2022 19th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE). Mexico City, Mexico. November 9-11, 2022

ECG Arrhythmia Classification for Comparing Pre-Trained Deep Learning Models

Oliverio Arellano Cárdenas
Electrical Engineering Department, Solid State Section, CINVESTAV-IPN, Mexico City, Mexico
phone (52) 55 5747 3800 x 6261
email: arellano@cinvestav.mx

Luis Martín Flores Nava
Electrical Engineering Department, Solid State Section, CINVESTAV-IPN, Mexico City, Mexico
phone (52) 55 5747 3800 x 6263
email: lmflores@cinvestav.mx

Felipe Gómez Castañeda
Electrical Engineering Department, Solid State Section, CINVESTAV-IPN, Mexico City, Mexico
phone (52) 55 5747 3800 x 6262
email: fgomez@cinvestav.mx

José Antonio Moreno Cadenas
Electrical Engineering Department, Solid State Section, CINVESTAV-IPN, Mexico City, Mexico
phone (52) 55 5747 3785
email: jmoreno@cinvestav.mx

***Abstract*—In order to perform the diverse tasks of deep learning, different models of neural networks have been proposed, and for several years now many deep neural networks have been developed for a variety of uses. In general, the labor of training these networks is the most challenging part of this methodology, and is by far the most time consuming, both in terms of effort required to configure the process and computational complexity required to perform it. To relieve this process, pre-trained models have been made available that already learned to extract powerful and informative features from big sets of images, thus reducing training time for particular problems, but, given the large number of options, it would consume a lot of time testing each of them. In this study, we tested the performance of nineteen pre-trained networks that are available for MATLAB and compared their performance for classification of ECG signals from MIT-BIH arrhythmia dataset. We report the main characteristics of these networks, training times, and performance for this classification job to serve as reference for their use by other researchers.**

Keywords—Pre-trained deep models, MIT-BIH database, arrhythmia classification, deep learning.

I. INTRODUCTION

Deep learning is a branch of machine learning, which in turn is a subset of Artificial Intelligence (AI) whose objective is to enhance machines capacities for tasks such as classification, recognition, detection, and description, imitating human skills. The phrase deep learning refers to machine learning algorithms that use a series of steps, or layers, of computation [1][2]. these systems are called "deep" only because of their appearance when are drawn stacked up vertically.

If there is a state-of-the-art algorithm that resolves a problem with the utmost accuracy, it can be used indirectly to resolve other related problems. A network can be trained with such algorithm in a lab with powerful machines and plenty of resources, and take the necessary time to get an optimal performance, then if this trained network is available for external users, they won't need to train it so meticulously to obtain remarkable results. It will only be necessary replacing some layers of the network, and focus the training on this new layers instead of the whole network, saving computing resources, time, and money. This procedure is called 'Transfer Learning'. Transfer learning has been described for a long time [3], but its usage in deep learning has happened just recently. Decaf [4] and Overfeat [5] are the first feature extractor networks published off-the-shelf. The first one is based on AlexNet [6], the latter is a custom architecture. Both were pre-trained on ImageNet [7] and offer general features for computer vision jobs. Since then, many models have been developed with different capacities and configurations.

Deep Learning Toolbox of MATLAB handles several pre-trained deep networks with different characteristics such as accuracy, speed and size. Picking one of them normally is a tradeoff among these characteristics. MATLAB's website section for Pre-trained Deep Neural Networks [8] has nineteen pre-trained models available to be installed in MATLAB platform as well as their respective original references. These models are briefly described in Table 1, and they are those that we tested in our study, so other researchers have a preliminary reference for using them, and have a better idea about their accuracy, training times and general performance.

II. MIT ECG ARRHYTMIA DATABASE

We used MIT-BIH Arrhythmia database (CD edition) as a source for medical ECG signals [9]. This is a reference in many papers found from literature. The database contains 48 half-hour two-channel ECG recordings from 47 patients recorded between 1975 and 1979. Signals are digitized at 360 samples per second per channel (2 channels per record) and has 11-bit resolution over a 10 mV range. Two or more cardiologists, working independently, labeled MIT-BIH Arrhythmia database. Annotation for each heart beat has been placed at the R-wave peak. Therefore, the exact location of R-wave peak can be seen from the timestamp of the label and there is no need of QRS detection algorithm.

MIT-BIH Arrhythmia database signals are annotated using 23 labels as shown in Table 2. We focused our study in pathologies that had at least around 1000 beats. Then we chose six cardiopathies and normal beats for the classification problem, they were beats labeled A, F, f, L, N, R and V. In order to have a balanced amount of beats, we took the total F and f beats, and for the others we took similar quantities from each record to complete roughly 1000 beats for each class. Our dataset was defined as follows: A=1032 beats, F=803 beats,

978-1-6654-5508-4/22/$31.00 ©2022 IEEE

f=982 beats, L=1064 beats, N=1200 beats, R=1085 beats and V=1539 beats.

TABLE 1. Pre-trained MATLAB deep models.

Network	Main contribution	Depth	Parameters (Millions)	Image Input Size
alexnet	Deeper than initial CNN architecture LeNet and uses relu and overlapping pooling	8	61	227-by-227
darknet19	Faster and more accurate than Darknet Reference Model. Backbone of YOLOv2 approach.	19	20.8	256-by-256
darknet53	Residual blocks, skip connections, and upsampling layers. Backbone for YOLOv3.	53	41.6	256-by-256
densenet201	Cross-layered information flow	201	20	224-by-224
efficientnetb0	Scaling method to scale all dimensions of depth/width/resolution a compound coefficient.	82	5.3	224-by-224
googlenet	Introduced block concept, split transform, and merge idea	22	7	224-by-224
inceptionresnetv2	Processed at reasonable cost for scenarios where memory or computational capacity is inherently limited	164	55.9	299-by-299
inceptionv3	Bottleneck issue is sorted and small filter size is added	48	23.9	299-by-299
mobilenetv2	Based on an inverted residual structure.	53	3.5	224-by-224
nasnetlarge	Improves generalization in the NASNet models.	*	88.9	331-by-331
nasnetmobile	Design of a search space (the "NASNet search space") which enables transferability.	*	5.3	224-by-224
resnet101	Better accuracy than resnet50	101	44.6	224-by-224
resnet18	Modularized architectures using residual learning	18	11.7	224-by-224
resnet50	Better accuracy than resnet18	50	25.6	224-by-224
shufflenet	Utilizes two operations, pointwise group convolution and channel shuffle, to reduce computation cost.	50	1.4	224-by-224
squeezenet	Achieves AlexNet-level accuracy on ImageNet with 50x fewer parameters.	18	1.24	227-by-227
vgg16	Uses small kernel size	16	138	224-by-224
vgg19	19 layer version of Vgg16 for larger images	19	144	224-by-224
xception	Depth-wise convolution followed by a point-wise convolution	71	22.9	299-by-299
* The NASNet-Mobile and NASNet-Large networks do not consist of a linear sequence of modules.				

TABLE 2. Labels used in MIT-BIH Arrhythmia database.

Symbol	Description	Number of events
A	Atrial premature beat	2546
a	Aberrated atrial premature beat	150
E	Ventricular escape beat	106
e	Atrial escape beat	16
F	Fusion of ventricular and normal beat	803
f	Fusion of paced and normal beat	982
J	Nodal (junctional) premature beat	83
j	Nodal (junctional) escape beat	229
L	Left bundle branch block beat	8075
N	Normal beat	70399
Q	Unclassifiable beat	28
R	Right bundle branch block beat	7259
S	Supraventricular premature or ectopic beat (atrial or nodal)	2
V	Premature ventricular contraction	11783
x	Non-conducted P-wave (blocked APC)	193
+	Rhythm change	1290
~	Change in signal quality	616
|	Isolated QRS-like artifact	132
/	Paced beat	7033
[	Start of ventricular flutter/fibrillation	6
]	End of ventricular flutter/fibrillation	6
!	Ventricular flutter wave	472
"	Comment annotation	437

III. PROPOSED METHOD

There have been several comparative reports for pre-trained neural networks e.g. [10][11], but they focus their analysis on chronological or relevant networks and for contribution in some relevant field, so excluding some models. Here we realized a study bearing in mind all the nineteen pre-trained networks available as part of MATLAB, to give to researchers interested in using these models, an idea of what can they obtain by using one model or other.

In order to realize our study, we selected MIT-BIH database because it is a well-known reference in many works and is available for anyone. Many researchers using MIT-BIH database have developed their classification procedures by considering the records as digitized signals. The pre-trained networks we worked with, are developed to use images as input, so we settled our tests based on the procedure described in [12]: As pre-processing stage (Fig. 1), we converted ECG signals from one channel into ECG images by plotting each ECG beat in an individual 664 x 520 image, centering the Q-wave and taking 150 samples before and after it to include P and T waves in the image. Images were separated according to their labels. We obtained 7705 images classified as shown in Fig. 2.

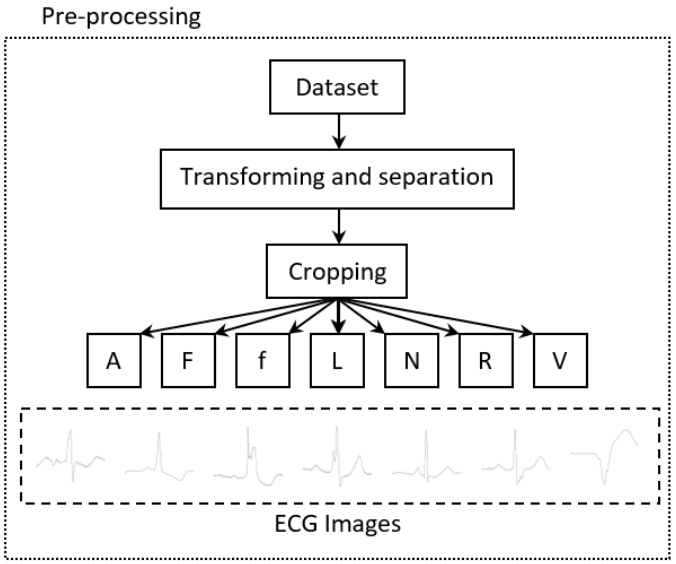


Fig. 1. Pre-processing stage to obtain ECG images from datataset.

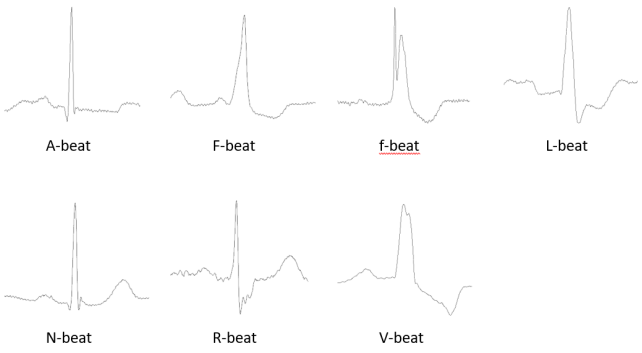


Fig. 2. Example of beats for the seven classes A, F, f, L, N, R and V.

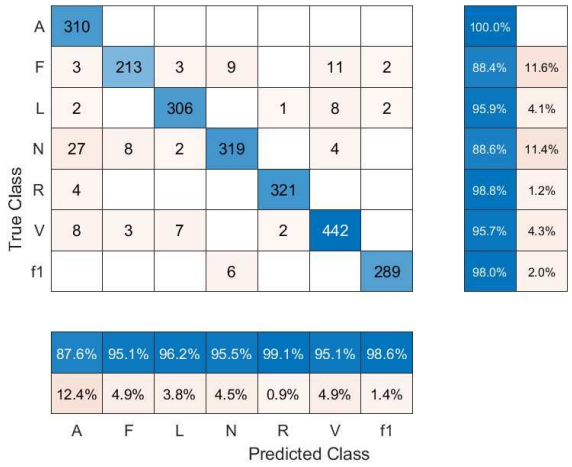
With exception of nasnetlarge (that ran using CPU rather than GPUs), and vgg16 and vgg19 (that ran with batch size = 100), we trained all networks with the following configuration: randomly took 70% images for training and 30% for validation, optimization method: Adam, 6 training epochs, batch size = 10, initial learning rate = 3×10^{-4}, no freezing layers, use of an

augmented image datastore to automatically resize the training images, additional augmentation operations to perform on the training images: randomly translate them up to 30 pixels and scale them up to 10% horizontally and vertically in order to prevent overfitting. The last three layers of all the pre-trained networks are configured for 1000 classes. These three layers must be replaced to fit the new classification problem.

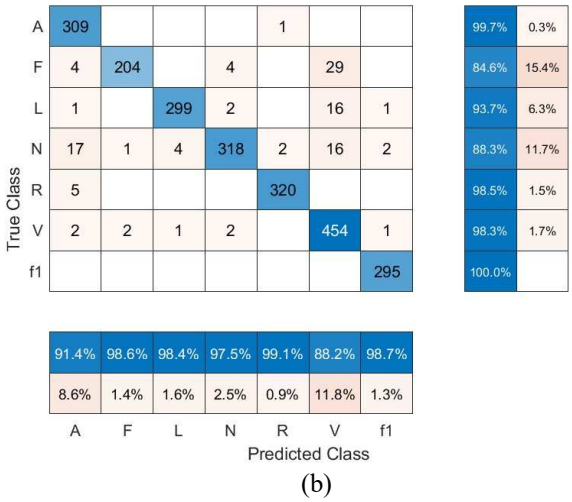
All the training processes were realized in a computer with the following features: 64-bit Microsoft Windows 10 Pro, MATLAB v. 2022a, CPU Intel Core I7-9700 @ 3.00 GHz 8 Cores, 32 GB physical memory, NVIDIA card model GeForce RTX 3060 with 12 GB memory and 3584 GPU cores.

IV. Results

For evaluating our results, it was plotted the training progress for each model. With the fine-tuned network the validation images were classified, and then calculated the classification accuracy. With these data, it was created a confusion matrix to assess the results. These confusion matrixes are illustrated for two networks in Fig. 3.



(a)



(b)

Fig. 3. Confusion matrixes for a) googlenet b) nasnetlarge.

The corresponding training progress for these two models, are shown in Fig. 4.

At least 2 runs for each model were performed to verify consistency of the outcome; due to the long times that some models consume for training, only two runs were carried out for some of them, the rest were simulated up to 5 times. It is important to remark that, as each run uses random information for training (the selected images for training and validation, augmentation operations, and initial parameters for the replaced three layers), each time it is carried out training, the values vary, so we report the worst and the best case that we obtained with each network in terms of accuracy. This is shown in Table 3.

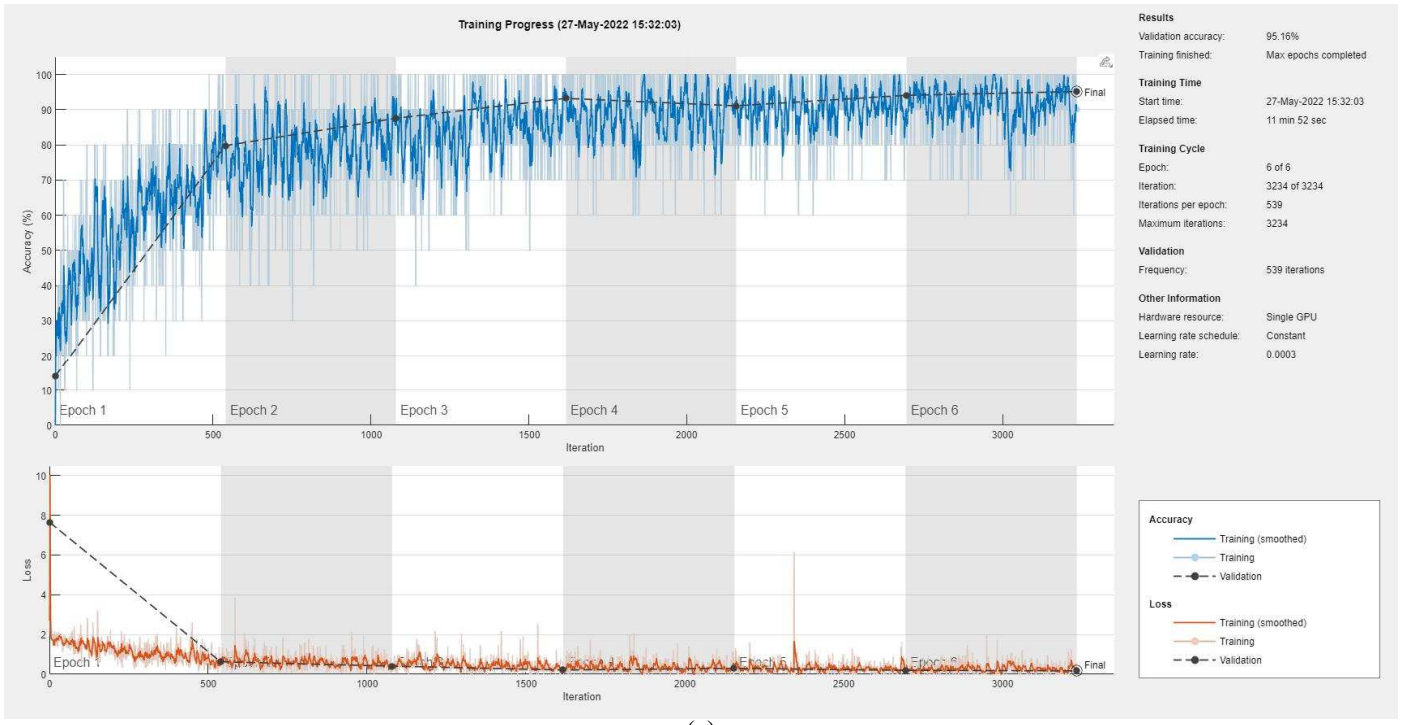
TABLE 3. Accuracy and run times for pre-trained networks (sorted by worst/best case of accuracy).

Network	Accuracy (%)	Run time
alexnet	85.90 90.70	4’20” 4’18”
darknet19	92.39 92.65	12’17” 12’12”
darknet53	91.96 92.99	40’50” 40’39”
densenet201	86.72 92.86	162’43” 161’29”
efficientnetb0	93.17 95.72	59’17” 58’40”
googlenet	90.79 95.16	11’57” 11’52”
inceptionresnetv2	96.11 96.28	145’53” 170’53”
inceptionv3	96.11 97.75	48’54” 48’56”
mobilenetv2	93.94 96.41	26’29” 26’17”
nasnetlarge	94.07 95.11	2597’51” 2559’49”
nasnetmobile	93.34 96.45	239’23” 234’20”
resnet101	92.86 95.24	60’55” 62’38”
resnet18	91.09 95.37	7’14” 7’14”
resnet50	92.95 94.25	25’41” 25’27”
shufflenet	94.42 95.42	23’59” 23’31”
squeezenet	89.49 90,79	4’59” 4’57”
vgg16	87.02 92.21	18’55” 18’27”
vgg19	80.10 88.75	21’56” 21’25”
xception	93.73 96.80	38’1” 38’3”

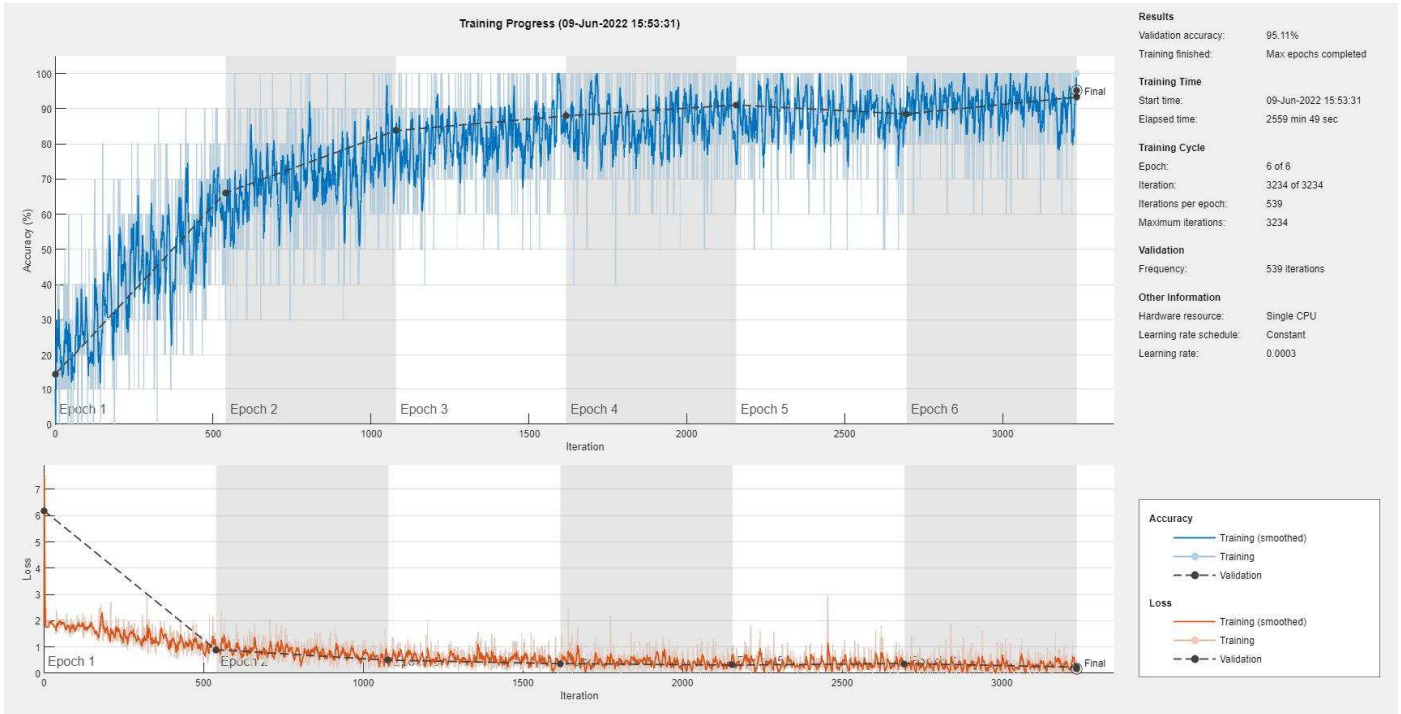
As we can observe in both confusion matrixes, A and f beats (f is termed f1 in the figure), present the smaller classification errors, whilst F and N beats show the biggest ones. This behavior was observed for most of the networks.

V. Conclusions

From results shown in Table 3, we can see that biggest networks take the longest training times but, counterintuitively do not always achieve the best accuracy values. *Nasnetlarge* network has the largest training times but their accuracy is comparable with *efficientnetb* that has considerably less



(a)



(b)

Fig. 4. Training progress for a) googlenet b) nasnetlarge.

parameters. Small networks show the worst performances, but they have short training times, so for problems when it should be necessary repeating several simulations, and it is not required a high performance, they could be an option.

We did not make any special adjustment for training parameters and take default values for all the cases, since our goal was to test all of the networks with the same conditions; however, to increase the performance of one network for a specific problem, it may be essential changing some parameters as training epochs, learning rate, optimization method, etc.

This is a preliminary study for comparing these nineteen models; due to the required time for simulating each network, we did not perform a larger number of runs, but in a subsequent

work, we consider producing an adequate number of runs so we can report a complete statistical analysis.

Finally, it is significant noticing that all of the networks had high accuracy for the proposed problem, even the worst of them reached above 80% in just a run of 6-epochs, so we can conclude that pre-trained networks represent a good option when it may be important saving time.

REFERENCES

[1] Bishop, Christopher M. 2006. "Pattern Recognition and Machine Learning". New York: Springer-Verlag. Available at https://docs.google.com/viewer?a=v&pid=sites&srcid=aWFtYW5kaS5ldXxpc2N8Z3g6MjViZDk1NGI1NjQzOWZiYQ.

[2] Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. "Deep Learning". Cambridge, 2017. MA: MIT Press. Available at https://www.deeplearningbook.org/.

[3] S. J. Pan and Q. Yang. "A survey on transfer learning". IEEE Transactions on knowledge and data engineering, 22(10) pp 1345–1359, 2010.

[4] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. "Decaf: A deep convolutional activation feature for generic visual recognition". In International conference on machine learning, pp. 647–655, 2014.

[5] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. "Overfeat: Integrated recognition, localization and detection using convolutional networks". arXiv preprint arXiv:1312.6229, 2013.

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

[7] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. "ImageNet: A large-scale hierarchical image database". In CVPR (2009). https://image-net.org/index.php.

[8] MathWorks, Pretrained Deep Neural Networks: https://www.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html;jsessionid=a8840636b34e3bc0d174df1d188d.

[9] Moody GB, Mark RG. "The impact of the MIT-BIH Arrhythmia Database". IEEE Eng in Med and Biol 20(3): pp. 45-50 (May-June 2001). (PMID: 11446209).

[10] Muhammad Asif Saleem et al. "Comparative Analysis of Recent Architecture of Convolutional Neural Network". Hindawi, Mathematical Problems in Engineering Volume 2022, Article ID 7313612, 9 pages. https://doi.org/10.1155/2022/7313612.

[11] D. Gupta, B. Bajpai, G. Dhiman, M. Soni, S. Gomathi, and D. Mane, "Review of ECG arrhythmia classification using deep neural network," Materials Today Proceedings, vol. 2022, pp. 2214–7853, 2021.

[12] T. Jun, Hoang Minh Nguyen, Daeyoun Kang, Dohyeun Kim, Daeyoung Kim, Young-Hak Kim, "ECG arrhythmia classification using a 2-D convolutional neural network", Computer Science, ArXiv, 2018.