Signal, Image and Video Processing (2022) 16:1945–1953
https://doi.org/10.1007/s11760-022-02155-w

ORIGINAL PAPER

A new transfer learning approach to detect cardiac arrhythmia from ECG signals

Mohebbanaaz[1] · L. V. Rajani Kumar[1] · Y. Padma Sai[1]

Received: 20 August 2021 / Revised: 14 November 2021 / Accepted: 17 January 2022 / Published online: 18 February 2022

Abstract
Deep Learning (DL) has turned into a subject of study in different applications, including medical field. Finding the irregularities in Electrocardiogram (ECG) is a critical part in patients' health monitoring. ECG is a simple, non-invasive procedure used in the prediction and diagnosis of Cardiac Arrhythmia. This paper proposes a new transfer learning-based end to end approach to automate the cardiac arrhythmia classification. The proposed approach begins with gathering ECG Dataset and extracting beats after ECG beat segmentation. Developing a Model from scratch is time-consuming, so the concept of transfer learning is used. For transferring the knowledge to our ECG classification domain, the last layers of the model are fine-tuned such that model becomes more domain-specific to our target ECG data. Three pre-trained Convolutional Neural Networks (CNNs), AlexNet, Resnet18, GoogleNet are explored, and then, our model is designed by block wise fine-tuning each layer with different model training parameters. To update the weights and offsets, Adaptive moment estimation, Root means square propagation and Stochastic gradient descent with momentum (SGDM) are three different optimizers used. Investigating the results obtained by training fine-tuned models, we select the model which gives the system's best accuracy. MIT-BIH arrhythmia database is considered in this study. Performance of each Fine-tuned Model is evaluated by calculating Precision, Recall, Specificity,F-score and Accuracy. Moreover, our proposed fine-tuned Deep-CNN Model is effective and outperformed the existing models in the literature with accuracy of 99.56%.

Keywords Deep learning · Convolutional neural network (CNN) · Fine-tuned deep-CNN · Transfer learning · Electrocardiogram (ECG)

1 Introduction

Heart attacks are the significant reason for death all around the world. 17.9 million individuals are deceased every year because of cardiac anomalies, assessing 31% of overall deaths [1]. Major part of cardiovascular deaths is due to cardiac arrhythmia. Cardiologists regularly prescribe patients with Arrhythmia to wear Holter for ceaseless observing of ECG for 24 h. As this recorded data are enormous, there is a need to arrange kind of every heartbeat naturally utilizing the computer-aided diagnosis tools [2]. The development and adoption of deep learning (DL) have flooded over the most recent couple of years. Conventional Machine learning procedures or kernel-based neural network (NN) needs domain experts to recognize significant features for investigation of abnormalities in ECG [2, 3]. Artificial Intelligence (AI) techniques have emerged as a boon in the advancement of technologies.

Various Scholars in the literature have proposed various techniques for cardiac arrhythmia classification [4]. Machine learning methods needs features for classification. Hence, significant features need to be extracted, which without a doubt needs a domain expert. DL models learn features in a gradual manner without the need of experts. These are the layered architecture that learns various significant features at various layers. All the initial input layers extract features, and these features are classified by output layers. Convolutional neural networks (CNN) are preferred in DL approaches for extracting significant features and classification [5, 6]. The Deep-CNN model comprises of at least one convolution layers. The convolution layer performs convolution by sliding the kernel or filter over the data. Each receptive field gener-

✉ Mohebbanaaz
mohebbanaaz@gmail.com

[1] Department of ECE, VNR Vignana Jyothi Institute of Engineering and Technology, JNTUH, Hyderabad, India

ates a feature map, which is added to obtain output feature maps.

Many approaches for classifying ECG beats utilizing CNN have been proposed. Kiranyaz et al. [7] proposed CNN to classify ECG beats utilizing time series data and further modified it to 2-D CNN. Rajendra et al. [8] solved dataset imbalance problem and did a five-class classification utilizing a nine-layer CNN. Zhai et al. [9] designed a CNN architecture with just seven convolution layers. Sellami and Hwang [10] introduced CNN with batch-weighted loss which gave incredible results while training but failed to produce a better accuracy. Wenhan liu et al. [11] used a patient specific approach and developed CNN to learn intrinsic features. The CNN proposed by Li [12] performed well for five class classification but the model has some cons while training from scratch. W. Yin et al. [13] proposed an integrated UW radar-based CNN. This model classified ECG data even in moving state. Using the concept of transfer learning CNN model was proposed by Salem [14]. The model performed well, but only four classes of arrhythmia were considered.

A DL model learns the basic features adaptively and creates end to end DL framework. It takes the ECG signal, extracts intrinsic features and the predicts cardiac arrhythmia [15–18]. The effect of noise when DL model is used was studied using a nine-layer CNN [19]. Results show that when DL models are trained from scratch the effect on noisy signal do not have much impact on classification results. For extracting significant feature maps time–frequency spectral entropy can be used [20]. Real time detection of abnormal cardiac rhythms using a robust CNN [21] may help cardiologist in early diagnosis. Detection myocardial infraction at an early stage can save life of diseases [22, 23]. When the dataset is large the performance of model is improved [24, 25]. The major challenge is to collect patients' data, which can be time-consuming and expensive. In the absence of real-time data, most widely used public database is MIT-BIH arrhythmia database [26].

In this study, our main motto is to detect cardiac arrhythmia from ECG signals. Noise is removed from signals using discrete wavelet transform (DWT) [27, 28]. ECG beats are segmented and labelled according to the annotations mentioned in database [29]. This labeled database is used for further processing. Using concept of transfer learning, deep learning models are adapted to our ECG signal database. These models are fine-tuned with different model training parameters. Further, these fine-tuned models are validated by testing for detection of cardiac arrhythmia.

The rest of work is systematized as follows. Section 2 defines our proposed methodology which includes preprocessing data extraction of feature maps and classification with fine-tuned models. In Sect. 3, investigative results are presented and compared with existing state-of-the-art model. Section 4 concludes our work.

2 Methodology

Transfer learning approach helps us to transfer the learned knowledge of pre-trained models and helps us to collect information required to handle new data. In this study, we explored DL models like AlexNet, ResNet18 and GoogleNet, utilizing MIT-BIH datasets and extract intrinsic features by applying a block-wise fine-tuning strategy to each layer. Utilizing a fully-connected layer along with SoftMax layer, classification of these features is done. Figure 1 represents the block diagram of proposed methodology. The proposed strategy begins with gathering signals from MIT-BIH Dataset, preprocessing the data to extract ECG beats, Feature extraction using Fine-tuned Deep Learning Models and Classification of ECG beats.

2.1 Pre-processing

Signals are taken from MIT-BIH Arrhythmia Database [1]. It consists of 48 records, each with a duration of 30 min. All records together contain 16 types of beats; of which one is normal, 14 are arrhythmia classes and one unclassified beat. Among all, Records 102, 104, 106, 107, 109, 111, 118, 124, 200, 207, 208, 212, 214, 217, 223, 231, 232, 233 are considered. These 18 Records (captured from 18 different patients) are sufficient to extract required type of ECG beats. Beats considered in these records are Normal Beats (N), Left Bundle Branch Block Beats (L), Right Bundle Branch Block Beats (R), Premature Ventricular contraction Beats (V), Paced Beats (P), Atrial Premature Beats (A) and Fusion of Normal and Paced Beats (F).

In this work, Noise removal is done using Discrete wavelet Transform (DWT) filters. DWT decomposed signal into low- and high-frequency components commonly known as approximation and detailed coefficients. Let $z(j)$ be the signal which decomposes as follows

$$z(j) = z_{l_0}(j) + \sum_{l-\infty}^{l_0} d_l(j) \tag{1}$$

where $z_{l_0}(j)$ denotes approximated coefficients, l_0 is the scaling factor and $d_l(j)$ are the detailed coefficients.

For finding approximation coefficients of next higher level of decomposition, we iterate as

$$z_l(j) = z_{l+1}(j) + d_{l+1}(j) \tag{2}$$

each level of decomposition, input coefficients given is approximation coefficients obtained by previous level. Daubechies wavelet 6 (db6) is used to decompose signal into six levels. Coefficients are obtained from third level to sixth level and retained and signal is reconstructed using these coefficients.

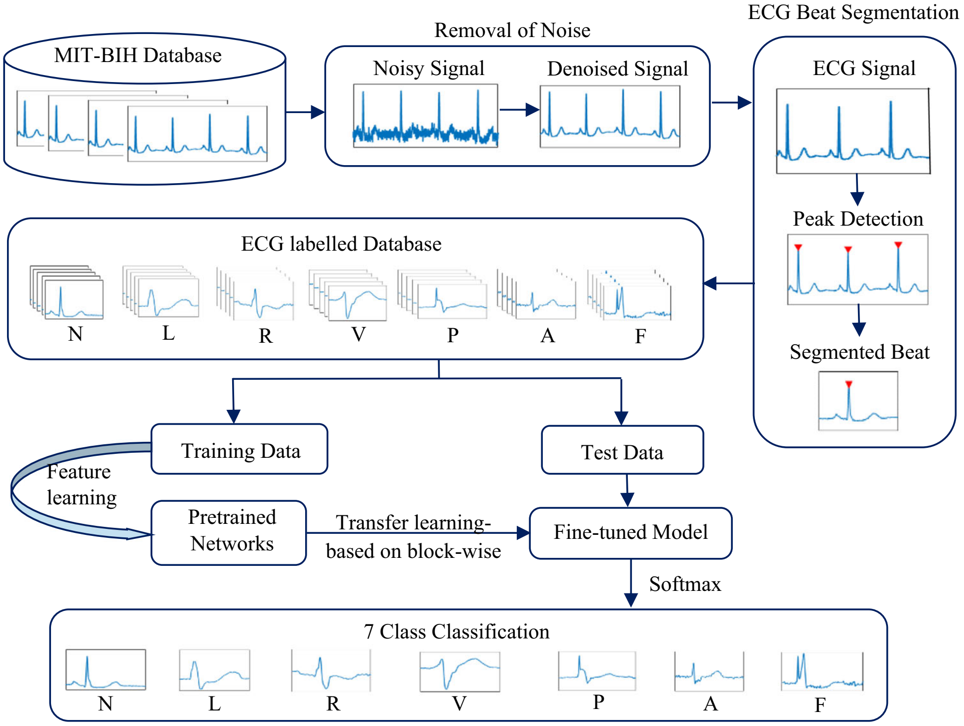


Fig. 1 Block Diagram of Proposed Approach

Noise-free ECG Signals are considered to extract ECG beats. Pan Tompkins algorithm is used to find R peaks [31]. Subtracting the consecutive R peaks gives us the RR interval. For segmenting ECG beats 30 s signal is considered at a time and average RR interval is found. An onset duration of one-third of average RR interval and offset duration of two-third of average RR interval is selected to segment one ECG beat.

Results obtained after Removal of Noise and Beat segmentation is shown in Sect. 3.1.

2.2 Feature extraction and classification through deep learning

DL models extracts features by using Convolutional Layers. These filters in these layers convolved with different kernels to generate a tensor of features. The filter will move from one point to next point depending on stride. In order to cover the entire beat, we use zero padding to filters. These tensor of features generated are activated using Rectified Linear Unit (ReLU) function. Let the convolution be defined as

$$x_k^l = f\left(\sum_{i \in M_k} x_i^{l-1} * w_{ik} + b_k\right) \tag{3}$$

M_K denotes the kernels used with different kernel size of 5 or 3. At layer$l$, x_k^l denotes output of k^{th} neuron, b_k is the bias of neuron$k$, w_{ik} is the weight kernel between neuron i at layer $l-1$ and neuron k at layer l. $f(.)$ denotes ReLU activation.

The model is trained until the loss function is minimized. The loss function is defined as

$$L = -\frac{1}{N}\sum_{n=1}^{N}\left[y_n \log(\widehat{y_n}) + (1-y_n)\log(1-\widehat{y_n})\right] \tag{4}$$

N denotes total neurons, y_n is the actual beat, $\widehat{y_n}$ is the predicted beat. To fine-tune the model, best hyper-parameter values and along with some regularization tricks is used to obtain ideal results. A batch of beats is given for training utilizing gradient descent algorithm. It optimizes the objective function $J(\theta)$ with model constraint $\theta \in R^d$. Gradient is defined as

$$g = \frac{1}{m}\nabla_\theta \sum_i L\left(f\left(x^{(i)};\theta\right), y^{(i)}\right) \tag{5}$$

where m denotes batch size, θ denotes the updating parameter, $f\left(x^{(i)};\theta\right)$ is predicted beat, $y^{(i)}$ is actual beat and $L(.)$ is the Loss function.

SGDM, ADAMs and RMSprop are optimizers used to update weights and offsets. SGDM accelerates a descent inappropriate path to reduce the oscillation. This is done by adding previous step update vector γ with learning rate η.

$$\begin{aligned} v_t &= \gamma v_{t-1} + \eta \nabla_\theta J(\theta) \\ \theta_{t+1} &= \theta_t - v_t \end{aligned} \tag{6}$$

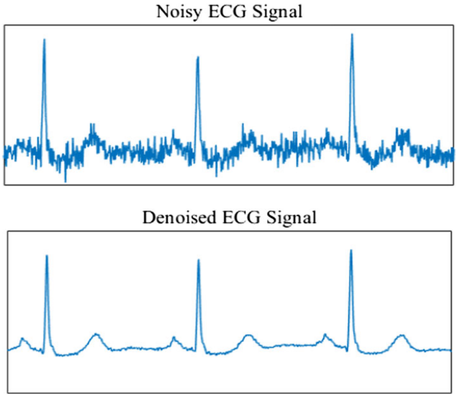


Fig. 2 Removal of Noise from ECG signal

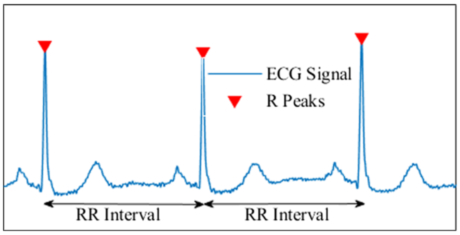


Fig. 3 ECG peak detection and RR interval

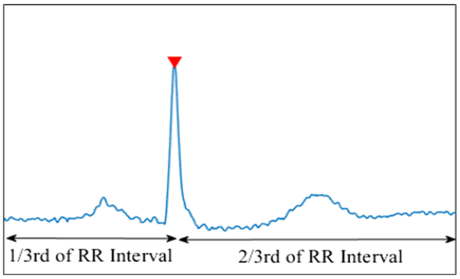


Fig. 4 Segmented ECG beat

where v_t is the momentum to be updated and θ_{t+1} is updated parameter.

RMSprop adaptively uses exponential decaying average to converge and is given as

$$E\left|g^2\right|_t = \gamma E\left|g^2\right|_{t-1} + (1-\gamma)g_t^2$$
$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E\left|g^2\right|_t + \in}} g_t \tag{7}$$

where $E\left|g^2\right|_t$ is the running average, $g_t = \nabla_\theta J(\theta)$ and θ_{t+1} is updated parameter.

Adams optimizer calculates adaptive learn rates. First moment (mean) m_t and second moment (variance) v_t estimates are updated as

$$m_t = \beta_1 m_{t-1} + (1-\beta_1)g_t$$
$$v_t = \beta_2 v_{t-1} + (1-\beta_2)g_t^2 \tag{8}$$

Corrections made to m_t and v_t to get exact mean $\widehat{m_t}$ and variance $\widehat{v_t}$ is given as

$$\widehat{m_t} = \frac{m_t}{\left(1-\beta_1^t\right)}$$
$$\widehat{v_t} = \frac{v_t}{\left(1-\beta_2^t\right)} \tag{9}$$

Using mean $\widehat{m_t}$ and variance $\widehat{v_t}$ the parameter θ is updated with step $\in$ as

$$\theta_{t+1} = \theta_t - \frac{1}{\sqrt{\widehat{v_t}} + \in}\widehat{m_t} \tag{10}$$

The final layer is the fully connected layer (FC) associated along with SoftMax layer. Three CNN architectures AlexNet, Resnet18 and GoogleNet are explored with different model training parameters. AlexNet model contains five convolutional layers and three fully connected layers. GoogleNet Model introduced a new inception layer that concatenated different filters into a single filter. It has two convolutional layers, nine inception layers and one fully connected layer. Resnet18 model has one convolution layer, eight Residual blocks and one fully connected layer. Each residual block internally has two convolution layers, i.e., ResNet18 has a total of 18 layers.

3 Results and discussions

The proposed model is simulated and tested on a system with Intel core i7-9th Gen, 3.6 GHz CPU, 16 GB RAM, and NVIDIA GeForce GTX 1060 8 GB GPU.

3.1 Pre-processing data

Noises that affect most frequently are baseline drifts, motion artifacts, power line interference and electromyography (EMG) noise. Noise is removed using DWT as mentioned in Sect. 2.1. Simulated result of noise removal is shown in Fig. 2.

For segmenting ECG beats 30 s signal is considered at a time and average RR interval is found (Fig. 3). An onset duration of one-third of average RR interval and offset duration of two-third of average RR interval is selected to segment one ECG beat as shown in Fig. 4.

A total of 40,213 ECG beats are extracted of which 27,344 beats are training beats, 4826 beats are validation beats and 8,043 beats are testing beats. Table 1 depicts the summary of total beats.

Table 1 Total beats extracted from MIT-BIH Arrhythmia Database

Beat Type	Total Beats	Training Beats	Validation Beats	Testing Beats
N	10,838	7370	1300	2168
L	8075	5491	969	1615
R	7259	4935	872	1452
V	4334	2947	520	867
P	7028	4779	843	1406
A	1697	1154	204	339
F	982	668	118	196
Total ECG Beats	40,213	27,344	4826	8043

Table 2 Model training parameters considered

Parameters	Values
Optimizers	SGDM, RMSprop, ADAMs
Pooling method	Maximum pooling and average pooling
Mini batch size	32,64,128
Initial learn rate	0.0001, 0.001, 0.01
Drop factor of learn rate	0.1,0.2,0.5
Maximum epochs	6,8,10

3.2 Fine-tuning CNN models

The Fine-tuned DL models are considered for identifying normal ECG beats and ECG beats with cardiac Arrhythmia. Pre-trained Model AlexNet, ResNet18, GoogleNet are fine-tuned to get system's best accuracy. Table 2 represents various model training parameters considered to fine-tune a model to achieve best outcome.

The optimum network configuration parameter is found by block wise fine-tuning with various model training parameters. The performance changes which occur due to change in batch size was observed. High batch sizes take less time to train but are very prone to overfitting. Low batch sized takes more time to train but improves performance. A batch size of 128 was considered, and then, it was slowly reduced to check performance improvement. Reduction in batch size beyond 32 did not show any significant improvement in performance. Hence, a batch size of 32 is considered. Different initial learn rate like 0.01, 0.001, 0.0001 were considered while fine-tuning. Lower the initial learn rate, higher the performance. So, an initial learn rate of 0.0001 is considered. Sharp features are not identified by average pooling. So, we here used maximum pooling in fine-tuning. Different drop factor of 0.1, 0.2 and 0.5 were considered. Increasing the drop factor more than 0.5 did not improve performance of model. Hence drop factor of 0.5 with initial learn rate 0.0001 is fixed for GoogleNet, ResNet18. As AlexNet has a few layers a drop factor of 0.2 is sufficient. So a drop factor of 0.2 is considered for AlexNet with initial learn rate 0.0001.

Table 3 Investigative results of fine-tuning AlexNet Model

Tested parameter	Optimizers		
	SGDM	Adam	RMSprop
Batch size	32	32	32
Maximum epochs	10	10	10
Initial learn rate	0.0001	0.0001	0.0001
Drop factor	0.2	0.2	0.2
Training time	37 min 10 s	40 min 8 s	36 min 58 s
Validation accuracy	**98.47%**	97.78%	97.14%

Table 4 Investigative results of fine-tuning ResNet18 Model

Tested parameter	Optimizers		
	SGDM	Adam	RMSprop
Batch size	32	32	32
Maximum epochs	10	10	10
Initial learn rate	0.0001	0.0001	0.0001
Drop factor	0.5	0.5	0.5
Training time	64 min 58 s	70 min 3 s	73 min 52 s
Validation Accuracy	97.24%	99.09%	**99.50%**

The CNN Models-AlexNet, ResNet18 and GoogleNet converged with firm accuracy after 10 epochs. So, 10 epochs have been considered for training models. Considering these parameters, Model was trained with different optimizers like SGDM, RMSprop and Adam.

Investigative Results obtained after fine-tuning the models are shown in Tables 3, 4 and 5 CNN Model-AlexNet achieved best training accuracy of 98.47% with training time of 37 min 10 s using SGDM optimizer. CNN Model-Resnet18 achieved best training accuracy of 99.50% with training time of 73 min 52 s using RMSprop optimizer. CNN Model-GoogleNet achieved best training accuracy of 98.57% with training time of 53 min 16 s using ADAMs optimizer. Figure 5 shows the training progress and loss function of the best Fine-tuned Models. This projected systems performance is evaluated by testing process using Test ECG beats.

Table 5 Investigative results of fine-tuning GoogleNet Model

Tested parameter	Optimizers		
	SGDM	Adam	RMSprop
Batch size	32	32	32
Maximum epochs	10	10	10
Initial learn rate	0.0001	0.0001	0.0001
Drop factor	0.5	0.5	0.5
Training time	86 min 0 s	77 min 16 s	99 min 3 s
Validation accuracy	96.97%	**98.57%**	97.70%

Table 6 Confusion matrix for N classes

True Class	Predicted Class			
	class 1	class 2	⋯	class N
Class 1	c_{11}	c_{12}	⋯	c_{1N}
Class 2	c_{21}	c_{22}	⋯	c_{11}
⋮	⋮	⋮	⋮	⋮
Class N	c_{N1}	c_{N1}	⋯	c_{NN}

3.3 Confusion matrix and performance measures

Confusion matrix represents the performance of model. For a N class classification the confusion matrix obtained is of N x N dimensions as shown in Table 6.

Figure 6 shows the confusion matrices obtained by our fine-tuned models. To evaluate the performance, performance measures like precision, recall. Specificity, F-score and Accuracy are calculated. Formulae for calculating performance measures are listed below

$$Precision(class\ i) = \frac{c_{ii}}{\sum_{j=1}^{N} c_{ji}} \quad (11)$$

$$Recall(class\ i) = \frac{c_{ii}}{\sum_{j=1}^{N} c_{ij}} \quad (12)$$

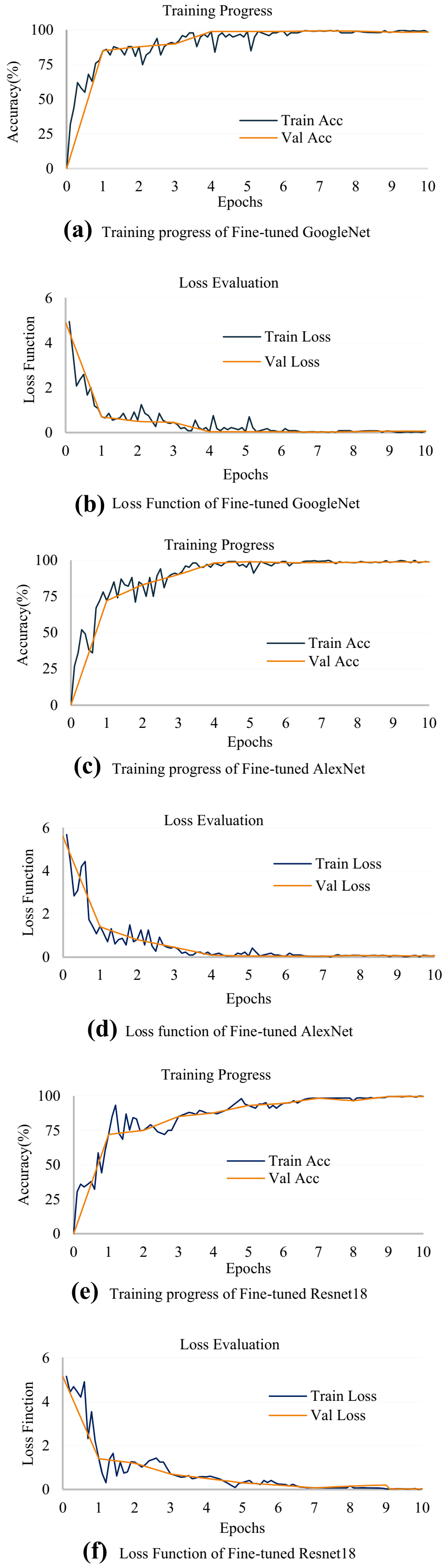
$$Specificity\,(class\ i) = \frac{\left(\sum_{j=1}^{N} c_{jj} - c_{ii}\right)}{\left(\sum_{j=1}^{N} c_{ji} - c_{ii}\right) + \left(\sum_{j=1}^{N} c_{jj} - c_{ii}\right)} \quad (13)$$

$$F - Score(class\ i) = \frac{2 \times c_{ii}}{\sum_{j=1}^{N} c_{ij} + \sum_{j=1}^{N} c_{ji}} \quad (14)$$

$$Overall Accuracy = \frac{\sum_{i=1}^{N} c_{ii}}{\sum_{i=1}^{N} \sum_{j=1}^{N} c_{ij}} \quad (15)$$

where Tp gives True positives, Tn gives True negatives, Fp gives False positives and Fn gives False negatives.

Performance measures of all the fine-tuned models is summarized in Tables 7, 8 and 9. From the tabulated results it is



(a) Training progress of Fine-tuned GoogleNet

(b) Loss Function of Fine-tuned GoogleNet

(c) Training progress of Fine-tuned AlexNet

(d) Loss function of Fine-tuned AlexNet

(e) Training progress of Fine-tuned Resnet18

(f) Loss Function of Fine-tuned Resnet18

Fig. 5 Training process and loss of Fine-tuned CNN Model

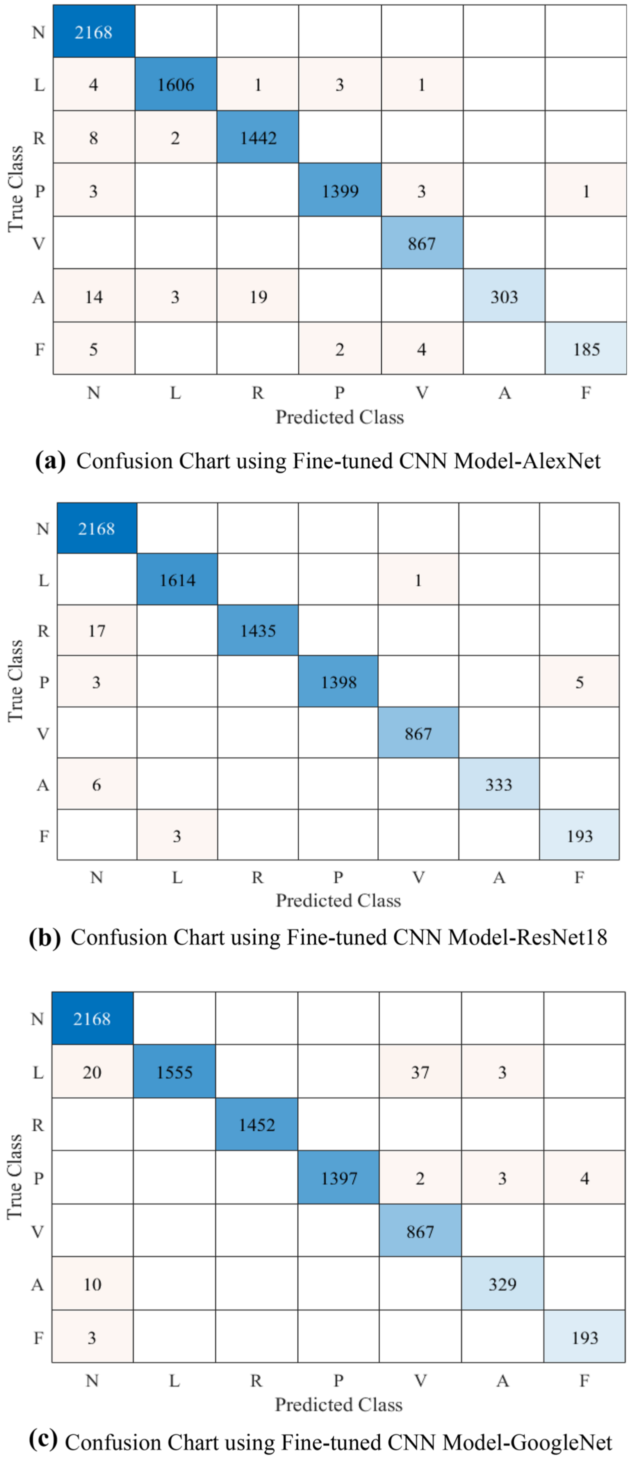


Fig. 6 Illustrations of Confusion Charts of Fine-tuned CNN Models. **a** Fine-tuned AlexNet Model. **b** Fine-tuned ResNet18 Model. **c** Fine-tuned GoogleNet Model

obvious that fine-tuned ResNet18 has attained highest test accuracy of 99.56%, while Fine-tuned AlexNet, GoogleNet, attained test accuracy of 99.09%, 98.98%.

Table 7 Performance Measures using Fine-tuned AlexNet Model

ECG beat type	Precision (%)	Recall (%)	Specificity (%)	F score (%)
N	99.45	100	100	99.22
L	99.69	99.44	99.86	99.56
R	98.63	99.31	99.84	98.97
V	99.08	100	100	99.54
P	99.64	99.50	99.92	99.57
A	100	89.38	99.57	94.39
F	99.46	94.38	97.74	96.85
Average	99.27	97.43	99.56	98.30

Overall test accuracy of model is 99.09%

Table 8 Performance Measures using Fine-tuned ResNet18 Model

ECG beat type	Precision (%)	Recall (%)	Specificity (%)	F score (%)
N	98.81	100	100	99.40
L	99.81	99.93	99.98	99.87
R	100	98.82	99.74	99.41
V	99.85	100	100	99.94
P	100	99.43	99.91	99.71
A	100	99.23	99.92	99.10
F	97.47	98.46	97.67	97.97
Average	99.42	99.26	99.60	99.34

Overall test accuracy of model is 99.56%

Table 9 Performance Measures using Fine-tuned GoogleNet Model

ECG Beat Type	Precision (%)	Recall (%)	Specificity (%)	F score (%)
N	98.50	100	100	99.24
L	100	96.26	99.07	98.10
R	100	100	100	100
V	95.69	100	100	97.80
P	100	99.36	99.09	99.67
A	98.20	97.05	99.88	97.62
F	97.97	98.46	97.71	98.21
Average	98.62	98.73	99.39	98.66

Overall Test Accuracy of Model is 98.98%

3.4 Comparisons of state-of-the-art techniques

Performance measures achieved by several state-of-the-art studies in the literature is given in Table 10. Our proposed fine-tuned ResNet18 model outperformed existing models in the literature with an accuracy of 99.56%.

Table 10 Comparison of State-of-the-art Models

Study	No. of beats	ECG classes	methods	Precision (%)	Recall (%)	Specificity (%)	F Score (%)	Accuracy (%)
Mousavi et al. [6]	750	5	CNN-LSTM	79.31	93.88	92.05	85.98	92.50
Wu et al. [16]	8,582	4	CNN	–	–	–	86.00	–
Acharya et al. [22]	50,728	5	CNN	–	93.71	92.83	–	93.53
Manish et al. [15]	50,977	5	DNN	94.85	85.63	98.27	89.65	98.10
Li et al. [17]	94,031	5	ResNet	–	99.38	98.14	–	99.38
Oh et al. [18]	83,648	5	Modified U-Net	–	–	–	–	97.32
Ozal et al. [24]	7,326	5	LSTM-WS	–	–	–	–	99.39
Yildirim et al. [25]	1000	13	Deep-CNN	92.52	93.52	99.61	92.45	95.20
Yildirim et al. [25]	1000	15	Deep-CNN	90.48	88.57	99.39	89.28	92.51
Yildrim et.al. [30]	10,436	7	DNN	80.31	80.15	98.72	80.04	92.24
Yildrim et al. [30]	10,588	4	DNN	95.78	95.43	95.43	95.57	96.13
Proposed Models	40,213	7	Fine-tuned AlexNet Fine-tuned ResNet18 Fine-tuned GoogleNet	99.27 99.42 98.62	97.43 99.26 98.73	99.56 99.60 99.39	98.30 99.34 98.66	99.09 **99.56** 98.98

4 Conclusions

In this Paper, AI-based fine-tuned CNN Models have been proposed to classify ECG beats. Pre-trained CNN Models AlexNet, ResNet18, GoogleNet are considered as backbone models in our work. We introduced a block-wise tuning strategy to optimize these backbone models and transfer learned knowledge to classify ECG beats. While fine-tuning each block it has been observed that the type of optimizer used while training has a large impact on the Performance of the Model. The time taken for training a Model depends on the depth of fine-tuned layers and the number of parameters considered while fine-tuning. It is observed that among all Models fine-tuned AlexNet has taken a minimum training time of 37 min 10 s using SGDM optimizer and produced test accuracy of 99.09%. Fine-tuned ResNet18 accomplished the highest accuracy of 99.56% using RMSprop optimizer with a Training time of 73 min 52 s. In future work, an Inference Model/working proof Hardware Model can be built considering real-time ECG Signals to detect abnormalities in ECG signals.

References

1. Benjamin, E.J., Virani, S.S., Callaway, C.W., Chamberlain, A.M., Chang, A.R., Cheng, S., Chiuve, S.E., Cushman, M., Delling, F.N., Deo, R., et al.: Heart disease and stroke statistics—2018 update: a report from the american heart association. Circulation **137**(12), e67–e492 (2018)
2. Kumari, L.V.R., PadmaSai, Y., et al.: FPGA based arrhythmia detection. Procedia Comput. Sci. **57**, 1 (2015)
3. Wang, J.S., Chiang, W.C., Hsu, Y.L., Yang, Y.T.C.: Ecg arrhythmia classification using a probabilistic neural network with a feature reduction method. Neurocomputing **116**, 38–45 (2013)
4. Sai, Y.P., et al.: A review on arrhythmia classification using ecg signals. In: 2020 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), pp. 1–6. IEEE (2020)
5. Mohebbanaaz, L.V., Sai, Y.P.: Classification of arrhythmia beats using optimized K-nearest neighbor classifier. In: Udgata, S.K., Sethi, S., Srirama, S.N. (eds.) Intelligent Systems. Lecture Notes in Networks and Systems. Springer, Singapore (2021).
6. Mousavi, S., Fotoohinasab, A., Afghah, F.: Single-modal and multi-modal false arrhythmia alarm reduction using attention-based convolutional and recurrent neural networks. PloS One **15**(1), e0226990 (2020).
7. Kiranyaz, S., Ince, T., Gabbouj, M.: Real-time patientspecific ecg classification by 1-d convolutional neural networks. IEEE Trans. Biomed. Eng. **63**(3), 664–675 (2015)
8. Acharya, U.R., Fujita, H., Lih, O.S., Hagiwara, Y., Tan, J.H., Adam, M.: Automated detection of arrhythmias using different intervals of tachycardia ecg segments with convolutional neural network. Inf. Sci. **405**, 81–90 (2017)
9. Zhai, X., Tin, C.: Automated ecg classification using dual heartbeat coupling based on convolutional neural network. IEEE Access **6**, 27465–27472 (2018)
10. Sellami, A., Hwang, H.: A robust deep convolutional neural network with batch-weighted loss for heartbeat classification. Expert Syst. Appl. **122**, 75–84 (2019)
11. Liu, W., Huang, Q., Chang, S., Wang, H., He, J.: Multiple-feature-branch convolutional neural network for myocardial infarction

diagnosis using electrocardiogram. Biomed. Signal Process. Control **45**, 22–32 (2018). https://doi.org/10.1016/j.bspc.2018.05.013

12. Li, D., Zhang, J., Zhang, Q., Wei, X.: Classification of ecg signals based on 1d convolution neural network. In: 2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom), pp. 1–6. IEEE (2017)
13. Yin, W., Yang, X., Zhang, L., Oki, E.: Ecg monitoring system integrated with ir-uwb radar based on cnn. IEEE Access **4**, 6344–6351 (2016)
14. Salem, M., Taheri, S., Yuan, J.S.: Ecg arrhythmia classification using transfer learning from 2-dimensional deep cnn features. In: 2018 IEEE Biomedical Circuits and Systems Conference (BioCAS), pp. 1–4. IEEE (2018)
15. Sharma, M., Acharya,U.R.: Automated heartbeat classification and detection of arrhythmia using optimal orthogonal wavelet filters. Inf. Med. Unlocked **16** (2019)
16. Wu, Q., Sun, Y., Yan, H., Wu, X.: Ecg signal classification with binarized convolutional neural network. Comput. Biol. Med. **121**, 103800 (2020)
17. Li, Z., Zhou, D., Wan, L.: Heartbeat classification using deep residual convolutional neural network from 2-lead electrocardiogram. J. Electrocardiol. **58**, 105–112 (2020)
18. Oh, S.L., Ng, E.Y., San Tan, R., Acharya, U.R.: Automated beat-wise arrhythmia diagnosis using modified u-net on extended electrocardiographic recordings with heterogeneous arrhythmia types. Comput. Biol. Med. **105**, 92–101 (2019)
19. Acharya, U.R., Oh, S.L., Hagiwara, Y., Tan, J.H., Adam, M., Gertych, A., San Tan, R.: A deep convolutional neural network model to classify heartbeats. Comput. Biol. Med. **89**, 389–396 (2017)
20. Asgharzadeh-Bonab, A., Amirani, M.C., Mehri, A.: Spectral entropy and deep convolutional neural network for ecg beat classification. Biocybern. Biomed. Eng. **40**(2), 691–700 (2020)
21. Kamaleswaran, R., Mahajan, R., Akbilgic, O.: A robust deep convolutional neural network for the classification of abnormal cardiac rhythm using single lead electrocardiograms of variable length. Physiological measurement **39**(3), 035006 (2018)
22. Acharya, U.R., Fujita, H., Oh, S.L., Hagiwara, Y., Tan, J.H., Adam, M.: Application of deep convolutional neural network for automated detection of myocardial infarction using ecg signals. Inf. Sci. **415**, 190–198 (2017)
23. Strodthoff, N., Strodthoff, C.: Detecting and interpreting myocardial infarction using fully convolutional neural networks. Physiological measurement **40**(1), 015001 (2019)
24. Yildirim, O.: A novel wavelet sequence based on deep¨ bidirectional lstm network model for ecg signal classification. Comput. Biol. Med. **96**, 189–202 (2018)
25. Yıldırım, O., P lawiak, P., Tan, R.S., Acharya, U.R.: Arrhythmia detection using deep convolutional neural network with long duration ecg signals. Computers in biology and medicine **102**, 411–420 (2018).
26. Moody, G.B., Mark, R.G.: The impact of the mit-bih arrhythmia database. IEEE Eng. Med. Biol. Mag. **20**(3), 45–50 (2001)
27. Mohebbanaaz, K.L.V.R., Sai, Y.P.: Classification of ECG beats using optimized decision tree and adaptive boosted optimized decision tree. SIViP (2021). https://doi.org/10.1007/s11760-021-02009-x
28. Nurmaini, S., Darmawahyuni, A., Sakti Mukti, A.N., Rachmatullah, M.N., Firdaus, F., Tutuko, B.: Deep learning-based stacked denoising and autoencoder for ecg heartbeat classification. Electronics **9**(1), 135 (2020)
29. Mohebbanaaz, S.Y.P., Kumari, L.R., et al.: Cognitive assistant deepnet model for detection of cardiac arrhythmia. Biomed. Signal Process. Control **71**, 103221 (2022)
30. Yildirim, O., Talo, M., Ciaccio, E.J., San Tan, R., Acharya, U.R.: Accurate deep neural network model to detect cardiac arrhythmia on more than 10,000 individual subject ecg records. Comput. Methods Programs Biomed. **197**, 105740 (2020)
31. Pan, J., Tompkins, W.: Real time algorithm detection for qrs. IEEE Trans. Eng. Biomed Eng **32**(3), 230–236 (1985)