JJEE

Volume 10 | Number 4 | December 2024 | Pages 687-698

Jordan Journal of Electrical Engineering

ISSN (print): 2409-9600, ISSN (online): 2409-9619

Homepage: jjee.ttu.edu.jo

Single Sample per Class Classification Using Convolutional Neural Networks

Inad A. Aljarrah[1*], Maad Ebrahim[2]

[1, 2] Computer Engineering Department, Faculty of Computer and Information Technology, Jordan University of Science and Technology, Irbid, Jordan

E-mail: inad@just.edu.jo

Received: Mar 14, 2024 *Revised: May 23, 2024* *Accepted: Jun 01, 2024* *Available online: Sep 15, 2024*

***Abstract*—** Recently, Convolutional Neural Networks (CNNs) have shown a great significance in the field of computer vision and image recognition. However, they have two major difficulties to be addressed. The first one is the need to have too many samples per class for training. It is not always an easy task to have a huge number of labelled samples per class for every problem; actually, in some cases only one sample per class is all what is available. The second difficulty is the enormous computational power needed to perform the training task. In this paper, an experimental study of how to handle problems where the data available provides only one sample per class is carried out. The implemented technique utilizes proper data augmentation in solving the aforementioned problems. A database of the world countries flags with one flag sample per country is used to start with. One sample per class is the worst case for CNN training, but the proposed approach helps in enhancing the accuracy from being unfeasible at the beginning of training to above 99% of validation accuracy when the technique is applied. Promising results have been achieved without the need for a very deep CNN. Also, the findings reveal that the utilized type of data augmentation technique must be carefully selected for each application to avoid over-fitting while obtaining the best validation accuracy. As a solution, an adequate selection of different augmentation options is tried out to improve the network ability to generalize well to new testing samples.

***Keywords*—** Deep neural networks; Convolutional neural network; Classification; Data augmentation.

1. INTRODUCTION

The idea of Convolutional Neural Networks in general was inspired by how our brains work that was discovered in 1959 by Hubel and Wiesel [1] by testing how cats brains react to small visual regions. Until 2012 there was no major enhancement in this field mainly due to the lack of big datasets and the lack in computational power at that period of time. In 2012, a huge breakthrough took place by Krizhevsky et al [2] when they worked on the huge ImageNet [3] dataset provided publicly by Stanford University. And of course, by that time, computational power has increased significantly. Since 2012, researches kept on proving that the major factor in enhancing the network architecture was the depth of the network and the size of the dataset. However, some researches proposed some techniques to minimize the computational power and time needed for training while maintaining similar results of old classical architectures.

One of those techniques was proposed by Ioffe et al [4], they proposed to add a batch normalization layer before every activation layer, they suggest that it reduces the internal Covariate Shift by maintaining a good distribution of values in every layer as the training progress. Another technique was introduced by Google Inception architecture [5] that was

* Corresponding author

DOI: https://doi.org/10.5455/jjee.204-1710412582

recently enhanced [6] using the residual technique by Microsoft [7]. For Google inception model, they suggest using more than one kernel size in the same layer; such as, 1x1, 3x3, and 5x5. And then concatenate their outputs in a depth manor. The idea behind that is to let the network choose the kernel size that is most suitable for the given problem instead of changing that as a hyper parameter. Batch normalization after is used every layer to speed up the training process. The residual connection helped deeper layers reference the values in the input layer and earlier layers. Dropout has also been introduced by Srivastava et al [8] to prevent over-fitting and to make the network generalizes well to new test samples. It works by randomly deactivating a number of the neurons in the fully connected layers during each training step. That makes the network more robust to minor changes in the input image.

Another technique to speed up the training process is to provide the input dataset zero-centered and normalized, which does the same job as batch normalization technique introduced earlier. But for the input layer this time. All the techniques introduced above focused on increasing the accuracy and speeding up the training processing assuming a big enough dataset to train on. However, having professionally maintained labelled datasets is extremely hard. That is mainly why most of the powerful techniques use the ImageNet dataset as bench mark with around 14M images and 21K categories. However, to train an application specific network, it is hard to provide such a huge and accurate dataset for training. Hence, in this paper, several data augmentation techniques are proposed to increase the training and validation accuracy and explain how the combinations of these techniques were able to enhance the network performance. To prove the quality of this work, a dataset of 224 countries' flags with one flag per country is used. This makes it a harder problem for CNN recognition task.

In 2012, Krizhevsky et al [2] has proposed the state of the art results using convolutional neural network to classify the immense ImageNet dataset of 1.2 million images with 1000 classes at that time. They were the first to achieve an accuracy of 62.5% for top 1 predictions and 83% accuracy in the top 5 predictions. That was done with what was considered a deep convolutional network at that time with around 650000 neurons. Due to the lack of computational power at the time, they had to split the network to work on two different GPU devices. That is the reason why their network looks like two different streams of data flow as in Fig. 1.

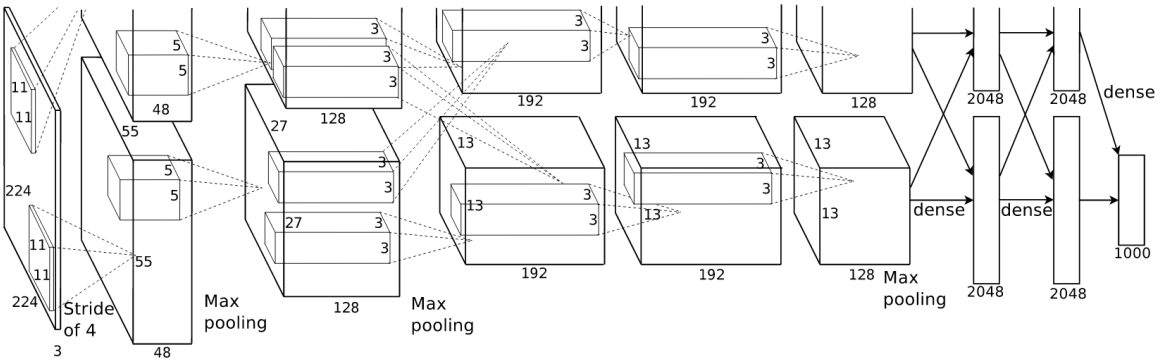

Fig. 1. AlexNet architecture.

It is good to mention that it took them around a week of training time. Since 2012, work in image classification and recognition has moved to convolutional neural networks. Starting from fined tuned versions of AlexNet architecture such as SF NET [9] in 2013 and VGG NET [10] in 2014. And ending with totally new concepts of the convolutional neural networks; such as GoogLeNet [11] and Microsoft ResNet [7] in 2015. GoogLeNet has introduced the idea of

Inception Layers as shown in Fig. 2. Which helps the network choosing the kernel sizes for a given problem by learning different weights for each of the three kernels provided at the same time at each layer.

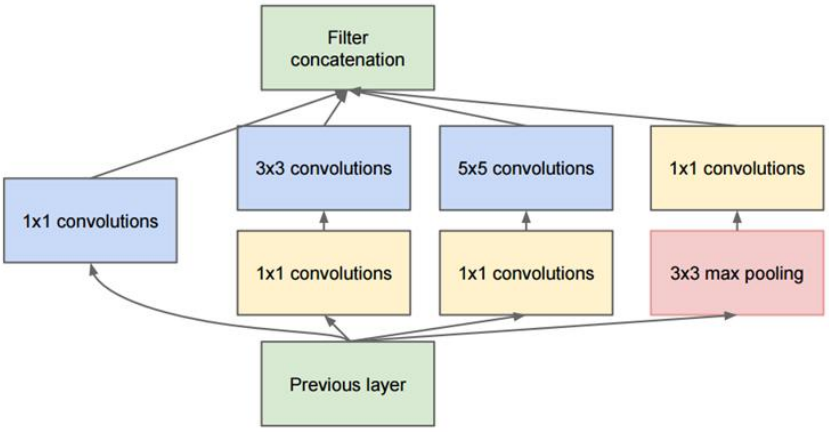

Fig. 2. Inception layer architecture in GoogLeNet.

ResNet introduced the idea of having residual connection from deeper layers back to input and earlier layers in the network as shown in Fig. 3. As can be seen as a common factor between all those previously mentioned architectures that they all need a huge, accurate and professionally labelled images by human. And since this kind of dataset is not always available for many applications, old image recognition techniques are still needed for some specific applications, especially when the dataset is very small. In this case, professional computer vision engineers try to pre-extract the required features for the system to work on.

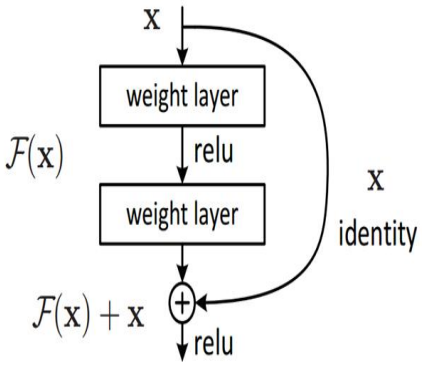

Fig. 3. Residual Block Architecture in ResNet.

One of these feature extraction techniques are called SIFT features [12], it is scale, rotational, translational, and illumination invariant feature extractors. They are inspired by how the neuron architecture in the temporal cortex in the human brain works. Handcrafted features and feature extractors such as SIFT are very powerful in case of very small dataset images, but they are time consuming and very hard to discover. They need expert knowledge in computer vision to get the best features for every specific problem. Hence, a work that combines both the easiness of convolutional neural networks and the small dataset size capabilities of handcrafted features and feature extractors is needed. Here comes the importance of using data augmentation techniques. Although data augmentation is one of the key rules in reducing over-fitting, it is still of a great importance in increasing dataset size when dataset acquisition is hard. For example, data augmentation techniques help to increase the size

of training data for medical-image classification problems, which normally have a small number of samples per class [13-16].

For instance, earlier work on Indirect Immune Fluorescence (IIF) images on Human Epithelial-2 (HEp-2) cells started using LBP [17] and SIFT handcrafted feature extractors. Later works focused on using convolutional neural networks with different data augmentation techniques [18, 19]. In Table 1, Jia et al [20] achieved three significantly different accuracy results with just augmenting the HEp-2 dataset with three different rotation angles. In the Table, ACA (Average Classification Accuracy) is computed as all the correct predictions of the test set over the number of test samples. While MCA (Mean Class Accuracy) is calculated as the summation of the percentages of correct predictions for every class over the number of classes. Since the number of samples of each class is not the same in HEp-2 dataset, the rotation angles differ from one class to another to ensure having as similar number of samples per class as possible after the rotation process. Table 1 shows that the accuracy increases as the rotation angle decrease. Specifically, Rotation Degrees 3 achieves higher accuracy compared to Rotation Degrees 2, which in turn outperforms Rotation Degrees 1. This trend occurs because smaller rotation degrees result in more augmented data.

Table 1. HEp-2 classification performance with data augmentation using three different rotation angles.

Accuracy rotation	Rotation degrees 1	Rotation degrees 2	Rotation degrees 3
ACA	94.74%	98.24%	98.49%
MCA	95.08%	98.05%	98.26%

Jetley et al [21] used Support Vector Machine (SVM) [22] to classify the 224-word flags using the HSI colour representation instead of RGB colour representation. They worked with HSI colour values because they believe that flags are distinguished even by humans with the colour combinations rather than the edges and shapes of the flags. Because one flag sample per country is not sufficient for the classification task, they have manually gathered 18935 images from the Internet and still used scaling data augmentation techniques to make the system scale invariant. Gu et al [23] gathered 1668 positive flag samples from the internet, which are partially blocked, deformed, blurred, and contain complex background and lighting conditions. In this work, however, convolutional neural networks to classify countries' flags from a one sample per class dataset is used. To help the CNN network to capture realistic variations in these samples, several data augmentation techniques suitable for flag datasets are carefully introduced.

2. MATERIALS AND METHODS

The experiment setting for the proposed work is described in Table 2. Due to the limited computational resources, the number of scenarios, Depth of the network, and the amount of data augmentation were limited to get the results within a reasonable amount of time. The nine scenarios/combinations for different data augmentation techniques were tested as listed in Table 3. Each combination consists of choosing between 1, 3, or 5-degrees rotation steps. After deciding on the rotation steps, one or more transformation methods were chosen, i.e., translation, shearing, and scaling. A white background is only used in scenario 5, while the rest use a black background. Finally, noise is added to the images of scenario 5 and 6 to test its effect with white and black backgrounds. The size of the augmented dataset will change based

on the selected set of augmentation techniques, e.g., scenario 9 has the largest number of augmented images because it uses a 1-degree rotation step with the three transformation methods. Each of those scenarios is then tested with three CNN network architectures with different depths.

Table 2. Experimental hardware and software environment.

Item	Description
Device	Dell Inspiron (15-7559) laptop
Processor	Intel(R) core (TM) i7-6700HQ @ 2.60GHz
RAM	16.0 GB DDR-3
Hard desk	128GB SSD SanDisk Z400s
GPU	NVIDIA GeForce GTX 960M
GPU CUDA compatibility	5.0
Cuda cores	640
Operating system	64-bit windows 10
Implementation software	MATLAB R2018b

Table 3. The utilized data augmentation combinations.

Augmentation\Scenario	1	2	3	4	5	6	7	8	9
Rotation 1◦	X								X
Rotation 3◦								X	
Rotation 5◦		X	X	X	X	X	X		
Translation		X			X	X	X	X	X
Shearing			X		X	X	X	X	X
Scaling				X	X	X	X	X	X
Noise (salt & pepper)					X	X			
Black background	X	X	X	X		X	X	X	X
White background					X				
Images per class	91	57	57	57	285	285	465	195	1365

The flags' images used are one sample computer generated flag per country. And there are 224 countries in the dataset. Adding non-flag images, by generating randomly colored, black, or white noisy images, caused a drop in the validation and prediction accuracy of all networks. Hence, classifying non-flag images was neglected and stuck with categorizing the test images into the given countries. Nine scenarios were used to augment the single flag image per class into more images used for training and validation. The number of images for every class is the same after the augmentation process. These images are then divided into two groups, for training and validation, in the ratio 3:1 respectively. The number of generated images per class for every augmentation scenario is listed in Table 3.

The original images are of different sizes. Hence, they were scaled to 100x100 pixels. From MATLAB documentation, augmenting the images using rotation, translation, and shearing is done by cropping the output image to make the output size the same as the input size. To avoid losing some of the details in the original image, a rescale of the output image to include all the input image after the performed augmentation technique is performed, i.e., loose transformation, and then rescaled the output image to be 100x100 after the augmentation process.

Image transformations are done by filling the generated new background with a pre-determined color. Filling the new background with colors other than black and white caused a significant accuracy to drop, which means that the network is considering that color part of

the flag. Hence, only black and white backgrounds were considered. Changing the background color was done by creating a white, i.e., Boolean True, image of the same size as the input image (image to be augmented), with the same augmentations as for the input image. Before saving the augmented input image, the Boolean image is negated so that all background pixels are set to a value of 1, which are at the same locations as the background of the augmented image. Now, the background of the augmented image can be changed using the Boolean image as an index for the pixels in the flag image. Fig. 4 shows the augmented flags with black, colored, and white backgrounds in each row of images, respectively. For images with white background, also salt & pepper noise using density of 0.001 is added. For the rotation Augmentation, rotating each image between 0 and 360 degrees is bad for flag images, because some flags are the flipped version of others either vertically or horizontally. Hence, we only rotate the images between 45 and -45 degrees without the flip augmentation technique. This allowed us to use smaller rotation steps with considerably less training size.

Fig. 4. Different flag backgrounds after augmentation.

MATLAB R2017b has a great support for Deep Neural Networks to build easily customizable network Architectures. It can easily extract the learned parameters from the network to run classification on older versions with simple for loops and matrix multiplications. It is also compatible with NVIDIA GPUs using the CUDA library. In addition, it has built-in support for data augmentation in its input layer, such as random cropping and flipping during training time. However, augmenting input images as they are fed to the network made the performance worst for the single sample per class dataset. Hence, a pre-augmentation was used in this work as it has shown superior performance compared to real-time augmentation of the input images [24].

The training for each scenario and for the three network architectures was done for 7 epochs with a mini-batch-size of 128 samples. Training longer, i.e., 100 epochs, achieves better validation accuracy on the validation set, i.e., a different augmented set of the single sample images. However, it performs bad on Internet flags with different patterns than the original dataset. The learning rate was 0.00001 with validation frequency between 100 and 1000 according to the training set size. Early stopping is also applied with a validation patience value of 5. The training time using our resources was approximately 2 days with the deepest CNN network and 1365 images per class.

3. RESULTS AND DISCUSSION

The first trivial observation is that with less training data (less data augmentation) there will be less validation and training accuracy. That was tested and verified using the shallowest CNN architecture for only 3 epochs of training. The network is trained with an augmented data with 20 degree of rotation step between 0 and 359 degrees. The network validation accuracy reached 54.80% after the final iteration as shown in Fig. 5.

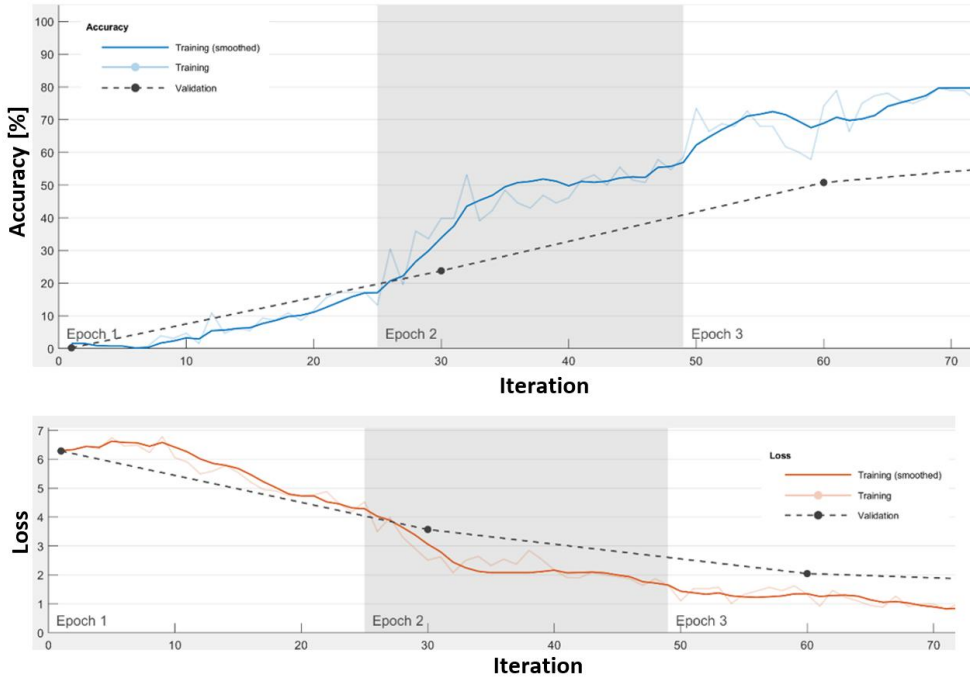

Fig. 5. Validation accuracy achieved using augmentation with only 20-degree rotation.

When the same network was trained with an augmented data using 1 degree of image rotation, the validation accuracy reached 98.84% at the second epoch of training (see Fig. 6).

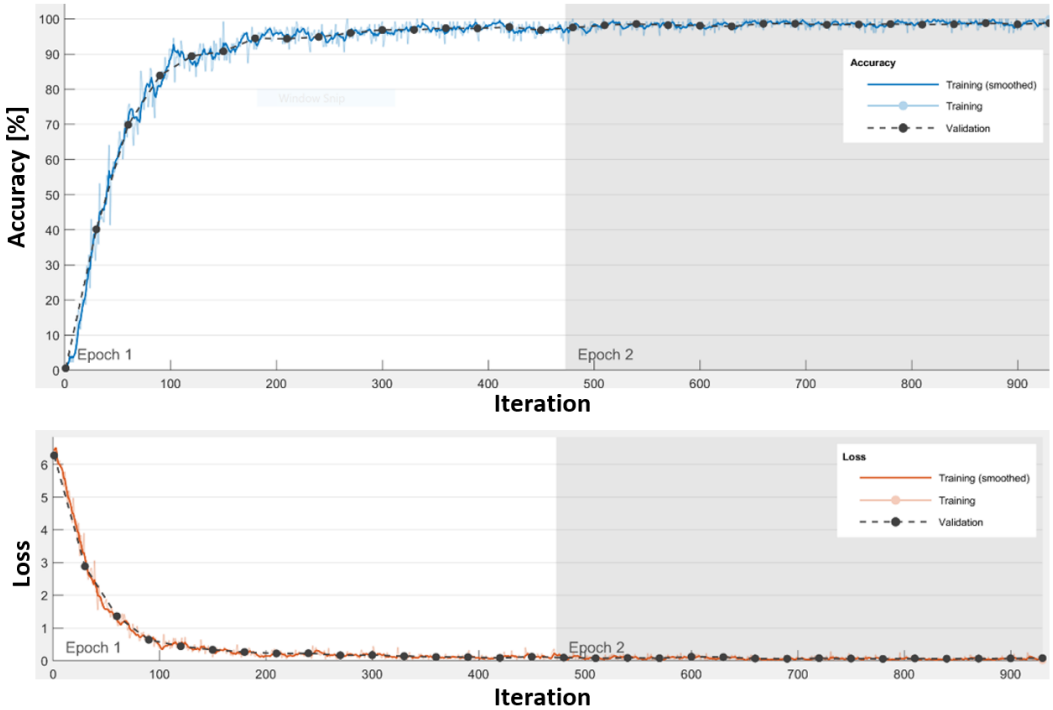

Fig. 6. Validation accuracy achieved using augmentation with only 1 degree rotation.

Although, the variations between validation and training sets are small when augmenting with 1 degree of rotation steps between 0 and 359 degrees. But the network performance against new flag images from the Internet, i.e., with different patterns, is astonishing. Which means that the network could somehow generalizes well to the problem.

Testing the trained network against images from the Internet, i.e., with different patterns, evaluates the proposed framework better than validation augmented images. Hence, 20 flag images from the Internet have been gathered and small modification to the original dataset images is applied, like cropping, addition of extra information, and adding random noise. The top-5 accuracy was used to evaluate these 20 images as another performance metric in addition to the validation accuracy. Training the shallow CNN architecture longer with 20-degree rotation augmentation images improves the validation accuracy as shown in Fig. 7.

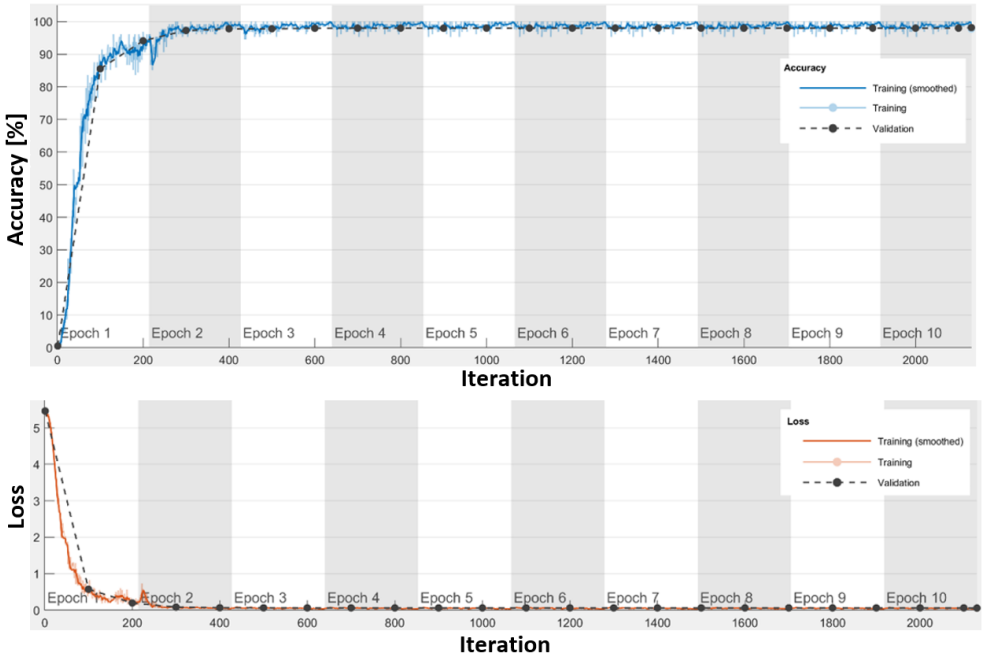

Fig. 7. Ten epoch validation accuracy for only 20-degree rotation.

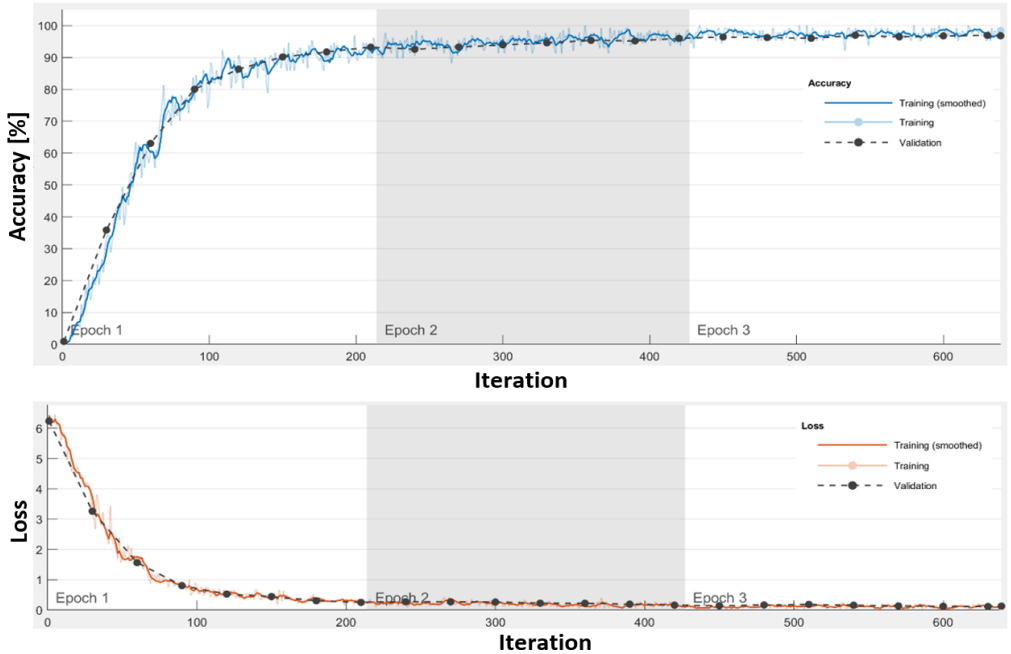

Fig. 8. Validation accuracy for 20-degree rotation with translation and shearing.

Nevertheless, longer training creates over-fitting in the network. This problem is clearly noticed when testing the network against the 20 extra images produced as another performance metric. To enhance the validation accuracy while avoiding over-fitting, other augmentation techniques are introduced to the 20-degree rotation, as in Fig. 8.

The validation accuracy of the shallow CNN architecture reached 99.06% with 5-degree rotation step between -45 and 45 degrees, 2 translations, 2 shearing transformations, and 2 scaling (stretching) transformations as shown in Fig. 9. Training and validation accuracies are enhanced in the first few epochs. Table 4 shows that scenario 9 achieves the best validation accuracy for the three network architectures. It provides good generalization with the 20 Internet flag images in its top1 and top-5 accuracy. So, from these results one can infer that augmentation helps to a given limit, and that augmentation techniques that are not suitable for the given application should not be used, e.g., background colour and noise for flag images, to avoid inappropriate results.

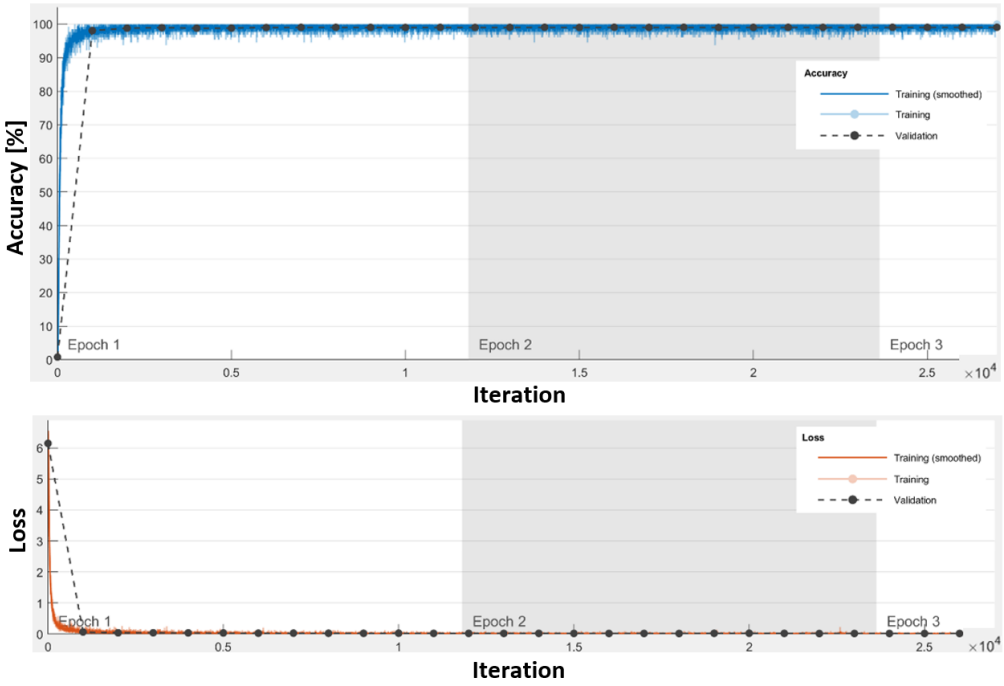

Fig. 9. Validation accuracy for 5-degree rotation with translation, shearing and scaling.

Table 4. Validation Accuracy for every network scenario combination for 7 epochs.

Scenario	Network validation accuracy [%]		
	Shallow CNN	Medium depth CNN	Deep CNN
1	98.70	98.34	98.11
2	98.01	98.0	98.0
3	98.83	98.25	98.56
4	98.50	98.29	98.55
5	97.48	97.39	97.44
6	98.13	97.93	97.79
7	99.01	98.85	98.92
8	99.05	99.04	99.05
9	99.60	99.58	99.59

It can be seen from Table 4 that all the results range between 97.39% and 99.60%. The shallow network architecture slightly outperformed deeper architectures using the validation augmented images in almost all the scenarios in Table 4. The best results are for scenario 9,

where all transformation methods are used with 1-degree rotation steps. Scenarios 8 and 7 produce the second and third best results as they use all transformation methods with 3 and 5-degree rotation steps, respectively. The worst results are with scenario 5, where white background after performing transformation on the images.

When testing the trained networks with the extra 20 Internet flag images, shallow CNN with augmentation scenario 9 achieved the best performance, i.e., accurately classifying 19/20 Internet flag images. The worst results are achieved using medium depth CNN with augmentation scenario 5, i.e., accurately classifying 5/20 Internet flag images. This shows that shallow networks are better than deeper networks with more parameters. The superiority of a shallow CNN is mainly due to having fewer parameters than deeper architectures, which can lead to reaching the global optima faster than deeper networks with problems with scarce training data [25]. Moreover, a shallow CNN generalizes better on simple datasets by capturing essential features compared to the unnecessary complexity that causes overfitting in deeper networks. The small number of details in flag images compared to natural images, e.g., landscapes, animals, or humans, simplifies the feature extraction process for shallow networks and reduces the complexity of the classification task.

4. CONCLUSIONS

In this work, a single sample per class classification using convolutional neural networks was carried out. To facilitate that, a combination of data augmentation techniques and proper network architectures was utilized, and a validation accuracy of 99% was achieved. The testing was performed using a subset of the augmented dataset used for the training process. Promising results were achieved without the need for a very deep convolutional neural network. Also, the type of data augmentation technique used must be carefully selected for each application to avoid over-fitting while obtaining the best validation accuracy. Selecting the wrong augmentation technique type can significantly drop the network ability to generalize well to new testing samples. As a future work, the same techniques used to augment the countries flags will be introduced but with natural images; such us, the human face recognition task. This task is a more difficult one because the number of details and features in the image will increase, and the importance of every pixel in the image will increase as well, unlike flags, where the redundancy is much higher. Another issue to consider is to input the images into the system using HSI color model instead of RGB. That should give more color information for the network to deal with. Also, studying the effect of augmenting the data with contrast stretching and histogram equalization will be considered to let the network tolerate small color variations in light illumination during image acquisition.

REFERENCES

[1] D. Hubel, T. Wiesel, "Receptive fields of single neurons in the cat's striate cortex," *The Journal of physiology*, vol. 148, no. 3, pp. 574–591, 1959, doi: 10.1113/jphysiol.1959.sp006308.

[2] A. Krizhevsky, I. Sutskever, G. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 60, pp. 84–90, 2012, doi:10.1145/3065386.

[3] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, L. Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015, doi: 10.48550/arXiv.1409.0575.

[4] 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, doi: 10.48550/arXiv.1502.03167.

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, " Going deeper with convolutions," IEEE Conference on Computer Vision and Pattern Recognition, 2015, doi: 10.48550/arXiv.1409.4842.

[6] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," Conference on Artificial Intelligence, 2017, doi: 10.48550/arXiv.1602.07261.

[7] K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition," IEEE Conference on Computer Vision and Pattern Recognition, 2016, doi: 10.48550/arXiv.1512.03385.

[8] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[9] M. Zeiler, R. Fergus, D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars, "Visualizing and understanding convolutional networks," European Conference on Computer Vision, 2014, doi: 10.48550/arXiv.1311.2901.

[10] K. Simonyan, A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015, doi: 10.48550/arXiv.1409.1556.

[11] [11] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, J. Wojna, "Rethinking the inception architecture for computer vision," IEEE Conference on Computer Vision and Pattern Recognition, 2016, doi: 10.48550/arXiv.1512.00567.

[12] D. Lowe, "Object recognition from local scale-invariant features," IEEE International Conference on Computer Vision, 1999, doi:10.1109/ICCV.1999.790410.

[13] M. Ebrahim, M. Alsmirat, M. Al-Ayyoub, "Advanced disk herniation computer aided diagnosis system," *Scientific Reports*, vol. 14, p. 8071, 2024, doi: 10.1038/s41598-024-58283-5.

[14] R. Deshpande, H. Patidar, "Detection of plant leaf disease using a lightweight parallel deep convolutional neural network," *Jordan Journal of Electrical engineering*, vol. 9, no.4, pp. 537-551, 2023, doi: 10.5455/jjee.204-1672998033.

[15] A. Alqudah, S. Qazan, H. Alquran, I. Abu Qasmieh, A. Alqudah, "COVID-19 Detection from X-ray Images Using Different Artificial Intelligence Hybrid Models", *Jordan Journal of Electrical Engineering*, 2020, vol. 6, no. 2, pp. 168-178, doi: 10.5455/jjee.204-1585312246.

[16] M. Al-Ayyoub, D. Alawad, K. Al-Darabsah, I. Aljarrah. "Automatic detection and classification of brain hemorrhages," *WSEAS Transactions on Computers,* vol. 12, no. 10, pp. 395–405, 2013, doi: 10.3390/diagnostics13182987.

[17] T. Ojala, M. Pietikäinen, T. Mäenpää, "Gray scale and rotation invariant texture classification with local binary patterns," European Conference on Computer Vision, 2000, doi: 10.1007/3-540-45054-8_27.

[18] N. Bayramoglu, J. Kannala, J. Heikkila, "Human epithelial type 2 cell ¨ classification with convolutional neural networksm," IEEE 15th International Conference on Bioinformatics and Bioengineering, 2015, doi: 10.1109/BIBE.2015.7367705.

[19] Z. Gao, J. Zhang, L. Zhou, L. Wang, "Hep-2 cell image classification with convolutional neural networks," Pattern Recognition Techniques for Indirect Immunofluorescence Images, 2014, doi: 10.1109/I3A.2014.15.

[20] X. Jia, L. Shen, X. Zhou,and S. Yu, "Deep convolutional neural network based HEp-2 cell classification,'' 23rd International Conference on Pattern Recognition, 2016, doi:10.1109/ICPR.2016.7899611.

[21] S. Jetley, A. Vaze, S. Belhe, "Automatic flag recognition using texture based color analysis and gradient features," IEEE Second International Conference on Image Information Processing, 2013, doi:10.1109/ICIIP.2013.6707635.

[22] C. Cortes, V. Vapnik, "Support vector machine," *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995, doi: 10.1007/BF00994018.

[23] M. Gu, K. Hao, Z. Qu, "Flag detection with convolutional network," International Conference on Computer Science and Artificial Intelligence, 2018, doi:10.1145/3297156.3297159.

[24] M. Ebrahim, M. Alsmirat, M. Al-Ayyoub, "Performance study of augmentation techniques for HEp2 CNN classification," International Conference on Information and Communication Systems, 2018, doi: 10.1109/IACS.2018.8355460.

[25] L. Brigato, L. Iocchi, "A close look at deep learning with small data," International Conference on Pattern Recognition, 2021, doi: 10.1109/ICPR48806.2021.9412492.