DEEP LEARNING STRATEGIES BASED ON FINE-TUNING. APPLICATION TO MEDICAL IMAGE CLASSIFICATION

Stefania BARBURICEANU, Andreia MICLEA, Romulus TEREBES
Technical University of Cluj-Napoca, Communications Department, Cluj-Napoca, Romania
Stefania.Barburiceanu@com.utcluj.ro, Andreia.Miclea@com.utcluj.ro, Romulus.Terebes@com.utcluj.ro

Abstract: This research focuses on the use of fine-tuning techniques to analyze various scenarios where some of the learnable layers in a CNN model are frozen and some are fine-tuned. The experimental section involves popular CNN architectures such as AlexNet and Vgg, and analyzes the classification performance of such models by freezing successively the convolutional and fully connected layers. The classification layers in the proposed technique are substituted with new ones that are trained starting with random values, while the other layers are either frozen or fine-tuned. For the frozen layers, the knowledge obtained from training on a vast dataset is applied to the new classification task, whereas for fine-tuned layers, the network parameters are changed to suit the new problem. The goal of this research is to identify the network location that enables the ideal balance between the generality and specificity of extracted features. The experimental study was undertaken using a small public dataset of medical images. The results are also compared against those obtained when fine-tuning is performed for all layers, as well as when all network parameters are directly transferred from pre-trained models for various learning rate values. For the considered models, the point that offered the optimum compromise between specificity and generality was found. The worst results are obtained when no layers are frozen. When utilizing the same weights from the pre-trained model, the results demonstrate that the classification scores may be improved by appropriately modifying the learning rate of the new classification layers.

Keywords: *deep learning, texture classification, image classification, Convolutional Neural Networks*.

I. INTRODUCTION

In this day and age, massive amounts of different types of information are created every day and images account for a considerable proportion of the data that is generated. As a consequence, one can rely on recent technology such as deep learning methods to rapidly interpret visual scenes and conduct image classification tasks. Deep learning approaches can solve this task by learning from labelled samples using supervized learning techniques. The task of providing a label to a given image is known as image classification. Deep learning approaches can estimate the class of membership for new images based on a training procedure in which a learning process takes place.

Such techniques have lately proved their capabilities in classifying efficiently images for a range of applications including but not limited to the medical and agricultural domains, home security, self-driving vehicles, biometric data-based applications, industry, and many more. The most extensively used deep-learning method for image processing tasks is the Convolutional Neural Network (CNN). CNNs include four main layer types:

- **Convolutional**

These layers are responsible for extracting different characteristics from the input data. The mathematical convolution is done between the input and a filter by swiping it over the input image. The result is referred to as the feature map which is later given to further layers in order for them to learn numerous other attributes about the original image.

- **Pooling**

A convolutional layer is typically succeeded by a pooling layer type. This layer's major objective is to reduce the dimension of the feature maps. In this way, the computational overhead of the network is lowered. This process is conducted independently for each obtained feature map. There are diverse types of pooling procedures depending on the employed approach such as max pooling (where the highest value from a considered region is selected) and average pooling (where the average value is kept) [1].

- **Non-linear**

The activation function, which imparts non-linearity to the network, is one of the most significant components of a CNN. The ReLU, tanh, and sigmoid functions are some of the most often utilized activation functions [1].

- **Fully connected**

These layers are often the final layers of a CNN model that based on the attributes acquired by the convolutional and pooling layers, perform the actual classification. A flattening process is required before the generated feature maps can be fed into a fully connected structure. The process entails converting the feature map into a 1D vector.

There are also other types of layers such as the normalization and dropout layers that can be used to improve the network performance in certain situations.

The appropriate weights (values of the filters) and biases of the network are learned by the CNN during training which is performed with the help of the backpropagation technique. It operates under the theory that by altering the weights of the inputs, the expected outcome can be obtained. The network learns based on knowing the actual correct response of an input, then using

Manuscript received: October 16, 2022; revised November 29, 2022

that information to modify the weights of its filters.

CNNs can serve as an entire classification chain by combining the convolutional layers that identify the most important attributes of the image under consideration with the fully-connected ones that are capable of conducting the classification process. This strategy's main drawback is that the classification performance is strongly influenced by the number of training samples employed. It is possible to not get adequately good results if a high amount of data is not accessible and as a result, the transfer learning strategy gains considerable value. According to this method, the data gathered for an initial assignment might serve as the foundation for a subsequent classification problem [2]. The benefit is the ability to handle data quickly and efficiently without manually creating complex CNN networks or using highly specialized hardware. The most popular strategy is to employ well-known CNN architectures that yielded significant results and that have already been pre-trained on massive amounts of data that come from a wide variety of categories.

The two primary methodologies in transfer learning involve either directly employing pre-trained CNN models or applying a fine-tuning procedure. We already explored methods that incorporate pre-trained models and used the same learnt weights in new classification tasks in [3]–[5].

This study focuses on the application of fine-tuning methods to examine multiple situations where a variable number of layers are frozen in a CNN. This is crucial for improving the model accuracy when datasets with a limited number of samples are employed. This is particularly helpful for such datasets containing images that deviate significantly from the original dataset that was used for training. This research aims to find the network node that achieves the best possible balance between generality and specificity. In this way, the model can then learn additional particular characteristics linked to the new classification topic once all deeper layers have been adjusted. Secondly, since there are weights already learnt that are linked to more general properties not necessarily particular to the primary classification scenario, they are maintained the same by freezing all layers before that ideal point.

Transfer learning has been effectively applied to a number of CNN models, including GoogleNet [6], VggNet [7], AlexNet [8], ResNet [9], Inception-v3 [10], and InceptionResnet-v2 [11]. In [12], the authors employed fine-tuning strategies for the classification of images belonging to a small medical image dataset for the detection of oral cancer. The classification of liver-related abnormalities from computerized tomography (CT) images is also addressed in [13] by considering fine-tuning approaches based on the Resnet CNN model. Fine-tuning was also part of the methodology considered in [14] for X-ray images involving data acquired for the detection of Covid-19 disease. Besides the medical domain, there are also other applications in which fine-tuning proved to be successful such as the identification of emotions from facial image data [15] or the classification of plant diseases [16].

II. PROPOSED METHOD

The proposed method makes use of a ConvNet that has previously been trained by considering ImageNet [17], a very large dataset. In order to tackle a new classification task, one of the models under consideration is used as a starting place. Transfer learning speeds up the fine-tuning process compared to starting from a random initialization of the network weights. The output layers of the considered pre-trained model, which are the fully connected, softmax, and classification layers, are configured for 1000 labels, which match the number of classes of ImageNet. As a result, new layers (termed in the following as new classification layers) that are suited for training data related to the current task are used to replace the output layers of the CNN architecture. The number of outputs corresponding to the newly introduced layers is determined by the number of categories associated with the new dataset.

The learning rate is a crucial training process variable. Although a high learning rate makes the model learn more quickly, it is more probable in this case to obtain suboptimal weight values. While training takes longer if the learning rate has a smaller value, it is more likely to reach better-suited values for the network parameters.

Since the output layers in the proposed technique are replaced with new ones, they must be trained from scratch. In this case, the learning rate is changed such as the learning is faster in these layers with respect to the other layers in the ConvNet model. This adjustment is done by multiplying the global learning rate by 10 for the weights and biases associated with these new layers.

The two models investigated in this study are shown in Figure 1. Their configuration can be found in [18].

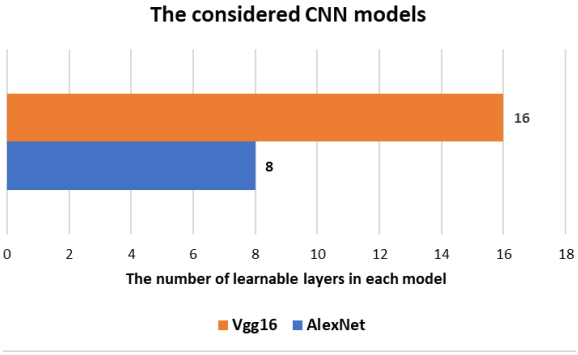

Figure 1. The CNN models employed in the study

The initial stage in the fine-tuning procedure is changing the size of the input images to suit the size criteria of the CNN model being taken into account and both training and testing images go through this process. The training images are also subjected to an augmentation process that takes into account random translation, flipping, rotation, and scaling. This method helps lower the possibility of overfitting which means that a model becomes so skilled at retaining the details of the training examples that it is unable to generalize to previously unseen images. Figure 2 presents the processing steps that are applied to the images from the considered dataset.

Figure 2. Processing of images from the dataset

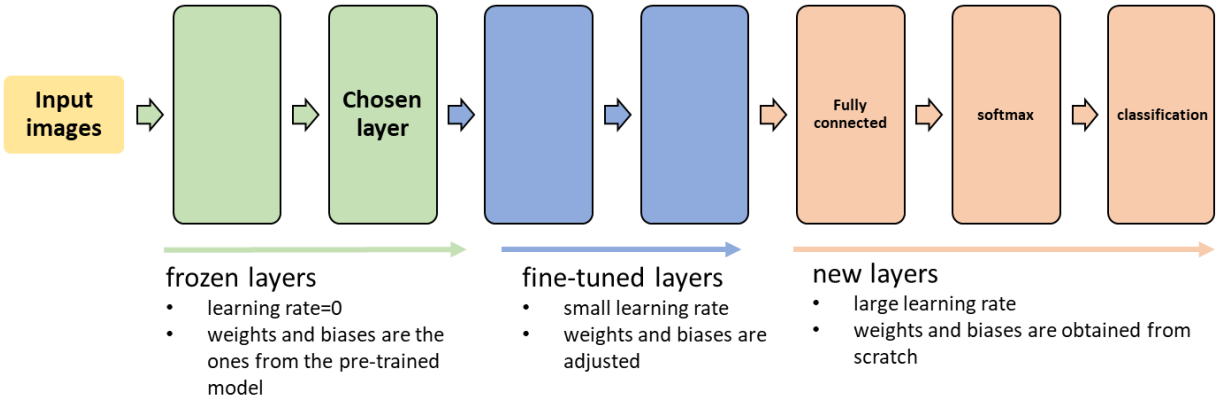

Figure 3. The considered fine-tuning strategy

The next step consists in considering all fully connected and convolutional layers (except the new classification layers at the end of the network) in each model to analyze the influence of freezing the weights and biases of the network. For each such learnable layer for the considered CNN architectures, there is analysed the scenario of freezing that specific layer and all layers before it. This means that for these layers (termed in the following as frozen layers), the weights and biases already acquired from the pre-training procedure (on ImageNet) are retained without any modification. This is done by considering a zero learning rate for these specific layers. In this case, no learning takes place and the parameters are kept exactly the same.

For the other layers in the network (termed in the following as fine-tuned layers) that are not frozen, fine-tuning is applied. This signifies that training is carried out for the fine-tuned layers in order to modify the weights and biases to suit the new classification task. Fine-tuning requires in this case a lower value for the learning rate relative to the new classification layers since the fine-tuned layers have previously been trained on ImageNet. The reason behind this is the fact that in these situations since training was done on a large dataset, the weights and biases are already adapted to detect different attributes in the input image. This means that the convolutional filters are already tuned to detect different types of features, so they do not need a large learning rate.

We also considered the situation in which the freezing is not performed at all and fine-tuning takes place for all layers.

The next step is the training process that is performed for each identified learnable layer for the considered CNN models. In this case, the optimizer used is the Stochastic Gradient Descent (SGD) with momentum [19] (with a contribution of 0.99 from the preceding step) and the global learning rate is 10^{-4}. The size of the mini-batch used is 10 and training is performed on 6 epochs, which is enough for fine-tuning operations.

For the validation step, several performance metrics are computed: accuracy, macro-averaging precision, macro-averaging recall, and F1 score. Apart from the F1 score, the other parameters are the same as in [5]. As the harmonic mean of precision and recall, the F1 score is a useful statistic that takes both values into account as expressed in Eq. (1):

$$F_1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (1).$$

Figure 3 presents the workflow involved in the proposed method.

III. EXPERIMENTAL RESULTS

1. Dataset

The experimental evaluation makes use of dermoscopic images of melanocytic lesions obtained with a mole analyzer. The dataset was obtained by [20] and is known as PH2. Figure 4 shows the setup used.

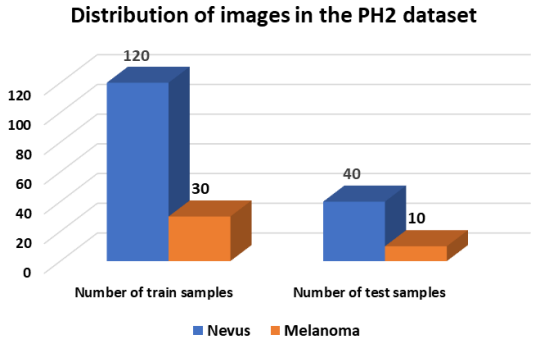

Figure 4. The considered setup

Some example images from this dataset are presented in Figure 5.

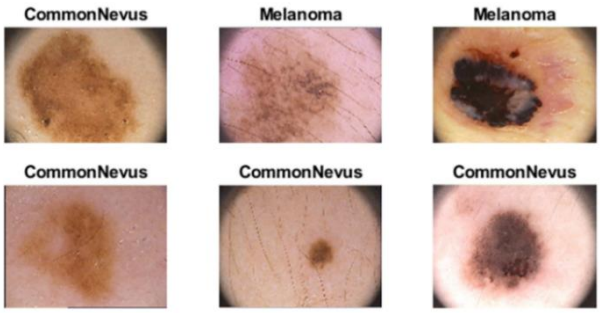

Figure 5. Image samples

2. Results and discussion

The experimental assessment entails assessing the classification ability of two CNN models in situations with a varying number of frozen layers.

The results are also compared to the ones achieved by

considering fine-tuning for all layers and by transferring all network parameters directly from pre-trained models. For all obtained results the same random partition of training and test images is considered. The layers in the CNN model are numbered as in [18] when all layers including learnable layers (convolutional and fully connected) but also ReLu, normalization, pooling and dropout layers are included.

Figure 6 presents the F1 score obtained using the AlexNet CNN architecture for three different values of the learning rate by freezing consecutively every learnable layer in the network. Table I presents the best results obtained for each learning rate value (lr) along with scores achieved when no layer is frozen and all layers are frozen.

Table I. Classification scores [%] obtained for AlexNet

Conditions	Accuracy	Precision	Recall	F1 score
Global learning rate: 0.000025				
Layer 6 and all layers before are frozen	90.00	85.23	82.50	83.84
No layer is frozen	84.00	91.67	60.00	72.53
All layers are frozen	78.00	70.50	78.75	74.40
Global learning rate: 0.0001				
Layer 10 and all layers before are frozen	92.00	87.50	87.50	**87.50**
No layer is frozen	82.00	90.82	55.00	68.51
All layers are frozen	86.00	78.57	87.50	82.80
Global learning rate: 0.0002				
Layer 12 and all layers before are frozen	80.00	75.00	87.50	80.77
No layer is frozen	20.00	60.00	50.00	54.55
All layers are frozen	92.00	90.18	83.75	86.85

The best result is obtained when a global learning rate of 10^{-4} is considered and all layers up until and including layer 10 are frozen. So, the optimal point that generated the best balance between specificity and generality is in this case **layer 10** (see [18]).

If no layer is frozen, it means that fine-tuning is performed for all layers and in this case, all network parameters are adjusted. For all learning rates, we can see that this situation is the one that obtained the worst classification scores. We can also observe that the F1 score decreases as the learning rate increases in this case. Since the new dataset is so small, overfitting occurs when all layers are trained and for higher learning rates, the results are even worse because there are taken bigger steps when adjusting the network parameters.

Table II. Classification scores [%] obtained by freezing all layers in AlexNet for different learning rate values

Metric	lr* = 0.002	lr* = 0.01	lr* = 0.015	lr* = 0.02
Accuracy	92	92	88	86
Precision	90.18	95.45	80.70	78.07
Recall	83.75	80	85	83.75
F1 score	86.85	**87.05**	82.8	80.81

*the learning rate of the new classification layers = 10×global learning rate

Table III. Classification scores [%] obtained for Vgg16

Conditions	Accuracy	Precision	Recall	F1 score
Global learning rate: 0.000025				
Layer 4 and all layers before are frozen	94.00	96.51	85.00	90.39
No layer is frozen	88.00	81.25	81.25	81.25
All layers are frozen	90.00	88.21	78.75	83.21
Global learning rate: 0.0001				
Layer 26 and all layers before are frozen	98.00	98.78	95.00	**96.85**
No layer is frozen	86.00	83.33	68.75	75.34
All layers are frozen	92.00	95.45	80.00	87.05
Global learning rate: 0.00015				
Layer 26 and all layers before are frozen	92.00	87.50	87.50	87.50
No layer is frozen	80.00	90.00	50.00	64.29
All layers are frozen	92.00	95.45	80.00	87.05

The situation when all layers are frozen corresponds to using in fact the same weights as for the pre-trained AlexNet model by transferring them directly to the new classification task. The obtained F1 score in this case increases as the learning rate increases for the considered learning rate values. This happens because a higher global learning rate involves a higher learning rate for the new classification layers that are trained from scratch. In this case, bigger steps are considered for adjusting the parameters in these layers, and thus, these layers achieve better capabilities in performing the classification of features derived by the previous convolutional layers. To better investigate this scenario, several experiments were performed to observe if by increasing the learning rate even further the results can be improved.

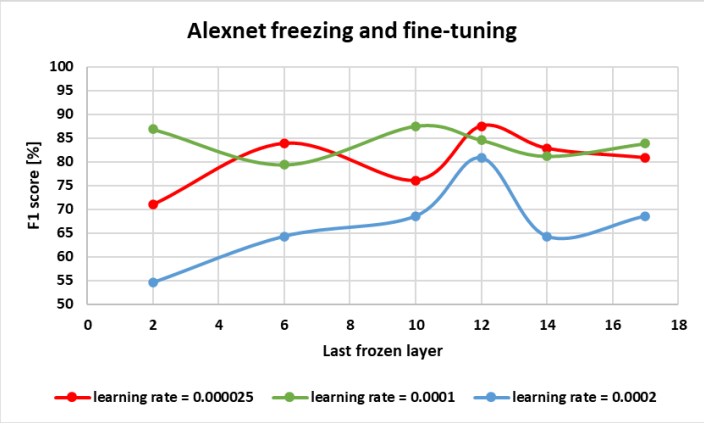

Figure 6. Obtained classification results using AlexNet

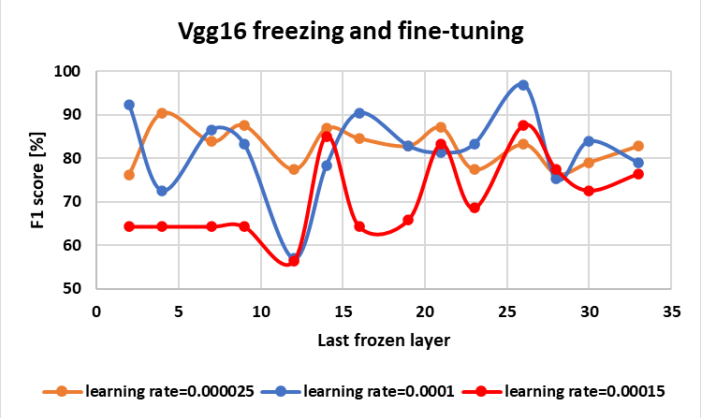

Figure 7. Obtained classification results using Vgg16

The obtained scores are shown in Table II. A small increase in performance can be observed when considering 0.01 as the learning rate for the new classification layers. However, the further increase seems detrimental to the classification performance.

Figure 7 shows the F1 score achieved by employing the Vgg16 model for different learning rate values by freezing successively every learnable layer in the network. Table III provides the top outcomes for each learning rate value together with the results achieved when no layer is frozen and all layers are frozen. In this case, the best score is achieved when considering the same global learning rate (10^{-4}) as for AlexNet and by freezing all layers up until and including **layer 26** (see [18]). The same remarks discussed for AlexNet regarding the scenarios when all layers are frozen and no layer is frozen remain valid for Vgg16.

In [21], [22], the authors reported a 97.5% accuracy for the same dataset using classical descriptors. Combining classical descriptors and features derived from deep learning methods produced an accuracy of 96.5% in [23], while the extraction of attributes from pre-trained CNNs generated a 93% accuracy in [24]. In [25], 88% accuracy was obtained by pooling features from the final layers of several pre-trained models. However, the authors enhance their results by using a feature selection technique and integrating CNN features with other classical descriptors, yielding a value of 98% accuracy.

IV. CONCLUSIONS

This study focuses on the use of fine-tuning deep learning methods to examine various situations in which certain learnable layers in a CNN model are frozen while for others the network parameters are adjusted. By freezing sequentially the convolutional and fully connected layers, the experimental section examines the classification performance of two well-known CNN designs, AlexNet and Vgg16. In the proposed technique, the classification layers are replaced with new ones for which training is performed starting with random values, while for the remaining layers, the parameters are either kept the same or adjusted. An exploratory investigation that used a limited public database of medical images was conducted by also varying the value of the learning rate. A comparison was made with two distinct scenarios: fine-tuning all layers and directly importing all network parameters from previously trained models. In the experimental section, the ideal location that produced the best balance between specificity and generality was identified for the two models that were taken into consideration. The least accurate classification results are obtained when no layer is frozen. Due to the small size of the new dataset, overfitting appears when all layers are trained, and the results are considerably worse at higher learning rates. When all layers are frozen, it is equivalent to using the identical weights from the pre-trained model and in this case, the obtained classification scores can be

improved by properly adjusting the learning rate of the new classification layers.

As future work, we intend to perform the presented evaluation on other small datasets from the medical or precision agriculture domains and to carry out a clinical validation.

REFERENCES

[1] L. Alzubaidi *et al.*, "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, no. 1, p. 53, Mar. 2021, doi: 10.1186/s40537-021-00444-8.

[2] M. Shaha and M. Pawar, "Transfer Learning for Image Classification," in *Proceedings of the Second International Conference on Electronics, Communication and Aerospace Technology (ICECA 2018)*, Piscataway, New Jersey, Mar. 2018, pp. 656–660. doi: 10.1109/ICECA.2018.8474802.

[3] S. Barburiceanu, S. Meza, B. Orza, R. Malutan, and R. Terebes, "Convolutional Neural Networks for Texture Feature Extraction. Applications to Leaf Disease Classification in Precision Agriculture," *IEEE Access*, vol. 9, pp. 160085–160103, 2021, doi: 10.1109/ACCESS.2021.3131002.

[4] S. Barburiceanu and R. Terebes, "FEATURE EXTRACTION METHODS BASED ON DEEP-LEARNING APPROACHES. APPLICATION TO AUTOMATIC DIAGNOSIS OF BREAST CANCER," *Acta Technica Napocensis Electronics and Telecommunications*, vol. 62, no. 1, 2022.

[5] S. Barburiceanu and R. Terebeș, "Automatic detection of melanoma by deep learning models-based feature extraction and fine-tuning strategy," *IOP Conf. Ser.: Mater. Sci. Eng.*, vol. 1254, no. 1, p. 012035, Sep. 2022, doi: 10.1088/1757-899X/1254/1/012035.

[6] C. Szegedy *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1–9. doi: 10.1109/CVPR.2015.7298594.

[7] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556 [cs]*, Apr. 2015, Accessed: Mar. 01, 2022. [Online]. Available: http://arxiv.org/abs/1409.1556

[8] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Neural Information Processing Systems*, vol. 25, pp. 1097–1105, Jan. 2012, doi: 10.1145/3065386.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 2818–2826. doi: 10.1109/CVPR.2016.308.

[11] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," *arXiv:1602.07261 [cs]*, Aug. 2016, Accessed: Apr. 02, 2022. [Online]. Available: http://arxiv.org/abs/1602.07261

[12] R. A. Welikala *et al.*, "Fine-Tuning Deep Learning Architectures for Early Detection of Oral Cancer," in *Mathematical and Computational Oncology*, Cham, 2020, pp. 25–31. doi: 10.1007/978-3-030-64511-3_3.

[13] W. Wang *et al.*, "Classification of Focal Liver Lesions Using Deep Learning with Fine-Tuning," in *Proceedings of the 2018 International Conference on Digital Medicine and Image Processing*, New York, NY, USA, Nov. 2018, pp. 56–60. doi: 10.1145/3299852.3299860.

[14] T. D. Pham, "Classification of COVID-19 chest X-rays with deep learning: new models or fine tuning?," *Health Inf Sci Syst*, vol. 9, no. 1, p. 2, Nov. 2020, doi: 10.1007/s13755-020-00135-3.

[15] H.-W. Ng, V. D. Nguyen, V. Vonikakis, and S. Winkler, "Deep Learning for Emotion Recognition on Small Datasets using Transfer Learning," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, New York, NY, USA, Nov. 2015, pp. 443–449. doi: 10.1145/2818346.2830593.

[16] Z. Qiang, L. He, and F. Dai, "Identification of Plant Leaf Diseases Based on Inception V3 Transfer Learning and Fine-Tuning," in *Smart City and Informatization*, Singapore, 2019, pp. 118–127. doi: 10.1007/978-981-15-1301-5_10.

[17] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *Int J Comput Vis*, vol. 115, no. 3, pp. 211–252, Dec. 2015, doi: 10.1007/s11263-015-0816-y.

[18] "Pretrained Deep Neural Networks - MATLAB & Simulink." https://www.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html (accessed Oct. 15, 2022).

[19] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Networks*, vol. 12, no. 1, pp. 145–151, Jan. 1999, doi: 10.1016/S0893-6080(98)00116-6.

[20] T. Mendonça, P. M. Ferreira, J. S. Marques, A. R. S. Marcal, and J. Rozeira, "PH2 - A dermoscopic image database for research and benchmarking," in *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, Osaka, Japan, Jul. 2013, pp. 5437–5440. doi: 10.1109/EMBC.2013.6610779.

[21] T. Akram, M. A. Khan, M. Sharif, and M. Yasmin, "Skin lesion segmentation and recognition using multichannel saliency estimation and M-SVM on selected serially fused features," *J Ambient Intell Human Comput*, Sep. 2018, doi: 10.1007/s12652-018-1051-5.

[22] M. Nasir, M. Attique Khan, M. Sharif, I. U. Lali, T. Saba, and T. Iqbal, "An improved strategy for skin lesion detection and classification using uniform segmentation and feature selection based approach," *Microscopy Research and Technique*, vol. 81, no. 6, pp. 528–543, 2018, doi: 10.1002/jemt.23009.

[23] N. Moura *et al.*, "Combining ABCD Rule, Texture Features and Transfer Learning in Automatic Diagnosis of Melanoma," in *2018 IEEE Symposium on Computers and Communications (ISCC)*, Natal, Brazil, Jun. 2018, pp. 00508–00513. doi: 10.1109/ISCC.2018.8538525.

[24] J. A. A. Salido and C. R. Jr., "Using Deep Learning for Melanoma Detection in Dermoscopy Images," *IJMLC*, vol. 8, no. 1, pp. 61–68, Feb. 2018, doi: 10.18178/ijmlc.2018.8.1.664.

[25] Y. Filali, H. EL Khoukhi, M. A. Sabri, and A. Aarab, "Efficient fusion of handcrafted and pre-trained CNNs features to classify melanoma skin cancer," *Multimed Tools Appl*, vol. 79, no. 41, pp. 31219–31238, Nov. 2020, doi: 10.1007/s11042-020-09637-4.