Information Sciences Letters
An International Journal

http://dx.doi.org/10.18576/isl/120755

Evaluation of Pre-Trained CNN Models for Cardiovascular Disease Classification: A Benchmark Study

*Sami Alrabie and Ahmed Barnawi**

Department of the Information Technology, Faculty of Computing and Information Technology, Jeddah 21589, King Abdul Aziz University, Saudi Arabia

Received: 19 Mar 2023, Revised: 1 Jun. 2023 Accepted: 13 Jun. 2023.
Published online: 1 Jul. 2023.

Abstract: In this paper, we present an up-to-date benchmarking of the most commonly used pre-trained CNN models using a merged set of three available public datasets to have a large enough sample range. From the 18th century up to the present day, cardiovascular diseases, which are considered among the most significant health risks globally, have been diagnosed by the auscultation of heart sounds using a stethoscope. This method is elusive, and a highly experienced physician is required to master it. Artificial intelligence and, subsequently, machine learning are being applied to equip modern medicine with powerful tools to improve medical diagnoses. Image and audio pre-trained convolution neural network (CNN) models have been used for classifying normal and abnormal heartbeats using phonocardiogram signals. We objectively benchmark more than two dozen image-pre-trained CNN models in addition to two of the most popular audio-based pre-trained CNN models: VGGish and YAMnet, which have been developed specifically for audio classification. The experimental results have shown that audio-based models are among the best-performing models. In particular, the VGGish model had the highest average validation accuracy and average true positive rate of 87% and 85%, respectively.

Keywords: Heart sound, Heart Diseases, CVDs, Convolutional Neural Network (CNN), Image pre-trained model, Deep Learning, Transfer learning, Sound pre-trained models.

1 Introduction

The most important organ in the human body is the heart, as all the body's organs receive blood from the heart. Cardiovascular diseases (CVDs) currently pose the greatest threat to human health globally. According to a World Health Organization (WHO) report, 17.9 million people died from CVDs in 2019, accounting for 32 percent of all global deaths [1].

Numerous techniques are used to diagnose CVDs. The auscultation of cardiac sounds with a stethoscope is one of the most commonly used techniques. The initial step toward confirming diagnoses of cardiovascular illness is auscultation diagnosis, which is a well-established procedure in the field of medicine. Although this approach is quick and simple to use, it is also challenging in that it requires extensive knowledge and experience. Primary care providers have accuracy rates of 20% to 40% [2]. These difficulties arise because each heart sound and murmur have its own unique properties, such as intensity, sharpness, pitch, radiation, auscultation location, and the start of the cardiac cycle [3]. Figure 1 depicts a time-domain example of both typical and abnormal phonocardiogram signals. Every cardiac cycle is represented by the normal heart sounds S1 and S2 on the normal phonocardiogram signal. An abnormal phonocardiogram signal suggests that each cardiac cycle's S1 and S2 heart sounds are preceded by a murmur. Recently, advancements in computing have led to the rapid expansion of healthcare applications. Cardiac applications aid in the care of heart health. They offer numerous benefits, including usability, self-care, monitoring, saving time for clinical visits, and quick medical intervention for treatment. Heart sounds and signals provide information on many heart diseases. An automated system capable of interpreting heart sounds could be utilized to test for CVDs early and effectively as well as to manage the disease's progress. Algorithms can be used to move the signal analysis accountability from the doctor toward technology [4].

In general, a heart sound analysis system consists of four steps. Denoising is the process of removing undesirable components from the PCG signals, such as background noise during heart sound recordings. The step of sound segmentation consists of identifying the boundaries between the primary cardiac sounds (S1, S2, S3, and S4) and murmurs. The feature extraction stage is the process of reducing the high dimension of the PCG signal to the low dimension and preparing it for use by machine learning algorithms. The signal's features can be extracted from a wide

*Corresponding author e-mail: ambarnawi@kau.edu.sa

range of domains, including time, frequency, time-frequency, and depth domain, and there are numerous feature extraction techniques, including mel-frequency of cepstral coefficients (MFCCs) and continuous wavelet. The final phase is the classification process, which distinguishes normal PCG signals from aberrant categories. All steps are critical to the accuracy of the model [5].

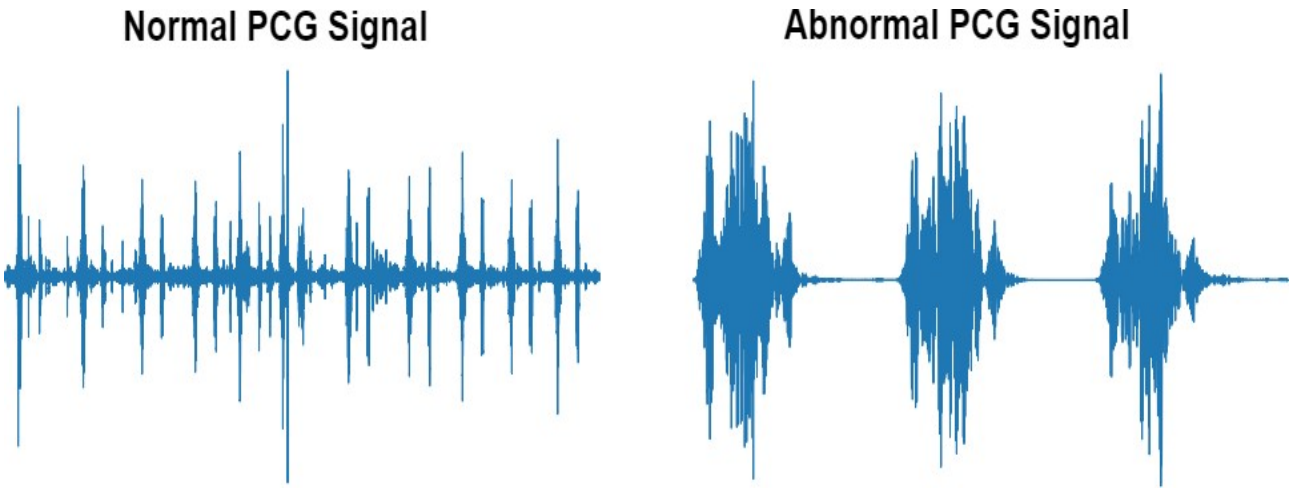

Fig. 1: An example of normal and abnormal phonocardiogram signals for heart sounds.

Researchers have put extensive effort into creating algorithms that automatically classify heart sounds with high accuracy. K-nearest neighbors (KNN), random forest, support vector machine (SVM), and decision tree were used as traditional machine learning techniques. They also used deep learning methods such as convolution neural networks (CNNs), recurrent neural networks (RNNs), and deep neural networks (DNNs). Recent advances in medical big data and artificial intelligence technologies have increased the emphasis on the development of deep learning methods for categorizing heart sounds [6], [7], [8].

Transfer learning is a way to reuse pre-trained CNN models and their weights that have been trained on very big datasets. Image pretrained CNN models have trained on the ImagNet dataset, which has over one million images for 1000 classes. Image pre-trained CNN models are available and can be employed for feature extraction and fine-tuning. Examples of image pre-trained CNN models are VGG family models, GoogleLetNet, InceptionV3, and residual network ResNet50. Image pre-trained CNN models were applied for heart sounds classification before the emergence of audio pre-trained models [9].

Currently, large audio datasets like AudioSet are being used to construct audio pre-trained CNN models. The AudioSet dataset has nearly two million audio files and 632 audio event classes. Audio pre-trained CNN models have been used for audio classification tasks such as bird audio detection, audio tagging, music genre classification, music mood classification, and environmental sound classification tasks. Audio pre-trained CNN algorithms such as VGGish, Yamnet, look listen and learn (L3), TRILL (TRIpLet loss network), and heart sound classification can be developed with PANNs (large-scale pretrained audio neural networks for audio pattern recognition) [10].

The availability of datasets that are standardized, high-quality, thoroughly vetted, and well documented encourages the advancement of algorithms for the study and analysis of heart sounds. Presently, the available and accessible public heart sound dataset recordings are the PhysioNet, PASACAL A and B, unnamed/GitHub and Sounds Shenzhen (HSS) datasets. Table 1 shows the technical details of the available public heart sound datasets. These datasets are limited in their number of recordings or classes. Other researchers have used their own private datasets [12].

In this study, we provide a phonocardiogram recognition benchmark using a combination of the image pre-trained models and popular sound pre-trained models with three merged public heart sound datasets: the Physionet/CinC 2016 dataset [13], the PASCAL Heart SoundChallenge (PHSC11) dataset [14] and a dataset collected from public databases [15].

Our primary objective is to suggest strong classification outcomes as a starting point reference benchmark for upcoming CVD recognition research and to simultaneously test the most appropriate model for heart sounds PCG classification. Can audio pre-trained CNN models accomplish greater CVD recognition accuracy than image pre-trained CNN models?

The rest of this study is arranged as follows. In Section 2, related work is presented. In Section 3, we show the proposed benchmark process. In Section 4, the experiments and results are provided. In Section 5, the conclusion is presented.

Table 1: Technical details of the available public heart sounds datasets

Ref	Dataset	Class	Recordings numbers	Total	Recordings lengths (s)	Sampling frequency	Acquisition Device	Limitation
[13]	PhysioNet	Normal	2575	3240	5 -120 s	2KHz	Digital stethoscope	Lacks other heart diseases
		Abnormal	655					
[14]	Pascal - A	Normal	45	167	1- 30s	44.1 KHz	istethoscope Pro iPhone app	Small number of recordings Lacks other heart diseases
		Murmur	48					
		Extrahs	27					
		Artifact	56					
[14]	Pascal - B	Normal	167	279	1-30s	44 KHz	Digital stethoscope	Lacks other heart diseases
		Murmur	69					
		extrasystole	39					
[15]	GitHub\unnamed	Normal	200	1000	Roughly 3s	8KHz	Collected from different sources	Lacks other heart diseases Small number of recordings
		Aortic stenosis (AS)	200					
		Mitral valve prolapse (MVP)	200					
		Mitral stenosis (MS)	200					
		Mitral regurgitation (MR)	200					
[34]	HSS	Normal	-	845	30s on average	4KHz	Electronic Stethoscope	Lacks other heart diseases
		Mild	-					
		Moderate/Severe	-					

2 Related work

CVD detection using the advancement of technologies such as deep learning algorithms has been a recent focus for many researchers. Even though those researchers have considered the same problem of CVD detection, they have dealt with using heart sound PCG signals for classifying normality or abnormalities the heart. Generally, there are four phases in classifying heart sounds PCG signals: de-noising, segmentation, feature extraction, and classification [16].

2.1 The de-noising phase

De-noising is the method used to eliminate unwanted components from heart sound signals. During the recording of heart sounds, environmental interference affects the quality of the recordings. Heart sound signals are affected by factors such as contact pressure between both the stethoscope and the skin and noise such as lung sounds, breathing sounds, and background sounds; thus, it is necessary to remove this noise. The effectiveness of the segmentation, feature extraction, and classification phases are significantly impacted by denoising. The most popular de-noising methods include wavelet de-noising, empirical mode de-noising, and filter de-noising. The filters can be used for band-pass, low-pass, and high-pass applications [17].

2.2 The features extraction phase

Segmentation aims to split PCG signals into four portions or sections: first heart sounds (S1), systole, second heart sounds (S2), and diastole. Each of these segments has features that make it possible to effectively distinguish the different categories [18].

2.3 The features extraction phase

Features play a crucial role in classifying a signal. The features consist of the important information in the signal. In the case of heart sounds classification, these feature extraction methods are a very important stage of PCG analysis for detecting any CVD. As a result, the feature extraction technique is useful in converting data into information, making it easier to achieve higher accuracy in the classification stage [19]. The features can be extracted from various domains such as the time domain, frequency domain, and time-frequency domain. In the time domain, the features of the time domain are the characteristics of the signal over time. The windowing technique is used to divide the signal into frames. The features are taken from each frame in the audio signal. In the frequency domain, frequency domain analysis is the most vital tool in audio signal processing. To analyze audio signals in the frequency domain, the time area signal is converted into the frequency domain using Fourier transform. The time-frequency domain provides frequency

information and the time of the audio signal. It bridges the gap between the time domain and frequency domain [8]. There is many time- frequency domain methods such as spectrogram [20] and MFCCs [21]. MFCCs can be used in the form of images as input for a CNN classifier.

2.4 The classification phase

The classification phase is the method used to split PCG signals into normal sounds or additional abnormal sounds. Classification methods can be applied to the takeout features to recognize the exact sound type. Machine learning algorithms and deep learning methods have been applied to PCG signals such as SVM, CNN, and image pre-trained models [22], [20],[23]. Mehrez Boulares et al. [22] applied image pre-trained models to the Pascal dataset. MFCCs features were extracted and saved as images. The main findings of this paper are a benchmark that can be used for classification results comparison of CVD recognition. Omair Rashed Abdulwareth Almanifi et al. [24] applied four image pre-trained models: inceptionV3, MobileNet, VGG16, and VGG19. The Pascal dataset was used for training and testing the models. The primary contribution of this paper is its demonstration that transfer learning models can enhance CVD recognition.

Tomoya Koike et al. [25] proposed a new audio pre-trained audio neural networks (PANNs) model. The PhysioNet dataset was used to train the model. Log mel spectrograms are used as features. The main findings of this paper are a novel audio pre-trained model. Moreover, audio-pre-trained models outperformed popular image pre-trained models.

Mukherjee et al. [26] applied images of six pre-trained CNN models: DenseNet169, ResNet152V2, MobileNetV2, MobileNet, InceptionResNetV2, and Xception. The Pascal A B Datasets were used. The spectrogram features were saved as images. The main finding is that image-pre-trained models are promising for the automation of heart sound classification.

Shahid Ismail et al. [20] suggested a mechanism that includes filtering the PCG signal, time segmentation, extracting spectrogram features, image pre-trained AlexNet with SVM classification, and a voting-based system. They used the Pascal A B dataset. The main finding of this paper is that 2–3-s length of data is enough for classification.

Menghui Xiang et al. [27] applied deep learning and transfer learning using four image pre-trained models: Xception ResNet50, InceptionV3, and MobileNet. In this study, four types of features were used: mel spectrogram, log power spectrogram, waveform, and envelop. The primary conclusions of this study are that using time-frequency features is preferable to using just time-domain features. Additionally, transfer learning can increase the model's classification precision.

Zhihua Wang et al. [28] utilized a pretrained deep neural network. The authors of the paper used eight different time-frequency feature extraction methods. The PCG features were converted to images to be input for the classifier. The main findings of this paper are that a fine-tuned deep model can improve the mean accuracy. In addition to this finding, Stockwell transformation provides noise robustness of PCG signals.

Miao Wang et al. [29] considered ten image pre-trained models. The PCG signals were converted to a 3-dimensional spectrogram by continuous wavelet transform (CWT). The dataset in GitHub [15] used an extra class of heart sound, pulmonary hypertension, comprised of 200 samples. The total number of samples is 1200, with 200 samples for each class. The main findings of this paper are that the proposed method achieved high accuracy despite the noisy background and that this proposed model can be used for CVD detection.

Guangyang Tian et al. [30] suggested a new deep learning model called DsaNet. The PhysioNet dataset was applied for training the model. The model's input was the 1D time domain. On the open-source 2016 PhysioNet dataset, the reported model outperformed seven different baseline models. This article's key conclusion is that DsaNet may compete favorably in imbalanced PCG signal classification with fewer parameters and calculations.

Neeraj Baghel et al. [31] employed a deep learning model, convolutional neural network (CNN). The PCG signals were converted to log-mel spectrogram images to use as input for the model. The GitHub dataset [15] contains 1000 samples, with 200 for each class of the 5 classes used. The main finding of this paper is that the proposed model can be deployed to any computer type as it achieved a high accuracy of 98.60%.

Jay Karhade et al. [32] proposed a time frequency domain deep learning framework. The authors of the paper examined both time and frequency-domain chirplet transform-based images as input for deep CNN to detect four types of CVDs. The key conclusion of this paper is that the recommended model can be validated in real-time for CVD detection.

Bin Xiao et al. [33] presented a 1D CNN with remarkably minimal parameter usage. The suggested model has three parts: prepossessing, deep CNN classification with an attention mechanism, and majority voting for final prediction. The PhysioNet dataset was used to evaluate the model. The proposed model shows superior results compared to cutting-edge approaches. It produces superior classification results and consumes fewer resources. Table 2 shows an overview

of related work. We conclude that audio-pre-trained CNN models have not been employed by researchers yet. In [22], the dataset used is the Pascal dataset, which is confined to a small number of samples and murmur classes. Also, the study does not include all Keras image pre-trained CNN models and audio pre-trained CNN models. In this study, we bridge the gap by using huge dataset samples, 26 image pretrained CNN models and popular audio pretrained CNN models.

Table 2: An overview of the related work papers.

Reference	Year	Dataset	Feature extraction methods	Classifier	Classification type	Accuracy (%)
Shahidi smail et al. [20]	2023	- PASCAL A & B [14]	- Spectrogram	- AlexNet -SVM	- Binary -multi-class	-> 97
Menghui Xiang et al. [27]	2023	- Physionet2016 [13] - Dataset on GitHub [15]	- Mel spectrogram - Log power spectrogram -Waveform -Envlop	- Xception -MobileNet	-- Binary	94%.
Zhihua Wang et al. [28]	2023	- Physionet2016 [13]	- Time–frequency (images)	VGG16 -VGG19	- Binary	65%
Miao Wang et al. [29]	2022	- Dataset on GitHub [15] - private	- Spectrograms (Image)	- ResNet101 -DenseNet201 - DarkNet19 -GoogleNet	-- Mult-calss	98%
Guangyang Tian et al. [30]	2022	PhysioNet/CinC [13]	- 1D time domain	- DsaNet	- Binary	-
Jay Karhade et al. [32]	2022	- Dataset on GitHub [15] - Physionet2016 [13]	- Time–frequency images	- Deep CNN	-Binary - multi-class	-99.48 multiclass - 85.16 binary
Omair Almanifi et al. [24]	2021	- PASCAL [14]	- Spectrograms	- VGG16 - VGG19 - MobileNet -InceptionV3		80.25
Mukherjee et al. [26]	2021	- Pascal A & B [14]	- Spectrograms (Image)	- DenseNet169 -ResNet152V2 -MobileNetV2 -MobileNet - InceptionResNetV2 -Xception	- Multi-class	AUROC 0.97
Mehrez Boulares et al. [22]	2020	Pascal A & B [14]	- Image MFCCs	- Image pre-trained models	- Binary	0.89
Tomoya Koike et al. [25]	2020	- PhysioNet2016 [13]	- Log Mel spectrogram - Spectrograms	- Audio pre-trained model	- Binary	UAR 89.7%.
Neeraj Baghel et al. [31]	2020	- Dataset on GitHub [15] - Private	- Log mel spectrograms	- CNN	- Multi-class	98.60%
Bin Xiao et al. [33]	2020	- Physionet2016 [13]	- I D time domain	- Deep CNN	- Binary	93

3 Benchmark process

We transform heart sound signals to mel spectrograms. For image pre-trained CNN models, we added three layers to the base model to increase the accuracy classification. As shown in Figure 2, the first added layer was Global Average Pooling 2D, and the second and third are dense layers 1024 and 512 respectively. The last layer is the dense classification layer with a sigmoid activation function. A stochastic gradient descent (SGD) optimizer is employed. The learning rate is 0.0001. The epochs are set to 30, and the batch size is set to 5. For the audio pre-trained CNN models heart sound signals were converted to log mel spectrograms. As shown in Figure 3, we add five dense layers 2287 with the relu activation function. To prevent systems from overfitting, a dropout with a rate of 0.5 is implemented after the three dense layers. Before the last layer, batch normalization was introduced to reduce overfitting. Dense layer of

classification by the softmax activation function. The optimizer is Adam, the batch size is 32, the epoch is 20, and the learning rate is 0.0001.

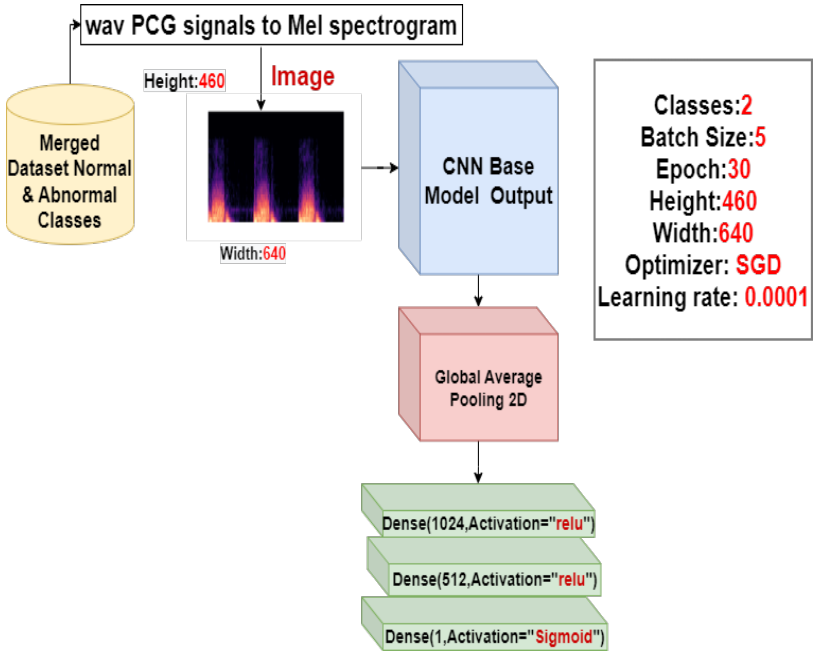

Fig. 2: The proposed approach architecture using Keras image CNN pre-trained models.

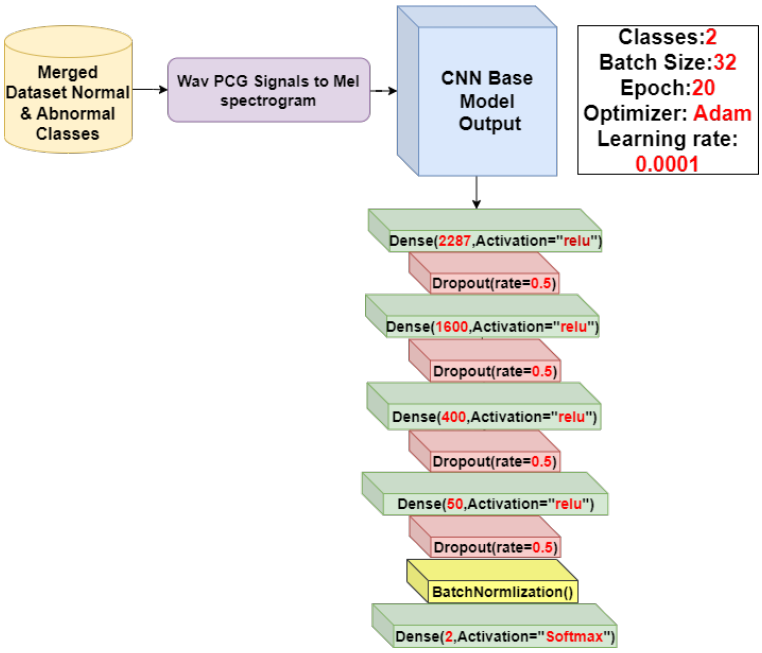

Fig. 3: The proposed approach architecture using YAMnet and VGGish audio CNN pre-trained models.

3.1 Dataset

In this section, we will demonstrate the open-access heart sounds public datasets and the merged dataset that has been used for this benchmark. Table 1 presents the technical aspects of the accessible public heart sounds datasets. The PhysioNet/CinC Challenge 2016 dataset is an openly accessible heart sounds public dataset [13]. The total number of recordings is 3240 files in .wav format. The PhysioNet dataset is gathered by seven research teams. The dataset contains only two classes: normal heart sounds and pathological abnormal sounds. The total number of normal heart sound recordings is 2575 and the total number of aberrant heart sound recordings that show pathological cases is 655. The recorded files are sampled at 2000 Hz. The duration of the recordings varies from 5 seconds to 120 seconds. This dataset lacks disease-based labeling and is comprised of only two groups (normal and abnormal). The accessible

available heart sounds Pascal dataset is set A and B [14]. The Pascal dataset A has four kinds of heart sounds: normal is 45 files, murmur is 48 files, extra heart sound is 27 files, and artifact is 56 files. The total is 149 recording files in .wav format. They were gathered publicly by the istethoscope Pro iPhone app. The sample rate is 44.1 kHz. The recordings' duration varies from 1 to 30 seconds. The Pascal dataset B is larger than the Pascal dataset A. The total of all recordings is 279 files in .wav format. There are three types of heart sounds: normal is 167 files, murmur is 69 files and extrasystole is 39 files. They were collected from a hospital clinic using a digital stethoscope called a DigiScop. The recording files' duration ranges from 1 to 30 seconds and the sampling rate is 4 kHz. The Heart Sounds Shenzhen (HSS) PCG signal corpus dataset contains 845 recording files [34]. They were gathered from 170 different persons. Files were recorded from individuals suffering from a wide range of heart conditions, including coronary heart disease, arrhythmia, valvular heart disease, congenital heart disease, and others. The PCG recordings are recorded at 4 kHz and labeled with three class labels: Normal, Mild, and Moderate/Severe (heart disease). This dataset has no heart sounds or murmurs. The class Moderate/Severe does not indicate the severity of the valve or disease. The audio was captured in .wav format using an electronic stethoscope (Eko CORE, USA). The duration of the length of the recordings is 30 seconds on average. They are relatively limited in terms of the number of samples. The fourth public dataset is proposed in [15]. There is a total of one thousand .wav audio files included, with five different heart sound types (normal, aortic stenosis, mitral valve prolapses, mitral valve stenosis, and mitral regurgitation). Each class has 200 recordings with a duration of 3 s. The dataset was collected from many resources such as websites and books. All the recording files are sampled at 8000 Hz. The limitation of this public dataset is that it is missing classes that are common such as aortic regurgitation, tricuspid regurgitation, tricuspid stenosis, and abnormal classes that indicate S3 and S4, which indicate many heart diseases. In this work, we merged the PhysioNet, PASCAL A–B and (GitHub) datasets to have more samples for normal and abnormal classes so that any pathological sounds are in the abnormal class and healthy (normal) sounds are in the normal class. After the merging, we obtained 3127 samples of the normal class and 1659 of the abnormal class. The total of both classes is 4785. Table 3 shows the merged dataset with the sample numbers of each class, and Figure 4 shows the merged dataset compared to the public heart sound datasets.

Table 3: Merged dataset.

Class	Samples Number
Normal	3126
Abnormal	1659
Total	4785

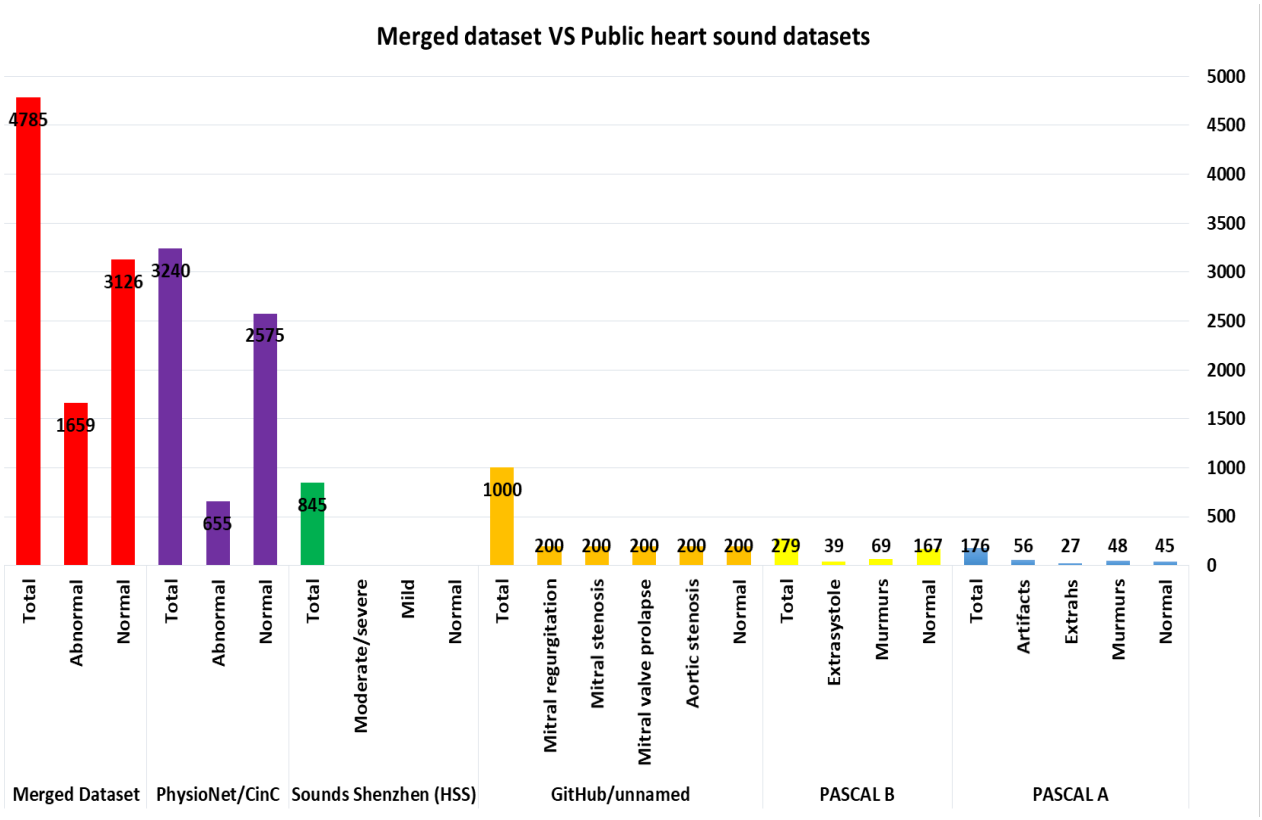

Fig. 4: The merged dataset vs public heart datasets.

3.2 Preprocessing

Mel spectrograms have been widely used for heart sound recognition tasks [35]. We use mel spectrograms for image pretrained CNN models and log mel spectrograms for audio pre-trained CNN models as they performed better in [37]. For the image pre-trained CNN model, we converted the heart sound signals to mel spectrograms and saved them as png format images. For audio pre-trained CNN models, the models take the audio clips as input and convert them to log mel spectrograms. For image pre-trained CNN models, mel spectrograms were computed as follows:

–The heart sound signal sampling rate 44,100

–Divide PCG signals into frames using the Hamming windowing function at interval 1024 and hop length 256. Obtain from this step the cepstral feature vector per each frame.

–Apply discrete Fourier transform for each frame.

–Apply 40 mel filter banks to the spectrograms. Figure 5 shows an example of converted PCG signals to mel spectrogram image for heart sound normal and abnormal.

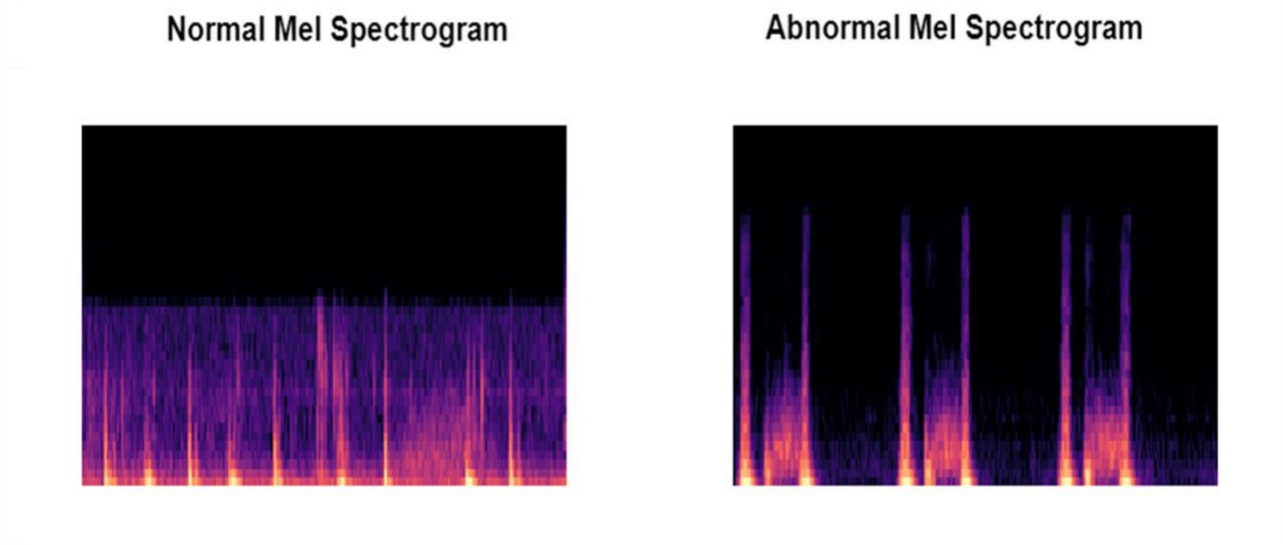

Fig. 5: An example of converted PCG signals to mel spectrogram images for normal and abnormal heart sounds.

For audio pre-trained CNN models, the log mel spectrograms are computed as follows:

–Audio clips are resampled to 16 KHZ mono.

–Short-time Fourier transform is employed to compute the spectrogram with a 25-ms window size, 10-ms window hop and a periodic Hann window. the spectrogram to 64 mel bins surrounding the range 125–7500 Hz.

–Log transform the magnitude of each bin and add a small offset to evade abiding of the logarithm of zero.

–Frame features include non-overlapping frames of 0.96 seconds, where every frame lids 64 mel bands and 96 frames of 10-ms apiece. Therefore, the form of log-mel spectrogram is 96 * 64 bins that input to the classifier.

We have used the PCG signal as raw without any of preprocessing methods such as de-noising or segmentation.

3.3 Image pre-trained CNN models

Deep CNNs models take days or weeks to train process on very big datasets. To overcome this issue, this procedure reuses the model weights from pretrained models created for standard computer vision datasets, for example, the ImagNet dataset for image recognition tasks. The best models that have elevated performance should be used (transferred) straight or joint over on a new model. The best models can be used as the basis for transfer learning in computer vision applications. These models have been taught on millions of images for 1000 classes. The models have state-of-the-art performance. They have been used for real image recognition tasks. The model weights can be downloaded for free and easily with the required Keras libraries. Many of these models are free and can be transferred. In our work, we use all the available models for a total of 26 pretrained models. Table 4 demonstrates Keras-image-trained CNN models and the popular audio-pretrained models with the layer total, model size, and model parameters. In our study, we used all the available models for a total of 26 image pre-trained CNN models.

Table 4: Keras image and audio pre-trained CNN Models.

Model's name	Layers	Size	Parameters	Ref
Xception	81	88 MB	22.9 million	[38]
VGG16	16	528 MB	138.4 million	[39]
VGG19	19	549 MB	143.7 million	[39]
ResNet50	107	98 MB	25.6 million	[40]
ResNet50V2	103	98 MB	25.6 million	[41]
ResNet101	209	171 MB	44.7 million	[40]
ResNet101V2	205	171 MB	44.7 million	[40]
ResNet152	311	232 MB	60.4 million	[41]
ResNet152V2	307	232 MB	60.4 million	[41]
InceptionV3	189	92 MB	23.9 million	[42]
InceptionResNetV2	449	215 MB	55.9 millions	[43]
MobileNet	55	16 MB	4.3 million	[44]
MobileNetV2	105	14 MB	3.5 million	[45]
DenseNet121	242	33 MB	8.1 million	[46]
DenseNet169	338	57 MB	14.3 million	[46]
DenseNet201	402	80 MB	20.2 million	[46]
NASNetMobile	389	23 MB	5.3 million	[47]
NASNetLarge	533	343 MB	88.9 million	[47]
EfficientNetB0	132	29 MB	5.3 million	[48]
EfficientNetB1	186	31 MB	7.9 million	[48]
EfficientNetB2	186	36 MB	9.2 million	[48]
EfficientNetB3	210	48 MB	12.3 million	[48]
EfficientNetB4	258	75 MB	19.5 million	[48]
EfficientNetB5	312	118 MB	30.6 million	[48]
EfficientNetB6	360	166 MB	43.3 million	[48]
EfficientNetB7	438	256 MB	66.7 million	[48]
VGGish	15	256.37 MB	72.1 million	[49]
YAMnet	86	13.58 MB	3.7 million	[49]

3.4 Audio pretrained CNN models

The image pre-trained CNN models have been used for audio classification tasks. Audio pre-trained CNN models were not developed due to the scarcity of data sets for audio classification. Nowadays, several large-scale audio datasets have been developed such as the YouTube-8M dataset and the Audioset dataset. Audioset consists of 1.9 million audio clips and 521 audio event classes [36]. Many CNN models have been trained on the Audioset dataset and can be ported to audio classification tasks. We will demonstrate the audio pre-trained audio CNN models.

–Pretrained audio neural networks (PANNs) model

PANNs is an audio pre-trained CNN model [50]. PANNs is trained on the Audioset dataset [51]. Audioset consists of 1.9 million audio clips and 521 audio event classes. PANNs inspired the VGG CNN model. PANNs consists of a 14-layer CNN. The 14-layer CNN has been transferred to several audio pattern tasks and performed well. PANNs consists of 14 layers: 5 blocks of 3*3 convolutional. Each convolutional block is comprised of two convolution layers with a kernel size of 3*3 followed by batch normalisation and a ReLU function. For each convolutional block, the 2*2 average pooling size has been implemented. Global pooling has been implemented after last convolutional block followed by two fully connected layers: one with ReLU and the second layer with softmax nonlinearity [50]. The input of the PANNs model is log mel spectrogram. Log mel spectrograms compute by employing short-time Fourier transform on the audio signal waveforms using a Hann window size of 1024 and 320 samples of hop size. One hundred frames per second are obtained, then 64 mel filter banks are applied. The mel banks are set to 50 Hz and 14 Khz for the upper and lower frequencies, respectively, to eliminate low frequency noise. The sample rate is 32,000 Hz [50]. In [25], the PANN was employed for classifying normal and pathological heart sounds. This is only the paper using the audio CNN pretrained model.

–Kumar embeddings

Kumar embeddings was suggested in AudioSet [52]. It is proposed for solving weak labels. It combines embeddings through the time for each file. It is trained as a supervised method. It uses mel spectrograms as input. It has good embeddings. These embeddings are used and fine tuned with support vector machine and showed a good result. Global

max pooling has been used.

–LOOK, LISTEN, AND LEARN (L3)

The L3 pretrained audio CNN model uses a supervised technique. It uses audio visual in the videos and produces robust embeddings. The selections made during the L3-Net design influence the output of audio classifiers trained with all these extracted features. It shows that audio-informed input representation choices are important and that it is necessary to use sufficient data for embedding training [53].

–OpenL3 embeddings

As an extension, OpenL3 embedding was suggested in in [53]. This embedding is proposed for Net-L3. It has been trained as an audio data set. Various models were trained in the identical of L3-Net (self-supervised way) to examine the AudioSet. Various design choices affect classification accuracy. The accuracy depends on the embedding size as well as the embedding input representation.

TRILL (TRIpLet Loss network) Model

TRILL representation is learned in a self-supervised method on speech containing clips from AudioSet. The model network defines audio such that segments that are nearer in time are also nearer in the embedding space. TRILL demonstrates that this simple proxy objective is very effective in learning a robust representation for several non-semantic speech tasks [54].

–FRILL: Non-Semantic Speech Embedding for Mobile Accessories

FRILL (fast TRILL) is a non-semantic speech embedding model. It is meant for mobile device speech embedding. It is sufficient for operating in real time on mobile devices and exhibits low performance cost on a benchmark of non-semantic speech tasks [55].

–VGGISH

VGGish is an audio pretrained CNN model. It has been developed in [56] using the VGG network architecture, which is a CNN model trained on over 2*106 YouTube videos and can forecast over 600 audio occurrence types. Figure 6 shows the VGGish Convolutional Neural Networks Architecture. In the VGGish model, three fully connected layers and four convolutional (CONV) layers come after a single channel input layer. The well-known VGG model is reimagined as VGGish, with configuration A having 11 weight layers. The audio signal was split into 0.96-second intervals with no overlap in order to prepare it for the VGGish model. Using a short-time Fourier transform, the window size is 25 ms. The hop length is 10 ms. For the Hanning window with periodic sampling, each window was converted into a spectrogram. The spectrogram was then translated to 64 mel bins with a frequency range of 125–7500 Hz to create a mel spectrogram. To avoid computing a logarithm of zero, a log offset value of 0.001 was added to construct a log mel spectrogram. Each window created a 96 x 64-pixel two-dimensional (2D) spectrogram picture (96 frames by 64 mel bands). VGGish is suitable for many kinds of applications. Many studies have used it and proved this. In [57], VGGish was employed for speech emotion recognition. In [58], the VGGish model was used for environmental sound classification. In [59]], VGGish was utilized for domestic activity recognition. In [60], VGGish was used for sound event detection. In [61], the VGGish model was used for speech and music classification. In [62], the VGGish model was applied to urban sound classification. In our review of the literature, the VGGish model was not employed to classify heart sounds. This motivates the adoption of the VGGish model for classifying heart sounds and determining its accuracy.

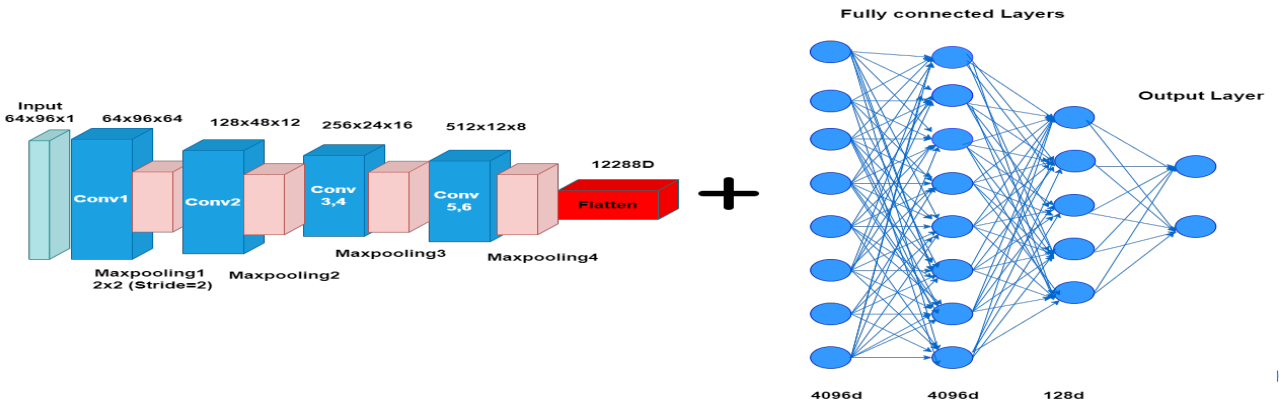

Fig. 6: VGGish Convolutional Neural Networks Architecture.

YAMnet Model

YAMnet is an audio-pre-trained CNN model proposed in [56] . YAMnet employs the MobileNet network architecture [44]. YAMNet classifies the audio files to sound categories detailed in the AudioSet dataset [36]. YAMnet was trained on the AudioSet dataset. The MobileNet network utilizes depth-separable convolutions. The depthwise separable convolutions consist of a conventional convolution (1 x 1 convolution filter) into a depthwise and a pointwise convolution. The filter is applied to each intake channel utilizing depthwise convolution, and the outcomes of depthwise convolution are merged using 1 x 1 pointwise convolution. Figure 7 shows the YAMNET neural network architecture using the MobileNet model. It is a conventional convolution in the first layer. The last layers are pooling, fully linked layers, and a softmax layer for classification. All layers are depth-wise separable convolutions. Each convolution layer's activation function was ReLU, and batch norm was employed to standardize the batch distribution. The design of the typical convolution layer with batchnorm and ReLU is shown in Figure 8. Figure 9 illustrates the depthwise separable convolutions using Depthwise and Pointwise layers followed by batchnorm and ReLU. One-side short-time Fourier transform (STFT) was applied. The periodic Hann window is 25 ms. The hop length is 10 ms. The discrete Fourier transform has 512 points. The segments were converted into a size (magnitude) spectrogram with 257 frequency bins (DFT). The magnitudes of each band were added after the spectrum was run through a 64-band mel-spaced filter bank. A 96 by 64 by 1 by L array, where 96 is the number of spectrums in the mel spectrogram and 64 is the number of mel bands, was used to describe the audio. The mel spectrograms were eventually scaled using the log scale. The 96 by 64 by 1 by L array of mel spectrograms was fed into YAMNet as the input layer. The outcome of the YAMnet model conveys confidence scores for each of the 521 sound categories for a given piece of audio over time. The YAMnet model has been extensively applied to audio classification. In [63], YAMnet was used for the automatic recognition of COVID-19 cough. In [64], YAMnet was used for speech emotion recognition. In [65], the YAMNet model was used to recognize COVID-19 cough. In [66], the YAMnet model was applied for identifying Parkinsonian speech. In [67], YAMnet was applied to respiratory sound classification applications. In our literature review, YAMnet was not used for heart sound classification tasks. This has motivated us to use the YAMnet model for heart sound classification tasks and discover the accuracy. In our work, we will use the popular audio pre- trained CNN models VGGish and Yamnet. Table 4 shows the popular audio CNN models' layers and the model's size. In our work, we use these pre-trained models.

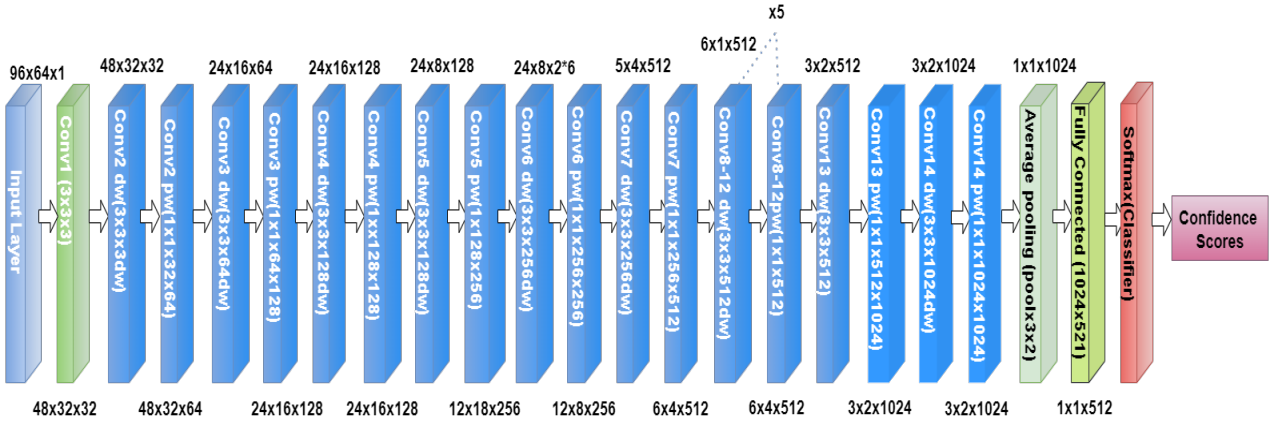

Fig. 7: YAMnet Convolutional Neural Networks Architecture.

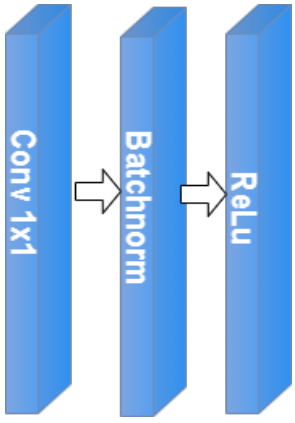

Fig. 8: The design of the typical convolution layer with batchnorm and ReLU.

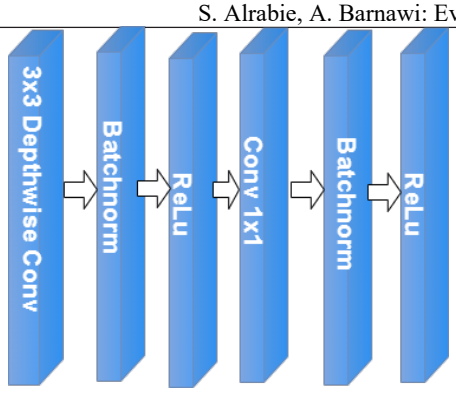

Fig. 9: The depthwise separable convolutions using depthwise and pointwise layers followed by batchnorm and ReLU.

In our work, we will use the popular audio pre-trained CNN models VGGish and Yamnet. Table 4 shows the image pre-trained models and the popular audio CNN models' layers and the model's size. In our work, we use these pre-trained models.

4 Experiments and results

In this part, we demonstrate our experiments using image pretrained models and the audio pretrained models VGGish and YAMnet with the merged dataset. The merged dataset is unbalanced and has two classes: 3126 normal samples and 1658 abnormal samples as shown in Table 3. We fine-tuned twenty-six Keras image pretrained CNN models and two popular audio pre-trained models listed in Table 4. We employ 2 and 3 cross-validation to the merged dataset. The configuration is using image pre-trained models, and the number of epochs is fixed 30. The batch size is set to 5. The learning rate is 0.0001. For audio pretrained CNN models, the learning rate is 0.0001. The epochs number is set to 20. The optimizer is Adam. The batch size is 32. We use Google Colab Pro for training. Dedicated GPU usage is provided by the Colab Pro platform, with 2496 CUDA processors, compute 3.7, and 12 GB, GDDR5 VRAM, and the GPU is a 1xTesla K80.

4.0.1 Using two folds

50% of the merged dataset is used for training, and 50% is used for testing where there are 1563 normal class samples and 829 abnormal class samples. We will discuss the obtained results for each fold; then, we will explain the results of 3-fold cross-validation as the average for two-folds.

Table 5: Validation TPR of the Keras image and audio pre-trained CNN models utilizing two folds.

Model	**2 Fold**						
	Fold one			**Fold two**			**AVG**
	Normal	**Abnormal**	**Avg**	**Normal**	**Abnormal**	**Avg**	
VGG16	0.91	0.76	0.84	0.72	0.92	0.82	0.83
VGG19	0.97	0.62	0.80	0.02	1.0	0.51	0.65
MobileNet	0.90	0.79	0.84	0.99	0.10	0.55	0.70
InceptionV3	1.0	0.0	0.5	0.99	0.08	0.54	0.52
InceptionResNetV2	0.94	0.24	0.59	1.0	0.0	0.5	0.54
Xception	1.0	0.06	0.53	0.99	0.39	0.69	0.61
DenseNet121	0.95	0.45	0.70	0.98	0.36	0.67	0.68
DenseNet169	0.99	0.00	0.49	0.47	0.95	0.71	0.60
DenseNet201	1.0	0.0	0.5	0.99	0.020	0.56	0.51
NasNetMobile	0.99	0.16	0.57	1.0	0.0	0.5	0.53
MobileNetV2	0.99	0.16	0.57	1.0	0.0	0.5	0.53
ResNet50	0.93	0.73	0.83	0.85	0.83	0.84	**0.84**
ResNet101	0.87	0.80	0.84	0.99	0.54	0.76	0.80
ResNet152	1.0	.002	0.50	0.89	0.80	0.84	0.67
ResNet50V2	1.0	0.0	0.5	1.0	0.0	0.5	0.50
ResNet101V2	1.0	0.0	0.5	1.0	0.0	0.5	0.5
ResNet152V2	1.0	0.0	0.5	1.0	0.0	0.5	0.5
NASNetLarge	1.0	0.0	0.5	1.0	0.0	0.5	0.5

EfficientNetB0	1.0	0.06	0.53	0.88	0.69	0.78	0.66
EfficientNetB1	0.98	0.58	0.78	0.98	0.63	0.81	0.80
EfficientNetB2	0.96	0.68	0.82	0.98	0.53	0.76	0.79
EfficientNetB3	0.99	0.46	0.73	1.0	0.10	0.55	0.64
EfficientNetB4	0.97	0.54	0.76	0.99	0.20	0.60	0.68
EfficientNetB5	0.99	0.32	0.66	0.99	0.58	0.79	0.72
EfficientNetB6	0.96	0.63	0.80	0.99	0.47	0.73	0.76
EfficientNetB7	0.98	0.51	0.75	0.99	0.57	0.78	0.76
VGGish	0.93	0.77	0.85	0.89	0.80	0.83	**0.84**
YAMnet	0.95	0.59	0.77	0.94	0.61	0.77	0.77

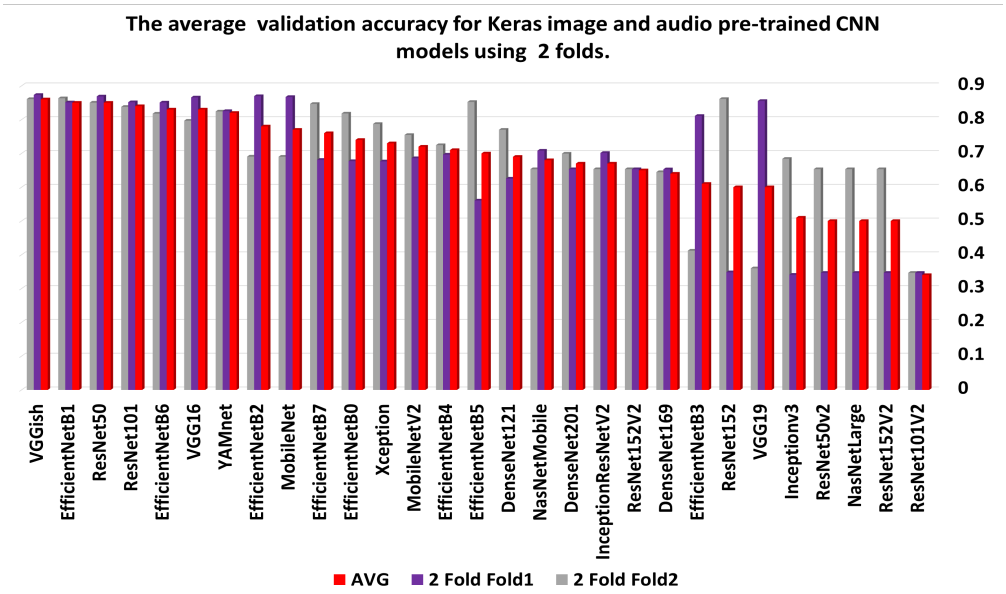

Fig. 10: The average validation accuracy for image and audio pre-trained CNN models utilizing two-folds.

–Fold1:

The VGGish audio pre-trained models reach the highest average validation accuracy of around 87% with the average true positive rate (TPR) of both classes of 85%. The normal class reaches TPR=93%, while the abnormal class reaches TPR=77%. The image pre-trained CNN models, MobileNet, ResNet50 and EfficientNetB2 pre-trained CNN models reach an average validation accuracy of around 86% as displayed in Figure 10. The MobileNet ResNet101 and VGG16 have the top average TPR at around 84% as shown in Table 5. This represents the depth of the model architecture and has no influence on the validation accuracy or the average of the TPR. Table 4 shows the layers of the models. The depth of the model layer does not play a large role in the accuracy. For example, VGG16 has 16 layers, MobileNet has 55 layers, and both of them have the same validation accuracy. ResNet50 has 107 layers and achieves 86% validation accuracy, the same as VGG16, which has 16 layers. ResNet101 has 209 and achieves 85% validation accuracy, the same validation accuracy of VGG19, which has 19 layers. EfficientNetB2 has 186 layers and achieves the VGG16's validation accuracy. An imbalanced dataset has a large impact on the abnormal class TPR for most models such as ResNet50V2, ResNet152V2 and NASNetLarg. ResNet50V2, ResNet152V2 and NASNetLarg reach 100% TPR for the normal class and a 0% TPR for the abnormal class, while ResNet101 has less impact. ResNet101 reaches a TPR of 87% for the normal class and 80% for the abnormal class as displayed in Table 5. Even though YAMnet is an audio pre-trained CNN model and trained on AudioSet, YAMnet achieves 82% validation accuracy and an average TPR that is less than some image pretrained CNN network models such as VGG16 or VGG19 and EfficientNetB6. YAMnet achieves the average TPR for both classes of 77%; this is 95% for the normal class and 59% for the abnormal class. The results of fold one proves our hypothesis that audio models can achieve better validation accuracy for audio tasks than image models as VGGish results. The imbalanced training samples impact the TPR. The depth of neural networks has no effects on increasing or decreasing the validation accuracy.

–Fold2:

VGGish, ResNet152 and EfficientNetB1 reach the highest validation accuracy of 86%. ResNet50 and ResNet152 reach the highest average TPR at 84%. The number of layers for ResNet152 and EfficientNetB1 are 311 and 186, respectively, as shown in Table 4. VGGish was not affected more by imbalanced training samples. VGGish reached an

average TPR of 83%, 89% for normal class and 80% for abnormal class. Moreover, ResNet50 was not affected by an imbalance in training samples. ResNet50 achieves an average TPR of 84%, 85% for normal class and 83% for abnormal class. For many models in this fold, the TPR was heavily influenced by imbalanced training samples such as InceptionResNetV2, ResNet50V2, and ResNet101V2, which had 100% for the normal category and zero for the abnormal category. We deduce again that the depth of the model architecture has little impact on the validation precision or average of the TPR.

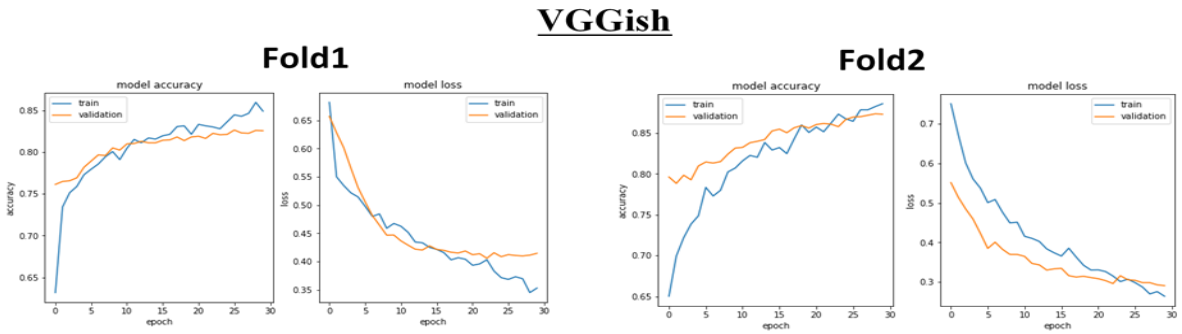

Fig. 11: The loss curves and training validations of the pre-trained CNN models using VGGish using two folds.

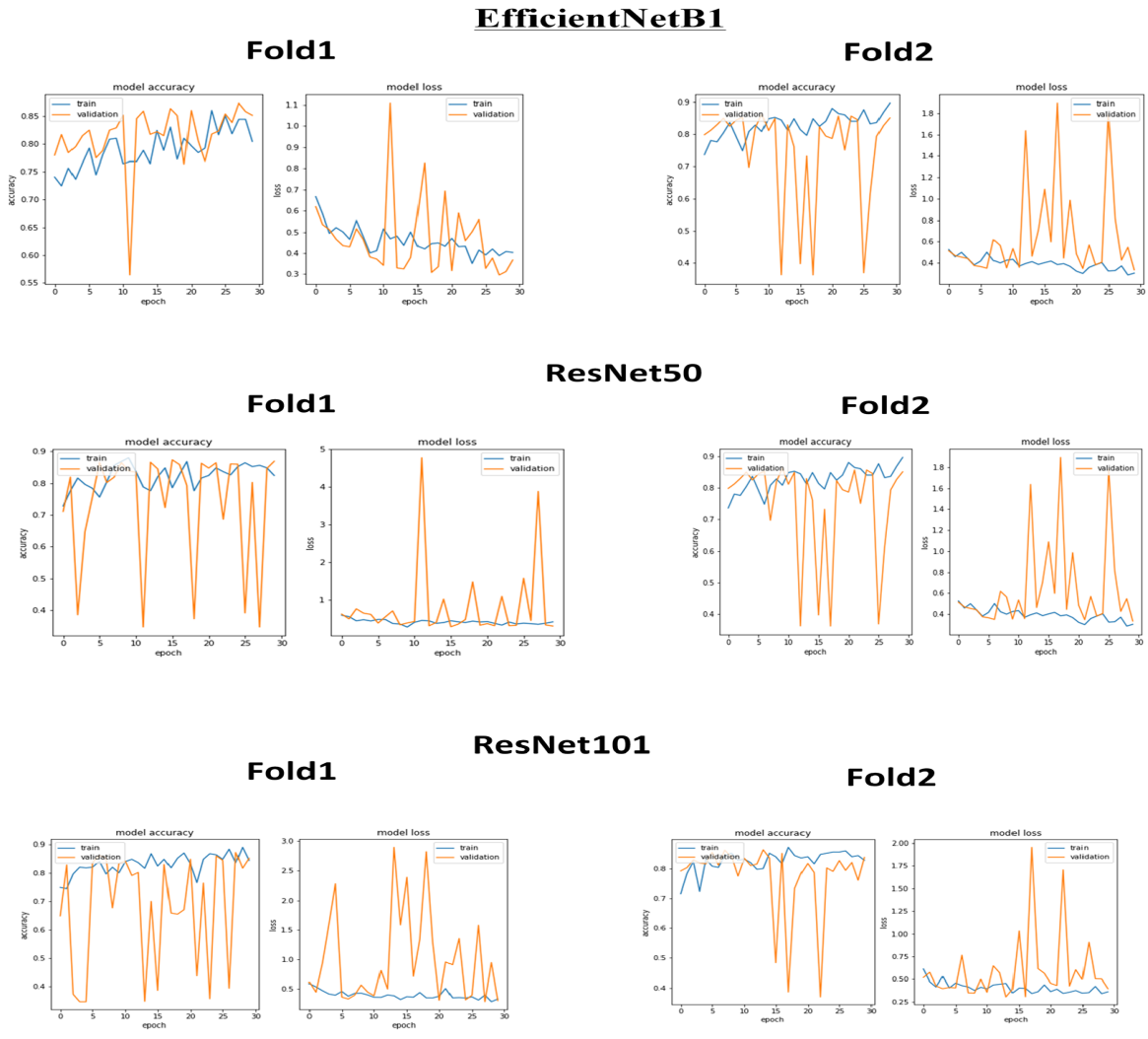

Fig. 12: The loss curves and training validations of EfficientNetB1, ResNet50 and ResNet101 image pre-trained CNN models using 2 folds

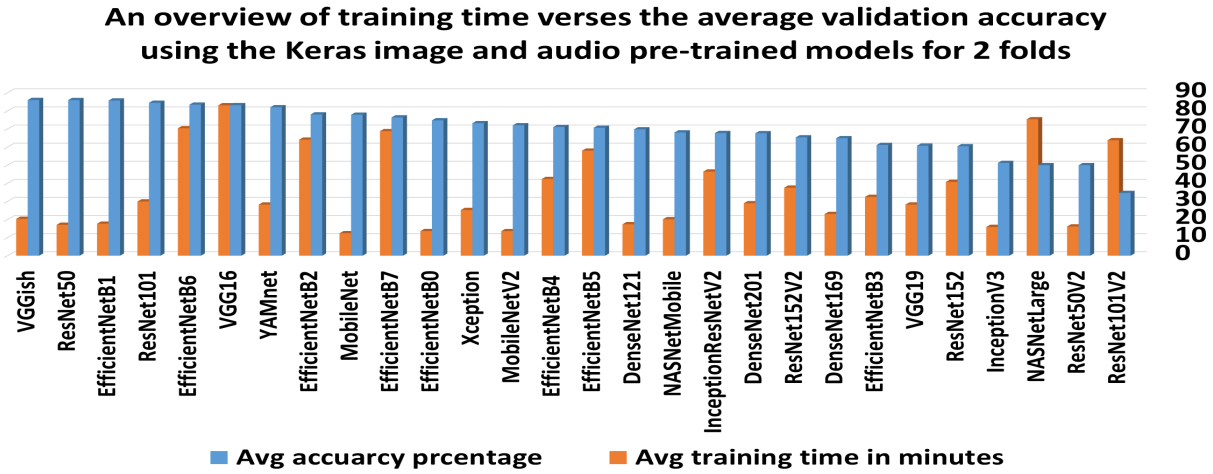

Fig. 13: Summary of the average of training time stanzas and average of validation precision using image pretrained CNN models and audio pretrained CNN models for 2 folds.

We deduce from using 2-fold cross-validation the average result of validation accuracy and average TPR as follows. The VGGish model has the highest average validation accuracy of 86%, which outperformed all image pre-trained models as seen in Figure 10. The ResNet50 and EfficientNetB2 reach an average validation accuracy of 85% and ResNet101 reaches 84%. Figure 11 demonstrates the loss curves and training validations of the pre-trained CNN models using VGGish using two folds. Figure 12 displays the loss curves and training validations of EfficientNetB1, ResNet50 and ResNet101 image pre-trained CNN models using 2 folds. VGGish and ResNet50 reach the highest average TPR of 84% as shown in Table 5. Figure 13 demonstrates a summary of the average of training time stanzas and average of validation precision using image pretrained CNN models and audio pretrained CNN models for 2 folds. We see from Figure 12 that VGGish, ResNet50 and EfficientNetB2 achieve the highest validation accuracy and lowest training time. We conclude from using 2-fold cross-validation that the VGGish pre-trained CNN model performed the best average validation accuracy and average TPR for both classes (normal-abnormal) amongst the twenty-six Keras image pre-trained CNN models and audio pre-trained CNN models. ResNet50 performed the best average validation accuracy and average TPR for image pre-trained models. In addition, the depth of the model architecture has little influence on the validation precision. Moreover, even the imbalanced dataset in training samples models can have a high TPR for both classes such as the Resent50 pre-trained CNN model in two-fold normal class TPR=85% and abnormal class TPR=83% as shown in Table 5. VGGish and Resent50 are suitable pre-trained models for CVD classification of their performance validation accuracy, TPR and training time.

4.0.1 Using three folds

The merged dataset was divided into 66 percent for training and 33 percent for validation. The number of training samples for the normal class is 2084, while the number for the abnormal category is 1224. The number of validation samples for the normal category is 1042, and for the abnormal class it is 553. We will explain the obtained results for each fold, then we will explain the results of 3-fold cross-validation as the average for all three folds.

Table 6: Validation true positive rate of image and audio pre-trained CNN models utilizing three folds.

Model	3 Fold									
	Fold one			Fold two			Fold three			AVG
	Normal	Abnormal	Avg	Normal	Abnormal	Avg	Normal	Abnormal	Avg	
VGG16	0.92	0.72	0.82	0.90	0.79	0.85	0.83	0.87	0.85	**0.84**
VGG19	0.93	0.76	0.84	0.89	0.85	0.87	0.36	0.99	0.67	0.80
MobileNet	1.0	0.0	0.5	0.03	1.0	0.51	0.73	0.90	0.82	0.61
InceptionV3	0.67	0.92	0.79	0.98	0.56	0.77	1.0	0.0	0.5	0.68
InceptionResNetV2	1.0	0.0	0.5	0.0	1.0	0.5	0.0	1.0	0.5	0.5
Xception	0.92	0.62	0.77	0.91	0.62	0.77	0.0	1.0	0.5	0.68
DenseNet121	0.0	1.0	0.5	0.00	1.0	050	0.99	0.15	0.57	0.52
DenseNet169	0.0	1.0	0.5	1.0	0.0	0.5	0.0	1.0	0.5	0.5
DenseNet201	1.0	0.0	0.5	0.02	0.99	0.50	0.80	0.76	0.78	0.59
NASNetMobile	0.60	0.81	0.70	0.92	0.40	0.66	1.0	0.0	0.5	0.62
MobileNetV2	0.95	0.68	0.82	0.99	0.47	0.73	0.92	0.71	0.82	0.79
ResNet50	0.98	0.61	0.80	0.98	0.56	0.77	0.54	0.95	0.75	0.77

ResNet101	0.99	0.58	0.78	1.0	0.41	0.70	0.08	0.99	0.54	0.67
ResNet152	0.99	0.55	0.77	0.72	0.93	0.82	0.84	0.86	0.85	0.82
ResNet50V2	0.0	1.0	0.5	0.0	1.0	0.50	0.0	1.0	0.5	0.5
ResNet101V2	1.0	0.0	0.5	1.0	0.0	0.50	0.99	0.01	0.5	0.50
ResNet152V2	1.0	0.0	0.5	1.0	0.0	0.5	1.0	0.0	0.5	0.5
NASNetLarge	0.99	0.26	0.63	1.0	0.0	0.5	0.99	0.32	0.66	0.59
EfficientNetB0	0.99	0.39	0.69	0.92	0.92	0.92	1.0	0.34	0.67	0.76
EfficientNetB1	0.96	0.74	0.85	0.47	0.99	0.73	0.99	0.47	0.73	0.77
EfficientNetB2	0.26	0.98	0.62	0.99	0.46	0.72	0.62	0.98	0.80	0.71
EfficientNetB3	0.40	0.99	0.70	0.85	0.84	0.84	0.85	0.85	0.85	0.80
EfficientNetB4	0.05	0.99	0.52	0.93	0.68	0.81	0.60	0.98	0.79	0.71
EfficientNetB5	0.62	0.97	0.80	0.98	0.59	0.78	0.63	0.98	0.80	0.79
EfficientNetB6	0.97	0.60	0.78	0.99	0.47	0.73	0.77	0.84	0.81	0.77
EfficientNetB7	0.99	0.50	0.75	0.98	0.62	0.80	0.98	0.57	0.78	0.77
VGGish	0.93	0.80	0.86	0.91	0.80	0.85	0.93	0.77	0.85	**0.85**
YAMnet	0.95	0.59	0.77	0.94	0.64	0.79	0.95	0.62	0.78	0.78

–Fold1:

The VGGish and EfficientNetB1 models achieve the highest validation accuracy of 88%, while the VGG19 reaches 87% validation accuracy as seen in Figure 14. We can see in Table 4 that the number of layers of VGGish, EfficientNetB1 and VGG19 is 15, 186 and 19, respectively. There is no significant difference in validation accuracy between both models in spite of the significant difference between them in layers number. For the TPR in this fold, VGGish achieves the highest average positive rate of 86%. EfficientNetB1 has an average positive rate of 85%, and VGG19 has 84%. The results of both classes in models such as mobileNet, InceptionResV2, DenseNet201, ResNet50V2, and ResNet101V2 have defecated by imbalanced training data, as shown in Table 6. They have TPR =0.0% for the abnormal class. We conclude that the model's architectural complexity has no bearing on its performance. The imbalance in sample training plays a large role in validation accuracy. This confirms our findings in using 2-fold cross-validation.

–Fold2:

VGGish and VGG19 reach the highest validation accuracy of 87%. VGG16 and EfficientNetB7 attain 86% as displayed in Figure 14. VGGish and VGG19 accomplish the top average of validation TPR of 85% as demonstrated in Table 6. VGGish and VGG19 have not been heavily affected by imbalanced training samples, while other models have a significant impact such as DenseNet169, ResNet152V2 and ResNet152V2, achieving TPR=0.0%.

[H]

–Fold3:

VGGish reaches the highest validation accuracy of 87%. After VGGish, VGG16, MobileNetV2, ResNet152 and EfficientNeB3 have nearly the same validation accuracy of 85%. Figure 14 shows these results. For the average TPR, VGGish VGG16, ResNet152, and EfficientNetB3 achieve the highest average of 85% as shown in Table 6. They have nearly the same TPR in spite of their difference in the number of layers in VGGish, VGG16, ResNet152, and EfficientNetB3, which have 15, 16, 311 and 210 layers, respectively. Also, they have not been impacted much by imbalanced training samples, while other models have been heavily impacted such as InceptionResNetV2, Xception and ResNet50. This ensures our inference from 2-fold cross-validation that the models' depth has no impact on validation accuracy while imbalanced training data does.

We will demonstrate the average results of using 3-fold cross-validation. We get from three-fold cross-validation the average validation accuracy and TPR with the following consequences: VGGish achieves the highest average validation accuracy of 87%. VGG16 achieves 85% average validation accuracy, while MobileNetV2 and EfficientNetB7 reach 84%, respectively. Figure 14 displays these results. Figure 15 demonstrates the loss curves and training tests of the pre-trained CNN models using three folds for the VGGish model. Figure 16 presents an overview of the training validations and the loss curves of VGG16, MobileNetV2 and EfficientNetB7 pre-trained CNN models employing three folds. VGGish achieves the highest TPR of 85% and VGG16 achieves the highest TPR of 84% as seen in Table 6. Figure 17 shows an overview of training time stanzas validation precision using the image pretrained models and audio pretrained models. In the figure, we can see that VGGish and MobileNetV2 perform the highest average validation with less training time. We also see that VGG16 has 85% average validation accuracy but the highest training time among other models. We conclude by using 3-fold as an average. VGGish models outperformed the other models, either image or

audio models, on average validation accuracy and TPR. MobileNetV2 performs the best on average validation accuracy amongst image pretrained CNN models and YAMnet audio pre-trained CNN models. The depth of layers has not had an impact on the average validation accuracy; for example, MobileNetV2 (105 layers) and EfficientNetB7 (438) have the same average validation accuracy of 84%. We summarize the experimental results as follows: The highest average validation accuracy amongst the image and audio pre-trained CNN models using 2 or 3 cross-validation is the VGGish pre-trained CNN model. The highest average validation accuracy and the lowest average training time amongst the image pre-trained CNN models using 2-fold cross-validation is ResNet50. The highest average validation precision amongst the image pretrained CNN models that employed three-fold cross-validation is VGG16. MobileNetV2 has the lowest average training time using 3-fold cross-validation. The VGGish audio pre-trained CNN models as trained on the AudioSet dataset achieve better average validation accuracy than all image pre-trained CNN models and YAMnet audio pre-trained models. When looking at the YAMnet model, it trained on AudioSet but performed lower on average validation accuracy than some of the image pre-trained CNN models either using 2 or 3-fold cross-validation. The depth of models has no impact on validation accuracy. The imbalanced dataset affects the performance of models' TPR. VGGish can be used for CVDs recognition as audio pre-trained CNN models and ResNet50 as image pre-trained CNN models. Our results can be used as a benchmark for comparing research results.

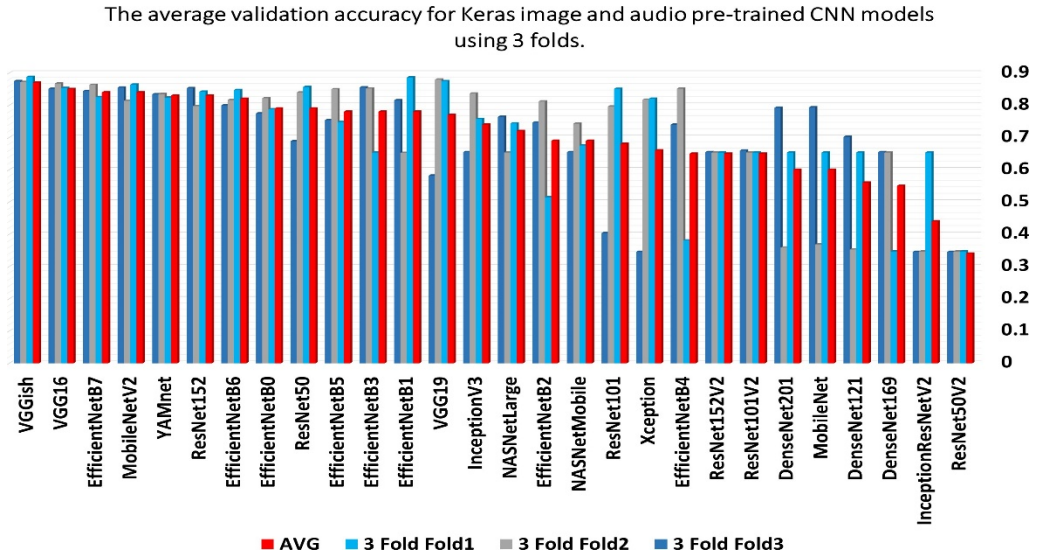

Fig. 14: The average validation accuracy for image and audio pre-trained CNN models utilizing three folds.

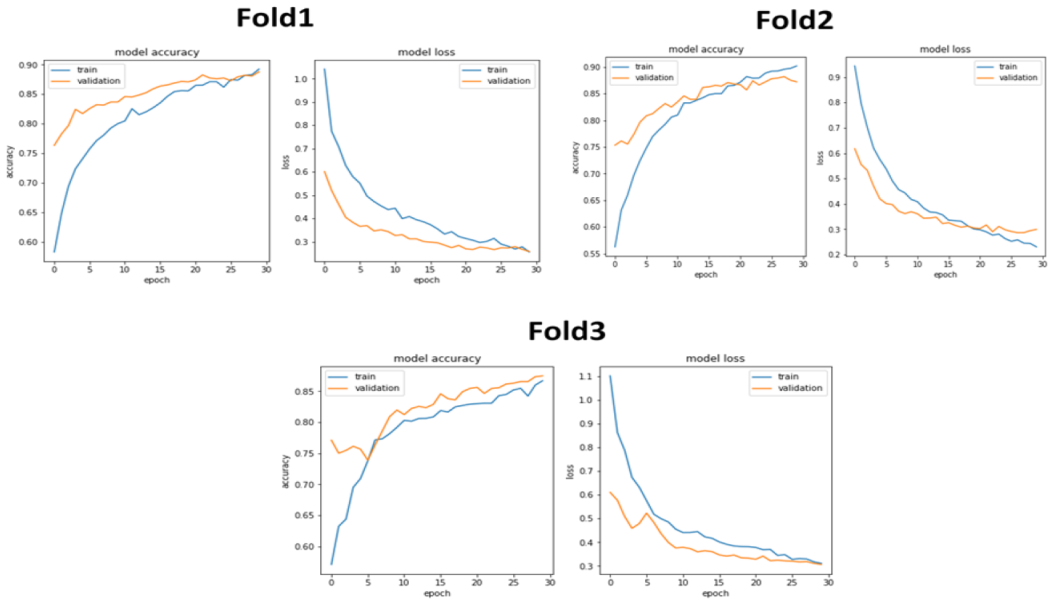

Fig. 15: The loss curves and training tests of the pre-trained CNN models using three folds for the VGGish model.

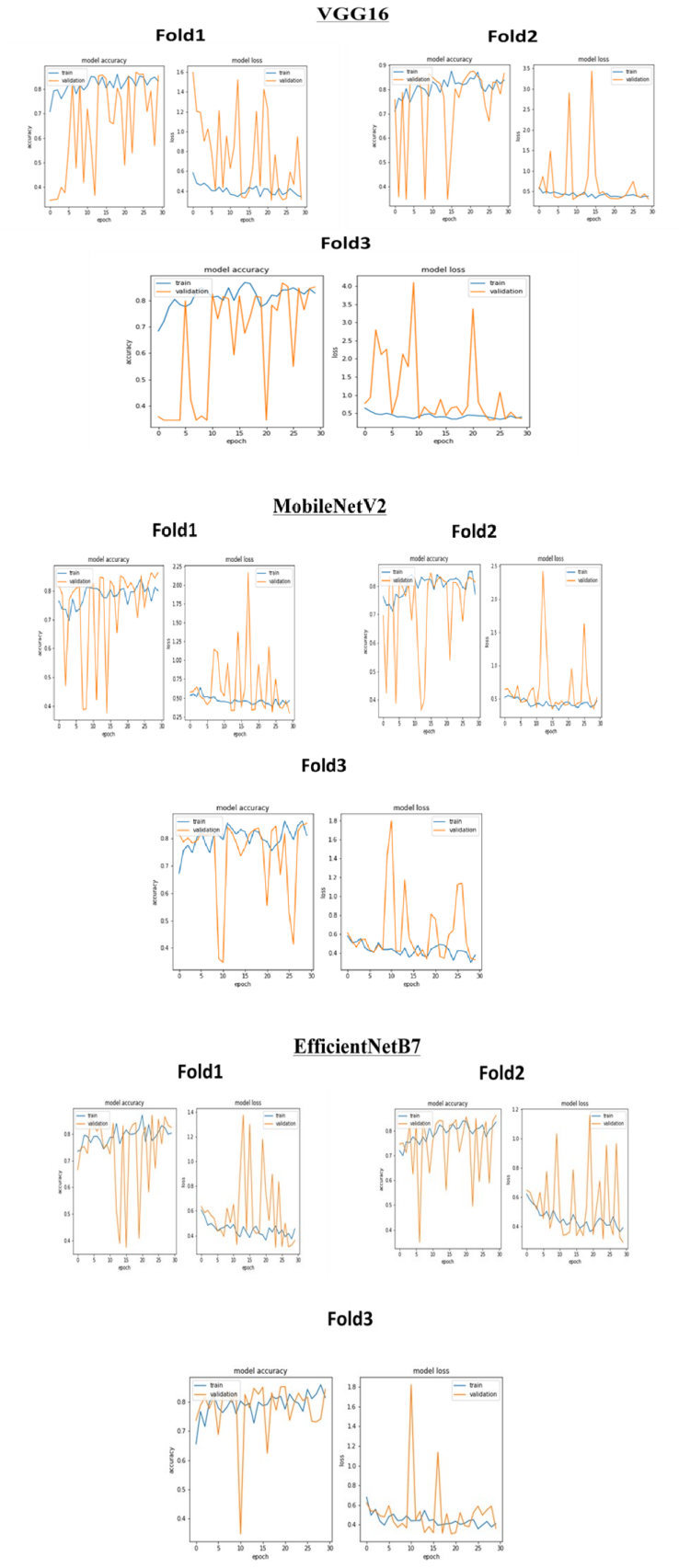

Fig. 16: The training validations and the loss curves of VGG16, MobileNetV2 and EfficientNetB7 pre-trained CNN models employ three folds.

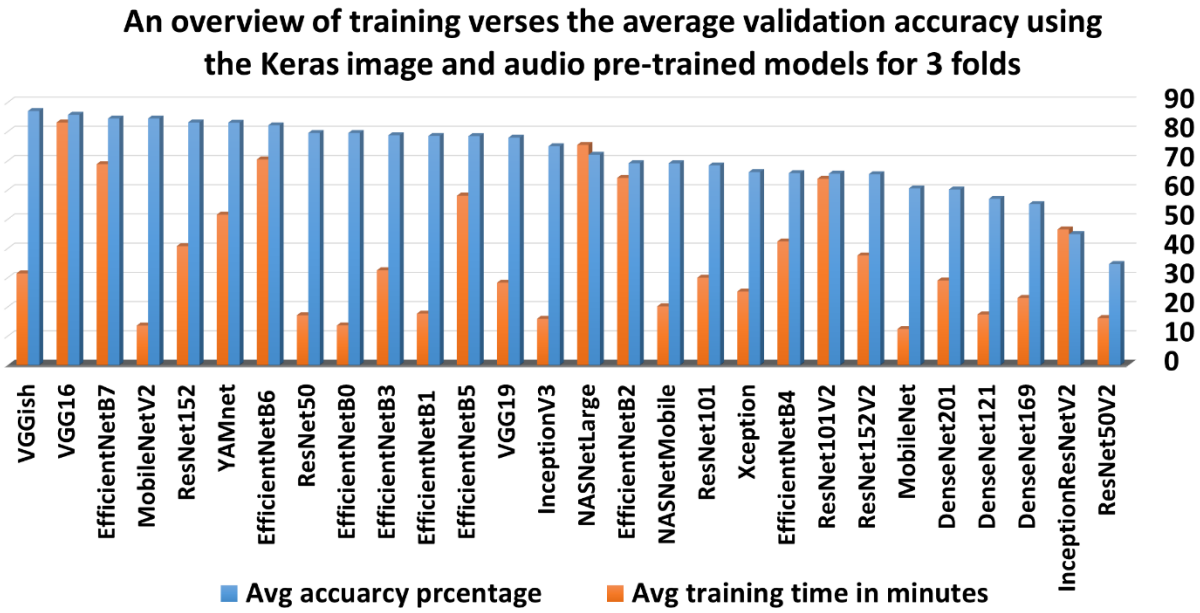

Fig. 17: Summary of training time against validation accuracy when employing the Keras image and audio pre-trained CNN models for three folds.

5 Conclusion

In this work, three public datasets were merged: PhysioNet 2016, PASCAL and open/Github to get enough samples for training the models. We benchmark the available Keras image pre-trained CNN models and the popular audio pre-trained CNN VGGish and YAMnet models as a starting point for research results comparison and to find out the best CNN model that can be used for CVD recognition. We cross-validate the datasets to 2 folds and 3 folds. The VGGish audio pre-trained CNN model attained the top average validation accuracy and true positive rate over all image pre-trained CNN models and the YAMnet audio pre-trained model for 2-fold and 3-fold cross-validation. ResNet50 and EfficientNetB1 using 2-fold and VGG16 using 3-fold achieved the best validation average accuracy. The depth of layers has no impact on validation accuracy as some of the models that have fewer layers produced better accuracy than models that have more layers. Imbalanced data impact the TPR. In future work, we will use the preprocessing methods for de-noising PCG signals to optimize the performance. A dataset should be built that contains the most common and difficult heart sounds of CVDs for diagnosis with a stethoscope and to apply transfer learning models for classification.

Conflict of interest

The authors declare that there is no conflict regarding the publication of this paper.

References

[1] Moran, Andrew E., et al. ”1990-2010 global cardiovascular disease atlas.” Glob Heart 9.1,3-16, (2014).

[2] MZC Lam et al. “Factors influencing cardiac auscultation proficiency in physician trainees”. In: Singapore medical journal 46.1, p. 11., (2005),

[3] Sami Alrabie, Mrhrez Boulares, and Ahmed Barnawi.“An Efficient Framework to Build Up Heart Sounds and Murmurs Datasets Used for Automatic Cardiovascular Diseases Classifications”. In: Enabling Machine Learning Applications in Data Science: Proceedings of Arab Conference for Emerging Technologies 2020, Springer, pp. 17–27, (2021).

[4] Muhaini Othman et al. “Empowering self-management through M-Health applications”. In: MATEC Web of Conferences. Vol. 150. EDP Sciences. 2018, p. 05018. [41] Jacob Peplinski et al. “FRILL: A Non-Semantic Speech Embedding for Mobile Devices”. In: arXiv preprint arXiv:2011.04609 (2020).

[5] Wei Chen et al. “Deep learning methods for heart sounds classification: a systematic review”. In: Entropy 23.6, p.667, (2021).

[6] Ali Raza et al. “Heartbeat sound signal classification using deep learning”. In: Sensors 19.21 (2019), p. 4819. [43]

Mark Sandler et al. “Mobilenetv2: Inverted residuals and linear bottlenecks”. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4510–4520,(2018).

[7] hasan, Nahian Ibn, and Arnab Bhattacharjee. ”Deep learning approach to cardiovascular disease classification employing modified ECG signal from empirical mode decomposition.” Biomedical signal processing and control 52, 128-140, (2019).

[8] Amit Krishna Dwivedi, Syed Anas Imtiaz, and Esther Rodriguez-Villegas. “Algorithms for automatic analysis 14 and classification of heart sounds–a systematic review”. In: IEEE Access 7,pp. 8316–8345 (2018).

[9] Dipanjan Sarkar, Raghav Bali, and Tamoghna Ghosh. Hands-On Transfer Learning with Python: Implement advanced deep learning and neural network models using TensorFlow and Keras. Packt Publishing Ltd, (2018).

[10] Sascha Grollmisch et al. “Analyzing the potential of pre-trained embeddings for audio classification tasks”. In: 2020 28th European Signal Processing Conference (EUSIPCO), IEEE,pp. 790–794, (2021).

[11] M. Kleiber, Hilgardia **6**, 315-332 (1932).

[12] Shuvo, Samiul Based, et al. ”CardioXNet: A novel lightweight deep learning framework for cardiovascular disease classification using heart sound recordings.” IEEE Access 9,: 36955-36967, (2021).

[13] G. D. Clifford, C. Liu, B. Moody, D. Springer, I. Silva, Q. Li, and R. G. Mark, “Classification of normal/abnormal heart sound recordings: The physionet/computing in cardiology challenge 2016,” in 2016 Computing in cardiology conference (CinC). IEEE,pp. 609–612, (2016).

[14] P. Bentley, G. Nordehn, M. Coimbra, and S. Mannor, “The PASCAL Classifying Heart Sounds Challenge 2011,” 2011, [Online]. Available: http://www.peterjbentley.com/heartchallenge/index.html,(2011).

[15] G.-Y. Son and S. Kwon, “Classification of heart sound signal using multiple features,” Applied Sciences, vol. 8, no. 12, p. 2344, (2018).

[16] HUSSAIN, Sayed Shahid, et al. Deep Learning Based Phonocardiogram Signals Analysis for Cardiovascular Abnormalities Detection. In: 2023 International Conference on Robotics and Automation in Industry (ICRAI). IEEE,p. 1-6, (2023).

[17] GONZA´ LEZ–RODR´IGUEZ, Cristo´bal; ALONSO–ARE´ VALO, Miguel A.; GARC´IA–CANSECO, Elo´ısa. Robust Denoising of Phonocardiogram Signals using Time-Frequency Analysis and U-Nets. IEEE Access, (2023).

[18] Omer DEPERL ¨ ˙IGLU. “Classification of segmented heart sounds with Artificial Neural Networks”. In: International Journal of Applied Mathematics Electronics and Computers 6.4, pp. 39–44 (2018).

[19] Muqing Deng et al. “Heart sound classification based on improved MFCC features and convolutional recurrent neural networks”. In: Neural Networks 130, pp. 22–32. (2020).

[20] Shahid Ismail et al. “PCG classification through spectrogram using transfer learning”. In: Biomedical Signal Processing and Control 79, pp. 104075, (2023).

[21] SHUVO, Samiul Based, et al. NRC-Net: Automated noise robust cardio net for detecting valvular cardiac diseases using optimum transformation method with heart sound signals. arXiv preprint arXiv:2305.00141, (2023).

[22] Mehrez Boulares, Tarik Alafif, and Ahmed Barnawi.“Transfer learning benchmark for cardiovascular disease recognition”. In: IEEE Access 8, pp. 109475–109491, (2020).

[23] Mohammad Khaleel MAlnajjar and Samy S Abu-Naser.“Heart Sounds Analysis and Classification for Cardiovascular Diseases Diagnosis using Deep Learning”. In: (2022).

[24] Omair Rashed Abdulwareth Almanifi et al. “The Classification of Heartbeat PCG Signals via Transfer Learning”. In: Recent Trends in Mechatronics Towards Industry 4.0. Springer, pp. 49–59, (2022).

[25] XIANG, Menghui, et al. Research of heart sound classification using two-dimensional features. Biomedical Signal Processing and Control, 79: 104190,(2023).

[26] Uddipan Mukherjee and Sidharth Pancholi. “AVisual Domain Transfer Learning Approach for Heartbeat Sound Classification”. In: arXiv preprint arXiv:2107.13237, (2021).

[27] XIANG, Menghui, et al. Research of heart sound classification using two-dimensional features. Biomedical Signal Processing and Control,79: 104190, (2023).

[28] Zhihua Wang et al. “Exploring interpretable representations for heart sound abnormality detection”. In:

Biomedical Signal Processing and Control 82, p. 104569 (2023).

[29] Miao Wang et al. “Transfer learning models for detecting six categories of phonocardiogram recordings”. In: Journal of Cardiovascular Development and Disease 9.3, p. 86, (2022).

[30] Guangyang Tian et al. “Imbalanced heart sound signal classification based on two-stage trained dsanet”. In: Cognitive Computation 14.4, pp. 1378–1391, (2022).

[31] Neeraj Baghel, Malay Kishore Dutta, and Radim Burget. “Automatic diagnosis of multiple cardiac diseases from PCG signals using convolutional neural network”. In: Computer Methods and Programs in Biomedicine 197, p. 105750, (2020).

[32] Jay Karhade et al. “Time–Frequency-Domain Deep Learning Framework for the Automated Detection of Heart Valve Disorders Using PCG Signals”. In: IEEE Transactions on Instrumentation and Measurement 71 pp. 1–11. DOI: 10.1109/TIM.2022.3163156, (2022).

[33] Bin Xiao et al. “Heart sounds classification using a novel 1-D convolutional neural network with extremely low parameter consumption”. In: Neurocomputing 392, pp. 153–159, (2020).

[34] Fengquan Dong et al. “Machine listening for heart status monitoring: Introducing and benchmarking hss—the heart sounds shenzhen corpus”. In: IEEE journal of biomedical and health informatics 24.7, pp. 2082–2092 (2019).

[35] NGUYEN, Minh Tuan; LIN, Wei Wen; HUANG, Jin H. Heart Sound Classification Using Deep Learning Techniques Based on Log-mel Spectrogram. Circuits, Systems, and Signal Processing, 42.1: 344-360, (2023).

[36] GEMMEKE, Jort F., et al. Audio set: An ontology and human-labeled dataset for audio events. In: 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2017. p. 776-780.

[37] Eleni Tsalera, Andreas Papadakis, and Maria Samarakou.“Comparison of Pre-Trained CNNs for Audio Classification Using Transfer Learning”. In: Journal of Sensor and Actuator Networks 10.4, p. 72, (2021).

[38] Franc¸ois Chollet. “Xception: Deep learning with depthwise separable convolutions”. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1251–1258, (2017).

[39] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: arXiv preprint arXiv:1409.1556 (2014).

[40] Kaiming He et al. “Deep residual learning for imagerecognition”. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016, pp. 770–778,(2016).

[41] Kaiming He et al. “Identity mappings in deep residual networks”. In: European conference on computer vision. Springer. pp. 630–645, (2016).

[42] Christian Szegedy et al. “Rethinking the inception architecture for computer vision”. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826, (2016).

[43] Christian Szegedy et al. “Inception-v4, inception-resnet and the impact of residual connections on learning”. In:Thirty-first AAAI conference on artificial intelligence,(2017).

[44] Andrew G Howard et al. “Mobilenets: Efficient convolutionalneural networks for mobile vision applications”.In: arXiv preprint arXiv:1704.04861 (2017).

[45] Mark Sandler et al. “Mobilenetv2: Inverted residuals and linear bottlenecks”. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4510–4520,(2018).

[46] Gao Huang et al. “Densely connected convolutional networks”. In: Proceedings of the IEEE conference on computer vision and pattern recognition.pp. 4700–4708,(2017).

[47] Barret Zoph et al. “Learning transferable architectures for scalable image recognition”. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8697–8710,(2018).

[48] Mingxing Tan and Quoc Le. “Efficientnet: Rethinking model scaling for convolutional neural networks”. In: International conference on machine learning. PMLR. pp. 6105–6114, (2019).

[49] Eleni Tsalera, Andreas Papadakis, and Maria Samarakou. “Comparison of Pre-Trained CNNs for Audio Classification Using Transfer Learning”. In: Journal of Sensor and Actuator Networks 10.4, p. 72, (2021).

[50] Qiuqiang Kong et al. “Panns: Large-scale pretrained audio neural networks for audio pattern recognition”. In:

IEEE/ACM Transactions on Audio, Speech, and Language Processing 28, pp. 2880–2894, (2020).

[51] Jort F Gemmeke et al. “Audio set: An ontology and human-labeled dataset for audio events”. In: 2017 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE.pp. 776–780,(2017).

[52] Anurag Kumar, Maksim Khadkevich, and Christian F¨ugen. “Knowledge transfer from weakly labeled audio using convolutional neural network for sound events and scenes”. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE. pp. 326–330, 2018.

[53] Jason Cramer et al. “Look, listen, and learn more: Design choices for deep audio embeddings”. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE. pp. 3852–3856, (2019).

[54] Joel Shor et al. “Towards learning a universal nonsemantic representation of speech”. In: arXiv preprint arXiv:2002.12764 (2020).

[55] Jacob Peplinski et al. “FRILL: A Non-Semantic Speech Embedding for Mobile Devices”. In: arXiv preprint arXiv:2011.04609 (2020).

[56] Shawn Hershey et al. “CNN architectures for largescale audio classification”. In: 2017 ieee international conference on acoustics, speech and signal processing (icassp). IEEE.pp. 131–135,(2017).

[57] Nikolaos Vryzas et al. “A web crowdsourcing framework for transfer learning and personalized speech emotion recognition”. In: Machine Learning with Applications 6, p. 100132, (2021).

[58] Yuan Liu et al. “AI for earth: rainforest conservation by acoustic surveillance”. In: arXiv preprint arXiv:1908.07517 (2019).

[59] Huaping Liu et al. “An ensemble system for domesticactivity recognition”. In: DCASE2018 Challenge, Tech. Rep. (2018).

[60] Gianmarco Cerutti et al. “Neural Network Distillation on IoT Platforms for Sound Event Detection.” In: Interspeech.pp. 3609–3613,2019.

[61] Honglie Chen et al. “Vggsound: A large-scale audiovisual dataset”. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE. pp. 721–725, (2020).

[62] Bongjun Kim. “Convolutional neural networks with transfer learning for urban sound tagging”. In: DCASE2019 Challenge (2019).

[63] Alberto Tena, Francesc Clari‘a, and Francesc Solsona.“Automated detection of COVID-19 cough”. In:Biomedical Signal Processing and Control 71 ,p. 103175,(2022).

[64] George Boateng and Tobias Kowatsch. “Speech emotion recognition among elderly individuals using multimodal fusion and transfer learning”. In: Companion Publication of the 2020 International Conference on Multimodal Interaction. pp. 12–16,(2020).

[65] Nebras Sobahi et al. “Explainable COVID-19 detection using fractal dimension and vision transformer with Grad-CAM on cough sounds”. In: Biocybernetics and Biomedical Engineering 42.3, pp. 1066–1080, (2022).

[66] Zafi Sherhan Syed, Sajjad Ali Memon, and Abdul Latif Memon. “Deep acoustic embeddings for identi15 fying Parkinsonian speech”. In: International Journal of Advanced Computer Science and Applications 11.10 (2020).

[67] Madison Cohen-McFarlane et al. “Impact of face covering models on respiratory sound classification applications”. In: IEEE Sensors Applications Symposium (SAS). IEEE. 2022, pp. 1–6, (2022).