	21.95	75.73	95.02

The training speed or the convergence of 4 networks is shown in Figure 6. The green dash-dotted and red dashed lines represent the Top-1 accuracy of NU-InNet 1.0 and 1.1, respectively. The pink dotted and blue solid lines represent the Top-1 accuracy of NU-ResNet 1.0 and 1.1, respectively. From these 4 lines, it is seen that the training speed from both NU-ResNets are faster than those from NU-InNets. For example, to reach Top-1 accuracy of 60%, NU-InNet 1.0 and 1.1 require 12 and 9 epochs, respectively while NU-ResNet 1.0 and 1.1 require 9 and 6 epochs, respectively. The increase in the training speed of NU-ResNet is the result of applying BN to the network.

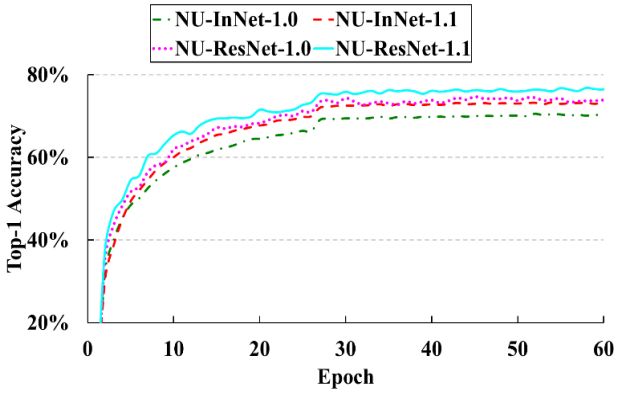

Figure 6: Top-1 accuracy vs. number of epochs; for NU-InNets and NU-ResNets

Next, the performance of deep NU-InNets and deep NU-ResNets for different depth values after testing with the THFOOD-50 dataset is shown in Table 2. From Table 2, the performance for deep NU-InNets and NU-ResNets is shown. Additionally, the performance of ResNet-50 is given. For ResNet-50, it is seen that its performance in all 4 considered factors is worse than that of NU-InNets and NU-ResNets. The performance of ResNet-50 is put in the table so that the superiority of NU-InNets and NU-ResNets can be viewed.

Table 2
Performance of Deep NU-InNets and Deep NU-ResNets: Tested with THFOOD-50 Dataset

Model	Depth	Parameter size (x10^6)	Avg. Forward-Backward Time (ms/image)	Avg. Accuracy (%)	
				Top-1	Top-5
ResNet-50 [3]	-	23.61	104.56	72.88	93.71
NU-InNet 1.0 [8]	4	0.95	37.66	79.68	96.00
NU-ResNet 1.0	4	1.48	48.23	82.58	96.96
	8	2.89	83.24	81.61	96.43
	12	4.29	117.78	80.77	96.30
NU-InNet 1.18	4	0.94	40.80	80.34	96.27
NU-ResNet 1.1	4	1.48	44.60	83.07	97.04
	8	2.88	78.33	82.59	96.80
	12	4.27	112.63	81.81	96.62

Comparing the deep NU-InNet 1.0 (with the depth of 4) to the deep NU-ResNets 1.0 (with the depths of 4, 8, and 12), it is seen that Top-1 accuracies obtained from all deep NU-ResNets are higher than Top-1 accuracy from the deep NU-InNet. The highest Top-1 accuracy is achieved from the deep NU-ResNet 1.0 with the depth of 4; that is, 82.58%. An increase of 2.90% for Top-1 accuracy is gained when compared to Top-1 accuracy from the deep NU-InNet 1.0 with the depth of 4. The parameter size and the average forward-backward time required by the deep NU-ResNet 1.0 with the depth of 4 are 1.48×10^6 and 48.23 ms/image, respectively. These two parameters are higher than those from the deep NU-InNet 1.0 with the depth of 4. However, the parameter size and the average forward-backward time required by the deep NU-ResNet 1.0 with the depth of 4 are not too large and still in the range that can be used practically with a smartphone.

Considering the deep NU-InNet 1.1 (with the depth of 4) and the deep NU-ResNet 1.1 (with the depths of 4, 8, and 12), it is seen that Top-1 accuracy from the deep NU-InNet 1.1 is less than that of the deep NU-ResNets 1.1. The highest Top-1 accuracy is obtained from the deep NU-ResNet 1.1 with the depth of 4; that is, 83.07% is obtained. With an increase of 2.63% in Top-1 accuracy compared to that of the deep NU-InNet 1.1 with the depth of 4, the accuracy improvement gained from the proposed network is clearly shown. Similar to the case of deep NU-ResNets 1.0, for deep NU-ResNets 1.1, the required parameter size and the average forward-backward time are increased compared to those of the deep NU-InNet 1.1 with the depth of 4. However, for the deep NU-ResNet 1.1 with the depth of 4, where the best Top-1 accuracy is obtained, the values of these two performance factors are 1.48×10^6 and 44.60 ms/image, respectively, which are in the acceptable range for being used in a smartphone.

Comparing only Top-1 accuracy, it is seen that the best values are from the deep NU-ResNet 1.1 (with a depth of 4) and the deep NU-ResNet 1.0 (with a depth of 4); that is, 83.07% and 82.58%, respectively. As explained, the accuracy improvement is clearly achieved from the proposed network.

Figure 7 shows the training speed or the convergence of 4 deep neural networks; that is, NU-InNet 1.0, NU-InNet 1.1, NU-ResNet 1.0, and NU-ResNet 1.1, all with a depth of 4. The pink dotted and green dashed lines represent the Top-1 accuracy of deep NU-InNet 1.0 and 1.1, respectively. The red dash-dotted and blue solid lines represent the Top-1 accuracy of deep NU-ResNet 1.0 and 1.1, respectively. Similar to Figure 6, from these 4 lines, it is seen that the training speed from both deep NU-ResNets are faster than those from deep NU-InNets. For example, to reach Top-1 accuracy of 60%, deep NU-InNet 1.0 and 1.1 require 9 epochs both while both deep NU-ResNet 1.0 and 1.1 require 6 epochs. The superiority in the training speed obtained from NU-ResNets is from adding BN layer to the network. Considering all 4 lines in Figure 7, it is also seen that after 27 epochs, all 4 networks reach their highest Top-1 accuracy, which are already shown in Table 2. Additionally, it is seen that after the 25-epoch, in all cases, the obtained accuracy is suddenly changed to a higher value. This is because of the learning rate that has been set to decrease by 1/10 in every 25 epochs.

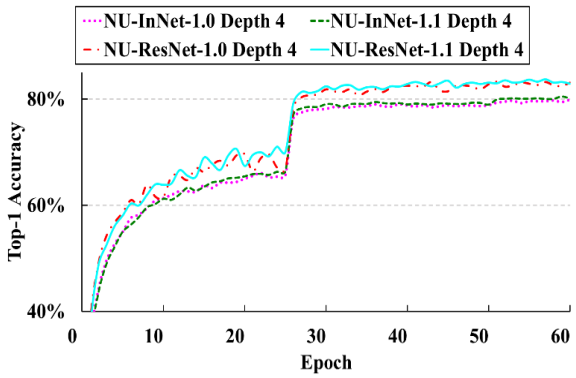

Figure 7: Top-1 accuracy vs. number of epochs; for deep NU-InNets and deep NU-ResNets

The proposed network is also tested with a smartphone to determine the execution time and model size required. The CPU of the smartphone is powered by Qualcomm Snapdragon 615 MSM8916 with a clock speed of 1.5 GHz and 8-core processor. It is also equipped with 2-GB RAM and run with Android 7.1.2 operating system. The tested image size is 1080×1080 pixels. The trained model by each network is uploaded to the smartphone and the tested image is used for

testing the performance of such model. The results of testing the proposed network and others are shown in Table 3.

Table 3
Performance of Deep NU-InNets and Deep NU-ResNets: Tested with THFOOD-50 Dataset

Model	Depth	Execution Time (ms)	Model Size (MB)
ResNet-50 [3]	-	5392	90.3
NU-InNet 1.0 [8]	4	890	3.66
NU-InNet 1.1 [8]	4	996	3.65
NU-ResNet 1.0	4	1169	5.69
NU-ResNet 1.1	4	1220	5.68

From Table 3, it is seen that the performance of testing the proposed networks (that is, NU-ResNet 1.0 and 1.1 (with a depth of 4)), ResNet-50, NU-InNet 1.0 and 1.1 (with a depth of 4) with a smartphone is shown. Considering the execution time, it is seen that ResNet-50 requires the longest time; that is, 5,392 ms while the other 4 networks require less than 2 seconds. Comparing between NU-ResNets (the proposed networks) and NU-InNets, it is seen that the execution time required by the proposed networks is higher than that of NU-InNets; that is, approximately 200 to 300 ms required additionally by the proposed network. However, the execution time required by the proposed network is not too long (that is, less than 1.3 seconds) to be used in practice.

Considering the model size, it is seen that the model size required by ResNet-50 is really large (that is, 90.3 MB) and if being used with a smartphone, it will surely consume a large amount of memory in the smartphone. On the other hand, for NU-ResNets and NU-InNets, it is seen that the model size required by these 4 networks are more than 15 times smaller than the model size required by ResNet-50. The model sizes for NU-InNet 1.0 and 1.1 are 3.66 and 3.65 MB, respectively. And, the model sizes for NU-ResNet 1.0 and 1.1 are 5.69 and 5.68 MB, respectively. Even though the required model size of the propose networks is larger than that for NU-InNets, such model size (that is, 5.69 or 5.68 MB) is still in an acceptable range for being used with a smartphone.

V. CONCLUSION

In this work, in order to improve the recognition accuracy of NU-InNet, the concept of the identity mapping was added. A shortcut connection linked between the input layer and the output layer of the NU-Inception module was inserted so that the residual representation of the analyzed data is kept to the next level of the process. The proposed network called NU-ResNet was tested with the THFOOD-50 dataset, which contains images of 50 famous kinds of Thai food. It was found that the recognition accuracy for Top-1 accuracy, in all cases that NU-ResNet was applied, was higher compared to that from the case of not using NU-ResNet. The best accuracy of 83.07% is obtained from the case of deep NU-ResNet 1.1 with a depth of 4. Additionally, the convergence of the training process was also better; that is, a faster training speed was achieved. It is clearly seen that the proposed NU-ResNet can help improving the recognition accuracy for Thai Food image recognition while the required parameter size and processing time per image are still in the acceptable range for being used with a smartphone.

ACKNOWLEDGMENT

This work was supported by Naresuan University, Thailand.

REFERENCES

[1] Y. LeCun, Y. Bengio, G. Hinton. Deep learning, Nature. 521 (2015) 436-444.
[2] A. Krizhevsky, I. Sutskever, G. E. Hinton. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, (2012) 1097-1105.
[3] K. He, X. Zhang, S. Ren, J. Sun. Deep residual learning for image recognition. 2016 IEEE Conference on in Computer Vision and Pattern Recognition (CVPR), Las Vegas, Nevada, USA, (2016) 770-778.
[4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, et al. Imagenet large scale visual recognition challenge, International Journal of Computer Vision. 115(2015) 211-252.
[5] T.-Y. Lin, et al. Microsoft coco: Common objects in context. Proceedings of European Conference on Computer Vision, (2014) 740-755.
[6] S. Ioffe, C. Szegedy. Batch Normalization: Accelerating deep network training by reducing internal covariate shift. Proceedings of the 32nd International Conference on Machine Learning, (2015) 448-456.
[7] C. Termritthikun, P. Muneesawang, S. Kanprachar. NU-InNet: Thai food image recognition using convolutional neural networks on smartphone. Proceedings of the International Conference on Computer Sciences and Information Technology (COMSIT), Krabi, Thailand, (2016).
[8] C. Termritthikun, S. Kanprachar. Accuracy improvement of Thai food image recognition using deep convolutional neural networks. Proceedings of the 2017 International Congress on Electrical Engineering (iEECON), Pattaya, Thailand, (2017) 645-648.
[9] K. He, X. Zhang, S. Ren, J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. Proceedings of the IEEE International Conference on Computer Vision, (2015) 1026-1034.
[10] C. Szegedy, et al. Going deeper with convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (2015) 1-9.
[11] K. He, X. Zhang, S. Ren, J. Sun. Identity mappings in deep residual networks. Proceedings of European Conference on Computer Vision, (2016) 630-645.
[12] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014.
[13] G. E. Dahl, T. N. Sainath, G. E. Hinton. Improving deep neural networks for LVCSR using rectified linear units and dropout. Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, (2013) 8609-8613.
[14] Y. Jia, et al. Caffe: Convolutional architecture for fast feature embedding. Proceedings of the 22nd ACM international conference on Multimedia, (2014) 675-678.
[15] Z. Akata, F. Perronnin, Z. Harchaoui, C. Schmid. Good practice in large-scale learning for image classification, IEEE Transactions on Pattern Analysis and Machine Intelligence. 36(3) (2014) 507 -520.