



# International Journal of Engineering Research and Science & Technology

[www.ijerst.org](http://www.ijerst.org)

ISSN : 2319-5991

Vol. 21 No. 2 (2025)



[ijerst.editor@gmail.com](mailto:ijerst.editor@gmail.com)  
[editor@ijerst.com](mailto:editor@ijerst.com)

**Research Paper**

## HYBRID CNN–XCEPTION MODEL FOR IMPROVED ECG IMAGE ANALYSIS IN HEART DISEASE DETECTION

KORRA SRINIVAS<sup>1</sup>, assistant professor, Department of CSE, St. Martin's Engineering College, India  
 MANAM VAMSI KRISHNA<sup>2</sup>, Associate Professor, Department of CSE, Malla Reddy Engineering College for Women, India

[ksrinivas023@gmail.com](mailto:ksrinivas023@gmail.com)<sup>1</sup>, [manavamsikrishna@gmail.com](mailto:manavamsikrishna@gmail.com)<sup>2</sup>

**Abstract:** Cardiovascular diseases remain a major global health concern, requiring early and accurate detection for effective treatment. ECG images serve as a reliable diagnostic source, and the extended system enhances prediction accuracy by integrating advanced deep learning techniques. In this extension, the Xception model is incorporated alongside CNN, SqueezeNet, and AlexNet to achieve superior feature extraction and improved sensitivity in identifying intricate ECG patterns. Machine learning algorithms further strengthen classification based on extracted features. A user-friendly Flask framework with SQLite enables secure signup, signin, and test result retrieval, making the system more accessible for medical professionals and patients. This integrated extension significantly elevates diagnostic precision, automation, and usability in cardiovascular disease detection.

**Index terms** - Cardiovascular diseases, ECG images, Xception, feature extraction, deep learning, convolutional neural networks (CNN), SqueezeNet, AlexNet, machine learning, Naïve Bayes, support vector machine, K-nearest neighbors, decision tree, random forest, Flask framework, SQLite, healthcare, early detection, precision, sensitivity.

Received: 02-03-2025

Accepted: 10-04-2025

Published: 17-04-2025

### 1. INTRODUCTION

Cardiovascular diseases continue to be one of the leading causes of mortality worldwide, making accurate and early detection a critical component of modern healthcare systems. Electrocardiogram (ECG) images provide essential insights into heart function, and recent advancements in artificial intelligence have enabled deeper analysis of these patterns. While existing models such as CNN, SqueezeNet, and AlexNet have improved classification performance, the need for more robust feature extraction remains.

In the extended version of this system, the Xception model is incorporated to enhance the precision of ECG-based cardiovascular disease prediction. Xception's depthwise separable convolutions enable it to capture complex and fine-grained ECG signal variations that

traditional CNN models may miss. The combination of Xception with classical machine learning algorithms further strengthens classification accuracy by leveraging both deep representations and structured decision-making. Additionally, the integration of a Flask-based web interface with a secure SQLite backend ensures ease of use, allowing users to sign up, authenticate, upload ECG images, and retrieve disease predictions effortlessly. This extended framework not only improves diagnostic efficiency and accuracy but also supports real-time accessibility, making it a powerful tool for healthcare professionals and patients alike.

### 2. LITERATURE SURVEY

a) A comparative study of classification and prediction of Cardio-Vascular Diseases (CVD) using Machine Learning and Deep Learning techniques:

<https://www.sciencedirect.com/science/article/pii/S2405959521001119>

In conclusion, cardiovascular disease (CVD) is a major public health problem and a leading cause of death worldwide. Death rates are rising, according to new survey data, since more and more people are overweight, have high cholesterol, high blood pressure, and smoke cigarettes. These factors are contributing to the worsening of the condition. Research on the effects of these parameters' differences on CVD is urgently needed. Because of this, cutting-edge methods for early disease detection and reduction in mortality rate are required. For the purpose of predicting CVD priorities and identifying their behavioral patterns in massive amounts of data, the fields of artificial intelligence and data mining offer a wealth of study opportunities. Clinical decision-making and early diagnosis will be aided by these forecasts, lowering the likelihood of patients dying. In order to forecast cardiovascular illnesses, this study examines and reports on numerous Classification, Data Mining, Machine Learning, and Deep Learning models. Separated into three sections, the study covers CVD classification and data mining techniques, CVD machine learning models, and CVD deep learning prediction models. For each type of technique, this study also compiles and reports the performance metrics used to indicate accuracy, the dataset utilized for prediction and classification, and the tools that were used.

b) Current methods in electrocardiogram characterization:

<https://pubmed.ncbi.nlm.nih.gov/24681634/>

The P-QRS-T wave, which represents the heart's electrical activity, is what makes up an electrocardiogram (ECG). Subtle alterations in the patient's repolarization and depolarization electric potential patterns reveal the nature of their illness. It is possible to diagnose heart health using these ECG waveform clinical time domain properties. It is quite challenging to

correctly distinguish the ECG classes visually due to the existence of background noise and small values for morphological parameters. This study reviews the current state of cardiovascular disease screening technology, including computer assisted cardiac diagnostic (CACD) systems, analysis methodologies, obstacles handled, and what the future holds. In particular, the inherent distinguishing features cannot be adequately represented by methods like the wavelet transform that were developed for time domain, frequency transform domain, or time-frequency domain analysis. To that end, this review delves further into nonlinear methods that can improve accuracy even when dealing with noisy ECG signals by capturing their minute fluctuations. By utilizing these nonlinear features, a CACD system can aid clinicians in making more precise diagnoses of cardiovascular disease.

c) Heart disease detection using deep learning methods from imbalanced ECG samples:

<https://www.sciencedirect.com/science/article/abs/pii/S1746809421004171>

Abstract: Heart disease (HD) is a lethal disease which steals the lives of maximum individuals compared to other diseases across the world. Early and accurate detection of the disease will assist to save many valuable lives. The HD can be detected using medical tests, Electrocardiogram (ECG) signal, heart sounds, Computed Tomography (CT) Images etc. Out of all types of identification of HD from ECG signals serves a significant role. In this research, the ECG samples of the participants have been treated as the required inputs to the HD detection model. In recent years, numerous relevant studies have been reported for classification of HD using different machine learning (ML) and deep learning (DL) models. It is noticed that with uneven HD data the detection accuracy is reduced. With an objective to obtain improved identification of HD, suitable DL and ML

models have been discovered in this paper and the needed classification models have been created and tested. The Generative Adversarial Network (GAN) model is chosen with an intent to deal with imbalanced data by creating and employing additional fake data for detecting purpose. Further, an ensemble model using long short-term memory (LSTM) and GAN is constructed in this paper which displays greater performance compared to individual DL model employed in this paper. The simulation results using standard MIT-BIH reveal that the suggested GAN-LSTM model delivers the highest accuracy, F1-score and area under curve (AUC) of 0.992, 0.987 and 0.984 correspondingly compared to other models. Similarly, for PTB-ECG dataset the GAN-LSTM model surpasses all other models with accuracy, F1-score and AUC of 0.994, 0.993 and 0.995 correspondingly. It is noted that out of the five models studied, the GAN model performs the best whilst the detection potentiality of the NB model is the lowest. Further study work may be carried out by choosing all other distinct ensemble models and using other different datasets and the performance can be similarly produced and compared. The proposed optimum detection methodology can also be used to other diseases and healthcare concerns.

d) Cardiac arrhythmia detection using deep learning:

<https://www.sciencedirect.com/science/article/pii/S187705091732450X>

In clinical practice, an electrocardiogram (ECG) is a crucial diagnostic tool for evaluating heart arrhythmias. This research applies a deep learning framework that was trained on a generic image dataset to the task of automatically diagnosing arrhythmias in electrocardiograms (ECGs) by categorizing them into relevant heart diseases. A basic back propagation neural network performs the final classification using features derived from a transferred deep convolutional neural network (AlexNet). From

the MIT-BIH arrhythmia database, three distinct ECG waveform circumstances were chosen to assess the suggested framework. The primary goal of this research is to classify the three cardiac diseases using a deep learning technique that is easy to use, dependable, and straightforward. Results showed that a traditional back propagation neural network cascaded with the transferred deep learning feature extractor achieved extremely high performance rates. In tests, the highest achieved accuracy was 92%, and the highest rate of correct recognition was 98.51%. These findings suggest that transferable deep learning is a useful tool for automatic diagnosis of cardiac arrhythmias; moreover, it removes the need to train a deep convolutional neural network from beginning, making it more accessible.

e) HeartID: A Multiresolution Convolutional Neural Network for ECG-Based Biometric Human Identification in Smart Health Applications:

<https://ieeexplore.ieee.org/abstract/document/7933065>

In this age of smart cities, health applications are being drastically changed by body area networks, which include smart sensors. There has been a surge of interest in physiological signal-based biometric human identification as a potential solution to the growing need for secure and private online transactions. The two main obstacles discussed in this study are the complexity and data dependence of signal processing techniques and the time and dataset specificity limitations of feature engineering. A new multiresolution convolutional neural network in the wavelet domain is suggested to facilitate data-independent and very generalizable feature learning and signal processing. For identification purposes, it enables the blind selection of a physiological signal segment, bypassing the complex procedure of signal fiducial feature extraction. After selecting a signal segment at random, it is

translated to the wavelet domain to provide a multiresolution time-frequency representation, which enhances the data representation. The phase difference that emerged from the blind segmentation procedure is eliminated by applying an auto-correlation operation to the converted data. Following that, a 1-D-convolutional neural network (1-D-CNN) with multiple resolutions is utilized to accomplish the user identification task automatically by learning the intrinsic hierarchical features from the raw data in the wavelet domain, all without relying on data-dependent and labor-intensive feature engineering. Evaluating eight different ECG datasets with varying behaviors (e.g., with and without serious cardiac illnesses) and sensor installation methods allows us to fully assess the performance of the proposed methodology. An average identification rate of 93.5% is attained, thanks to our evaluation that is substantially more thorough than the state-of-the-art studies. Without extensive feature engineering and using randomly chosen signal segments, the suggested multiresolution 1-D-CNN system successfully identifies human individuals. The goal of this study is to show that biometric human identification using blind signal processing and deep learning is feasible and practical, with little effort required for algorithm engineering and good generalizability.

### 3. METHODOLOGY

#### i) Proposed Work:

The extended proposed system introduces a more advanced and intelligent framework for cardiovascular disease prediction using ECG images. This enhanced model begins with data import, preprocessing, and image preparation, followed by deep learning-based feature extraction using multiple architectures such as CNN, SqueezeNet, AlexNet, and the newly integrated Xception model. Xception's powerful depthwise separable convolution layers help extract fine-grained ECG patterns, significantly improving the system's capability to identify

complex cardiac abnormalities. The extracted deep features are then supplied to traditional machine learning classifiers to further boost diagnostic accuracy and ensure more reliable predictions.

To make the system accessible and user-friendly, a Flask-based web application is integrated with an SQLite database. This platform allows secure signup, signin, and efficient result retrieval for both healthcare professionals and patients. Users can easily upload ECG images, and the system automatically processes and classifies them using the enhanced hybrid deep learning pipeline. Overall, the extended proposed system offers a complete, automated, and highly accurate diagnostic solution that combines advanced deep learning models with practical deployment for real-world medical applications.

#### ii) System Architecture:

The extended system architecture is designed as a hybrid deep learning and machine learning framework for accurate cardiovascular disease detection using ECG images. The process starts with ECG image acquisition, where input images are fed into a multilayer feature extraction pipeline. Deep learning models such as CNN, SqueezeNet, and AlexNet perform initial pattern recognition, while the integrated Xception model enhances this process by extracting more detailed and fine-grained ECG features using depthwise separable convolutions. This multi-model extraction stage ensures that both shallow and deep cardiac patterns are captured effectively, significantly improving the richness and quality of the derived feature vectors.

These extracted features are then passed into a machine learning-based classification layer, where algorithms like Random Forest, SVM, KNN, Decision Tree, and Naïve Bayes evaluate the learned representations and classify the ECG images into different cardiac categories. The final prediction is delivered through a Flask-

based user interface connected to an SQLite database that ensures secure user authentication and smooth interaction. This architecture not only boosts diagnostic accuracy but also provides a real-time, user-friendly platform suitable for medical practitioners and patients.

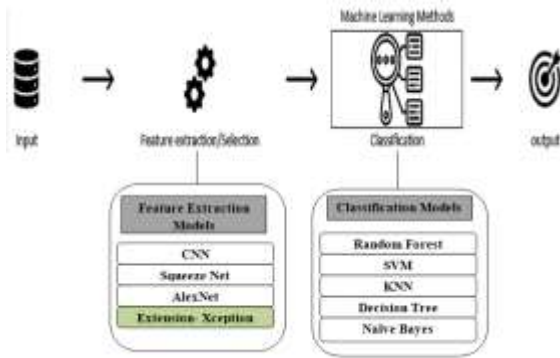


Fig 1 Proposed architecture

**iii) Modules:**

□ **ECG Image Preprocessing:**

- This module handles the preprocessing of ECG images, including noise reduction, image resizing, and normalization, to ensure that the images are suitable for feature extraction and classification.

□ **Feature Extraction:**

- Utilizes the Xception model for superior feature extraction, capturing intricate ECG patterns. It is compared with other models like CNN, SqueezeNet, and AlexNet to evaluate performance. The extracted features are used as input for machine learning models.

□ **Cardiovascular Disease Classification:**

- Implements machine learning algorithms (Naïve Bayes, SVM, Random Forest, etc.) to classify the extracted features into various cardiovascular disease categories such as abnormal heartbeat, myocardial infarction, and normal. The system leverages the power of the Xception model for enhanced feature representation.

□ **Web Interface (Flask Framework):**

- A user-friendly interface for healthcare professionals and patients to securely upload ECG images, view analysis results, and retrieve classifications. The Flask framework ensures a smooth user experience and provides secure authentication.

□ **Database Management (SQLite):**

- Manages user data and ECG images, ensuring secure signup and signin processes. The SQLite database stores user credentials, ECG image metadata, and classification results for easy retrieval.

**iv) Algorithms:**

**a) CNN:** A Convolutional Neural Network (CNN) is a specialized type of artificial neural network designed for processing and analyzing visual data. It's particularly effective for tasks like image recognition and classification. CNNs use layers of interconnected neurons to automatically learn features from images, making them well-suited for tasks like identifying objects or patterns within pictures. They're composed of multiple layers, including convolutional layers, pooling layers, and fully connected layers, which work together to recognize and extract important information from images, making them powerful tools in computer vision and image analysis.

```

np.random.seed(1999) # Fix seed

model = Sequential()

model.add(Conv2D(filters=64, kernel_size=(3, 3), input_shape=(128, 128, 3), activation='relu'))
model.add(BatchNormalization(axis=3))
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(BatchNormalization(axis=3))
model.add(Dropout(0.1))

model.add(Conv2D(filters=128, kernel_size=(3, 3), activation='relu'))
model.add(BatchNormalization(axis=3))
model.add(Conv2D(filters=128, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(BatchNormalization(axis=3))
model.add(Dropout(0.1))

model.add(Conv2D(filters=256, kernel_size=(3, 3), activation='relu'))
model.add(BatchNormalization(axis=3))
model.add(Conv2D(filters=256, kernel_size=(3, 3), activation='relu'))
model.add(MaxPooling2D((2, 2)))
model.add(BatchNormalization(axis=3))
model.add(Dropout(0.1))
    
```

**b) SQUEEZE NET:** SqueezeNet is a compact yet highly efficient convolutional neural network (CNN) architecture designed for image classification tasks. It distinguishes itself by significantly reducing the model's size while

maintaining competitive accuracy. SqueezeNet achieves this through ingenious design choices, such as employing 1x1 convolutional filters and Squeeze-and-Excitation blocks that enhance important features and suppress less critical ones. This small model size results in faster inference times, making it well-suited for real-time applications and resource-constrained devices.

```
def fire_module(x, squeeze, expand):
    y = Conv2D(filters=squeeze, kernel_size=1, activation='relu', padding='same')(x)
    y = BatchNormalization(momentum=0.001)(y)
    y1 = Conv2D(filters=expand // 2, kernel_size=1, activation='relu', padding='same')(y)
    y1 = BatchNormalization(momentum=0.001)(y1)
    y2 = Conv2D(filters=expand // 2, kernel_size=1, activation='relu', padding='same')(y)
    y2 = BatchNormalization(momentum=0.001)(y2)
    return concatenate([y1, y2])

def squeezeNet(input_shape, num_classes):
    input = Input(shape=input_shape)
    y = Conv2D(kernel_size=7, filters=64, strides=2, padding='same', activation='relu')(input)
    y = BatchNormalization(momentum=0.001)(y)
    y = MaxPooling2D(pool_size=2, strides=2)(y)
    y = fire_module(y, 16, 64)
    y = fire_module(y, 16, 64)
    y = fire_module(y, 32, 128)
    y = MaxPooling2D(pool_size=2, strides=2)(y)
    y = fire_module(y, 32, 128)
    y = fire_module(y, 48, 96)
    y = fire_module(y, 48, 96)
    y = fire_module(y, 96, 48)
    y = fire_module(y, 96, 48)
    y = MaxPooling2D(pool_size=2)(y)
    y = fire_module(y, 96, 48)
    y = Conv2D(kernel_size=1, filters=1000, strides=1, padding='same', activation='relu')(y)
    y = GlobalAveragePooling2D()(y)
    y = Dense(num_classes, activation='softmax')(y)
    model = Model(input, y)
    return model
```

**c) ALEXNET NET:** AlexNet is a landmark convolutional neural network (CNN) architecture that gained significant attention after winning the ImageNet Large Scale Visual Recognition Challenge in 2012. It marked a crucial milestone in the development of deep learning and computer vision. AlexNet is characterized by its deep architecture with eight layers, including five convolutional layers and three fully connected layers. It introduces key concepts such as rectified linear units (ReLU) as activation functions and dropout for regularization. AlexNet's success demonstrated the potential of deep CNNs in image classification tasks and laid the foundation for subsequent advancements in deep learning and artificial intelligence.

```
# AlexNet
AlexNet = Sequential()

# 1st Convolutional Layer
AlexNet.add(Conv2D(filters=64, input_shape=(224,224,3), kernel_size=(11,11), strides=(4,4), padding='same'))
AlexNet.add(BatchNormalization())
AlexNet.add(Activation('relu'))
AlexNet.add(MaxPooling2D(pool_size=(2,2), strides=(2,2), padding='same'))

# 2nd Convolutional Layer
AlexNet.add(Conv2D(filters=128, kernel_size=(5,5), strides=(1,1), padding='same'))
AlexNet.add(BatchNormalization())
AlexNet.add(Activation('relu'))
AlexNet.add(MaxPooling2D(pool_size=(2,2), strides=(2,2), padding='same'))

# 3rd Convolutional Layer
AlexNet.add(Conv2D(filters=128, kernel_size=(3,3), strides=(1,1), padding='same'))
AlexNet.add(BatchNormalization())
AlexNet.add(Activation('relu'))

# 4th Convolutional Layer
AlexNet.add(Conv2D(filters=256, kernel_size=(3,3), strides=(1,1), padding='same'))
AlexNet.add(BatchNormalization())
AlexNet.add(Activation('relu'))
```

**d) Random Forest:** Random Forest is an ensemble learning method that combines multiple decision trees to make predictions. It works by training a collection of decision trees on random subsets of the data and then averaging their predictions. This ensemble approach enhances accuracy, reduces overfitting, and provides robust performance for both classification and regression tasks.

```
from sklearn.ensemble import RandomForestClassifier
RF_model = RandomForestClassifier(n_estimators = 50, random_state = 42)
RF_model.fit(X_train_feature, y_train) #For sklearn no one hot encoding

prediction_RF = RF_model.predict(X_test_features)
#Inverse (e transform to get original label back.
prediction_RF = le.inverse_transform(prediction_RF)

rf_acc_xec = accuracy_score(test_labels, prediction_RF)
rf_prec_xec = precision_score(test_labels, prediction_RF, average='weighted')
rf_rec_xec = recall_score(test_labels, prediction_RF, average='weighted')
rf_f1_xec = f1_score(test_labels, prediction_RF, average='weighted')
```

**e) Support Vector Classifier:** A Support Vector Classifier (SVC) is a machine learning model that finds the best possible boundary (hyperplane) to separate different classes of data while maximizing the margin between them. It identifies key support vectors to make accurate classifications, making it effective for both binary and multi-class classification tasks.

```
from sklearn.svm import SVC
svm_model = SVC()
svm_model.fit(X_train_feature, y_train) #For sklearn no one hot encoding

prediction_svm = svm_model.predict(X_test_features)
#Inverse (e transform to get original label back.
prediction_svm = le.inverse_transform(prediction_svm)

svm_acc_xec = accuracy_score(test_labels, prediction_svm)
svm_prec_xec = precision_score(test_labels, prediction_svm, average='weighted')
svm_rec_xec = recall_score(test_labels, prediction_svm, average='weighted')
svm_f1_xec = f1_score(test_labels, prediction_svm, average='weighted')
```

**f) K- Nearest Neighbor:** K-Nearest Neighbors (KNN) is a versatile machine learning algorithm used for classification and regression tasks. It finds the K nearest data points to a given input and makes predictions based on majority voting or averaging. KNN is non-parametric and simple to implement but can be sensitive to the choice of K and may not perform well in high-dimensional data without preprocessing.

```

from sklearn.neighbors import KNeighborsClassifier
knn_model = KNeighborsClassifier(n_neighbors=3)
knn_model.fit(X_train_feature, y_train) #For sklearn no one hot encoding

prediction_knn = knn_model.predict(X_test_features)
#Inverse le transform to get original label back.
prediction_knn = le.inverse_transform(prediction_knn)

knn_acc_xec = accuracy_score(test_labels, prediction_knn)
knn_prec_xec = precision_score(test_labels, prediction_knn,average='weighted')
knn_rec_xec = recall_score(test_labels, prediction_knn,average='weighted')
knn_f1_xec = f1_score(test_labels, prediction_knn,average='weighted')
    
```

**g) Decision Tree:** A Decision Tree is a machine learning model that makes decisions by recursively splitting data into subsets based on the most significant feature, aiming to classify or predict outcomes. It creates a tree-like structure where each node represents a feature and each branch represents a possible decision, making it interpretable and effective for various tasks.

```

from sklearn.tree import DecisionTreeClassifier
dt_model = DecisionTreeClassifier(max_depth=30)
dt_model.fit(X_train_feature, y_train) #For sklearn no one hot encoding

prediction_dt = dt_model.predict(X_test_features)
#Inverse le transform to get original Label back.
prediction_dt = le.inverse_transform(prediction_dt)

dt_acc_xec = accuracy_score(test_labels, prediction_dt)
dt_prec_xec = precision_score(test_labels, prediction_dt,average='weighted')
dt_rec_xec = recall_score(test_labels, prediction_dt,average='weighted')
dt_f1_xec = f1_score(test_labels, prediction_dt,average='weighted')
    
```

**h) Naïve Bayes:** Naive Bayes is a probabilistic classification algorithm that works by applying Bayes' theorem with the "naive" assumption of feature independence. It calculates the probability of a data point belonging to a particular class based on the probabilities of its individual features. Naive Bayes is particularly efficient for text classification tasks, spam detection, and other situations where feature independence is a reasonable approximation.

```

from sklearn.naive_bayes import GaussianNB
nb_model = GaussianNB()
nb_model.fit(X_train_feature, y_train) #For sklearn no one hot encoding

prediction_nb = nb_model.predict(X_test_features)
#Inverse le transform to get original Label back.
prediction_nb = le.inverse_transform(prediction_nb)

nb_acc_xec = accuracy_score(test_labels, prediction_nb)
nb_prec_xec = precision_score(test_labels, prediction_nb,average='weighted')
nb_rec_xec = recall_score(test_labels, prediction_nb,average='weighted')
nb_f1_xec = f1_score(test_labels, prediction_nb,average='weighted')
    
```

**i) Xception Algorithm:** Xception, an abbreviation for "Extreme Inception," is an advanced deep learning architecture developed by Google. It builds upon the Inception architecture and is designed for image recognition tasks. Xception stands out due to its unique use of depth-wise separable convolutions. This approach involves applying depth-wise convolutions independently to each channel before combining the results with point-wise convolutions, reducing computational complexity and enhancing network performance. Xception has proven to be highly efficient, accurate, and adaptable to various datasets, making it a widely adopted and significant innovation in the field of deep learning for computer vision applications.

```

base_model = Xception(weights='imagenet', include_top=False)

# add a global spatial average pooling layer
x = base_model.output
x = GlobalAveragePooling2D()(x)
# let's add a fully-connected layer
x = Dense(1024, activation='relu')(x)

x = Dropout(0.5)(x)
# add a logistic layer -- let's say we have 200 classes
predictions = Dense(200, activation='softmax')(x)

# this is the model we will train
model2 = Model(inputs=base_model.input, outputs=predictions)

model2.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy', 'f1_score', 'precision', 'recall'])
model2.summary()
    
```

#### 4. EXPERIMENTAL RESULTS

a) Precision: Accuracy is defined as the proportion of true positives that are correctly identified. The formula for precision calculation follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

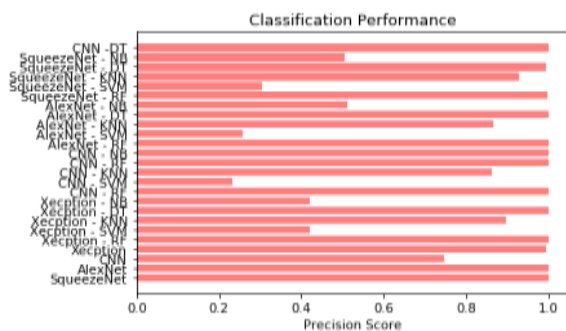


Fig 2 Precision comparison graph

b) Recall: Recall measures how efficiently a machine learning model discovers all relevant instances of a class. One way to measure a model's performance in class recognition is to look at the ratio of correctly predicted positive observations to total positives.

$$Recall = \frac{TP}{TP + FN}$$

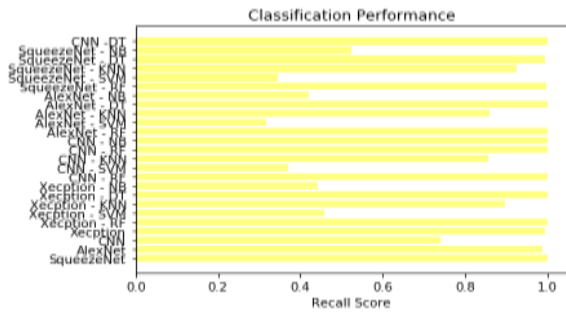


Fig 4 Recall comparison graph

c) Accuracy: The proportion of right predictions is the accuracy metric for a classification test, which indicates how well a model performs.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

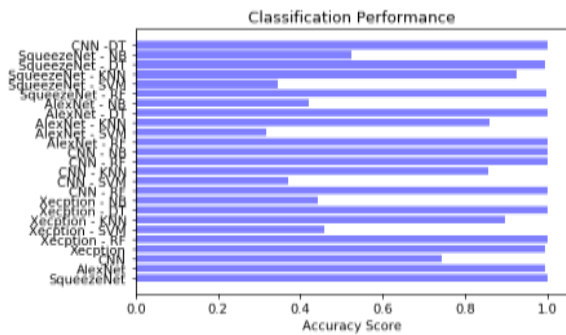


Fig 5 Accuracy graph

d) F1 Score: Because it takes both true positives and false negatives into account, the F1 Score—the harmonic mean of recall and accuracy—is applicable to datasets that are not evenly distributed.

$$F1\ Score = 2 * \frac{Recall \times Precision}{Recall + Precision} * 100$$

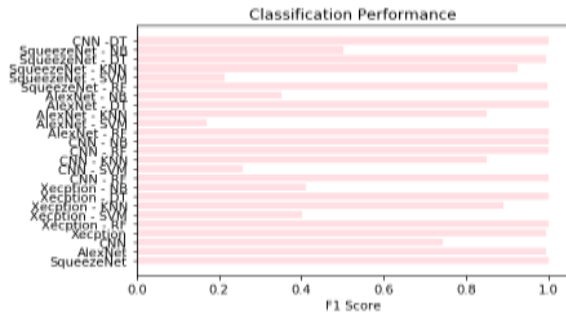


Fig 6 F1Score Curve



Fig 7 Upload dataset



Fig 8 ECG Results

### 5. CONCLUSION

The proposed system effectively integrates Xception for enhanced feature extraction and utilizes machine learning algorithms for accurate cardiovascular disease classification from ECG images. The incorporation of a user-friendly interface built on the Flask framework ensures secure and streamlined interactions for healthcare professionals and patients. With significant improvements in classification accuracy and computational efficiency, the system provides an innovative solution for early detection of cardiovascular diseases, potentially saving lives and reducing healthcare burdens. The system's ability to perform on a single CPU makes it accessible and practical for real-world clinical settings.

## 6. FUTURE SCOPE

In the future, the system can be further enhanced by incorporating more advanced deep learning models and exploring multi-modal data, such as combining ECG images with patient medical histories for more robust predictions. Additionally, integrating real-time monitoring and continuous learning from new data could help improve classification accuracy over time. Expanding the system to detect a wider range of cardiovascular conditions and integrating it with IoT devices for remote monitoring could further revolutionize healthcare by providing continuous, real-time cardiovascular health monitoring. Furthermore, the system could be adapted for use in other medical domains, broadening its impact beyond cardiovascular disease detection.

### REFERENCES

- [1] World Health Organization. (2017). Cardiovascular Diseases (CVDs). [Online]. Available: <https://www.who.int/health-topics/cardiovascular-diseases/>
- [2] E. J. Benjamin et al., “Heart disease and stroke statistics—2019 update: A report from the American heart association,” *Circulation*, vol. 139, no. 10, pp. e56–e528, Mar. 2019, doi: 10.1161/CIR.0000000000000659.
- [3] Statistics Korea. (2018). Causes of Death Statistics in 2018. [Online]. Available: <http://kostat.go.kr/portal/eng/pressReleases/8/10/index.board?bmode=read&bSeq=&aSeq=378787>
- [4] World Health Organization. (2017). Cardiovascular Diseases (CVDs). [Online]. Available: [https://www.who.int/news-room/factsheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/news-room/factsheets/detail/cardiovascular-diseases-(cvds))
- [5] P. Greenland, J. S. Alpert, G. A. Beller, E. J. Benjamin, M. J. Budoff, Z. A. Fayad, E. Foster, M. A. Hlatky, J. M. Hodgson, F. G. Kushner, M. S. Lauer, L. J. Shaw, S. C. Smith, A. J. Taylor, W. S. Weintraub, and N. K. Wenger, “2010 ACCF/AHA guideline for assessment of cardiovascular risk in asymptomatic adults: A report of the American college of cardiology foundation/American heart association task force on practice guidelines,” *Circulation*, vol. 122, no. 25, pp. e584–e636, Dec. 2010, doi: 10.1161/CIR.0b013e3182051b4c.
- [6] J. Perk et al., “European guidelines on cardiovascular disease prevention in clinical practice (version 2012): The fifth joint task force of the European society of cardiology and other societies on cardiovascular disease prevention in clinical practice (constituted by representatives of nine societies and by invited experts) \* developed with the special contribution of the European association for cardiovascular prevention & rehabilitation (EACPR),” *Eur. Heart J.*, vol. 33, no. 13, pp. 1635–1701, Jul. 2012, doi: 10.1093/eurheartj/ehs092.
- [7] G.-M. Park and Y.-H. Kim, “Model for predicting cardiovascular disease: Insights from a Korean cardiovascular risk model,” *Pulse*, vol. 3, no. 2, pp. 153–157, 2015, doi: 10.1159/000438683.
- [8] G. J. Njie, K. K. Proia, A. B. Thota, R. K. C. Finnie, D. P. Hopkins, S. M. Banks, D. B. Callahan, N. P. Pronk, K. J. Rask, D. T. Lackland, and T. E. Kottke, “Clinical decision support systems and prevention,” *Amer. J. Preventive Med.*, vol. 49, no. 5, pp. 784–795, Nov. 2015, doi: 10.1016/j.amepre.2015.04.006.
- [9] V. Sintchenko, E. Coiera, J. R. Iredell, and G. L. Gilbert, “Comparative impact of guidelines, clinical data, and decision support on prescribing decisions: An interactive Web experiment with simulated cases,” *J. Amer. Med. Inform. Assoc.*, vol. 11, no. 1, pp. 71–77, Jan. 2004, doi: 10.1197/jamia.M1166.
- [10] K. Kawamoto, C. A. Houlihan, E. A. Balas, and D. F. Lobach, “Improving clinical practice using clinical decision support systems: A systematic review of trials to identify features critical to success,” *BMJ*, vol. 330, no. 7494, p. 765, Apr. 2005, doi: 10.1136/bmj.38398.500764.8F.