Classification of Echocardiogram Views using Deep Learning Models

Lee Qi Zian1, Yogan Jaya Kumar2, Ong Sing Goh3 and Zi Ye4
1,2,3,4Centre for Advanced Computing Technology, Faculty of Information and Communication Technology
Universiti Teknikal Malaysia Melaka, Hang Tuah Jaya, 76100 Durian Tunggal, Melaka
4Department of Information Technology, Wenzhou Polytechnic, China
Email: 1alexlee7399@hotmail.com, 2yogan@utem.edu.my, 3goh@utem.edu.my, 4yezi1022@gmail.com

Abstract— Cardiovascular disease has always been one of the main causes of death in the world. One of the way to diagnose cardiovascular disease is by using echocardiography. However, this method of diagnosis requires cardiovascular knowledge and sometimes it can be very hard to recognize the views of echocardiogram without expertise in that particular field. The main purpose of this study is to develop and compare deep learning models to classify the views of echocardiogram. VGG16, VGG19, InceptionV3 and MobileNet are used to develop the model to classify the echocardiogram view. After training, the models are evaluated by using classification measures, confusion matrix and confidence test. From the experimental findings, the VGG16 model obtained the best result on both F1 score and accuracy. However, for the confidence score test, MobileNet model achieved better results.

Index Terms— Echocardiogram, Deep Learning, Convolutional Neural Network, Digital Imaging and Communication in Medicine (DICOM).

I. Introduction

Cardiovascular disease (CVD) is one of the main causes of death globally taking an estimate of 17.9 million lives each year according to the World Health Organization [35]. To combat this issue, various computer-aided diagnostic systems (CAD) are used to assist health personals. Echocardiography is one of the CAD that is a frequently used as a screening modality for asymptomatic patients as well as in order to diagnose and manage patients with complex cardiovascular disease. However, when diagnosing a cardiovascular disease, a cardio expert is always required for the job. With the number of CVD patients arising, it is difficult for hospitals to have the cardio experts available all time. This can slow down the process of treating patients and may cost their lives. With the emerging importance of imaging to cardiovascular care, we believe that an automated pipeline for interpreting cardiovascular imaging can improve perioperative risk stratification, manage the cardiovascular risk of patients with oncologic disease undergoing chemotherapy, and aid in the diagnosis of cardiovascular disease [1].

While other works applying machine learning to medical imaging required the explanations of images by human experts, the clinical workflow for echocardiography is often reported through structured reporting systems. The ability to use previous annotations and interpretations from clinical reports can greatly accelerate adoption of machine learning in medical imaging. Given the availability of previously annotated clinical reports, the density of information in image and video datasets, and many available machine learning architectures already applied to image datasets, echocardiography is a high impact and highly tractable application of machine learning in medical imaging.

The aim this study is to use image classification technique using Convolutional Neural Network to train the models for recognizing the views of echocardiogram. The VGG16, VGG19, InceptionV3 and MobileNet model will be used in the training. The four models will then be evaluated by using accuracy, validation loss, confusion matrix and confidence test to determine which model has the best performance for the classification task.

The rest of the paper will be structured as follows: Section 2 will discuss the related works. Section 3 presents the classification models. Section 4 discusses the result of the study Finally Section 5 concludes the work of this study.

II. RELATED WORKS

In this section, we will discuss the related works based on the neural network architectures used in this study.

A. Artificial Neural Network and Convolutional Neural Network

Artificial neural network (ANN) is a subset in machine learning. Artificial intelligence techniques are used to enable machine like computer to learn automatically and improve from past experience by observing historical data [4]. As for ANN, they are inspired by human biological brain neuron shown in Fig. 1. This enable computer to replicate the way human learn. ANN generally consist of input layer, hidden layers and output layers. They are good at finding pattern in data which are too numerous for human programmer to extract and program machine to recognize [3]. A single layer neural network is called perceptron and it provides single output [5]. A perceptron generally contains input, bias, activation function and output. An activation function is very important as it enables ANN to learn and understand complicated patterns. The main purpose of activation function is to convert an input signal to an output signal. Deep learning is an ANN that consist multiple processing layers. The processing layer consist of complex structures and non-linear transformations which enable the model to process high level abstraction in data [3].

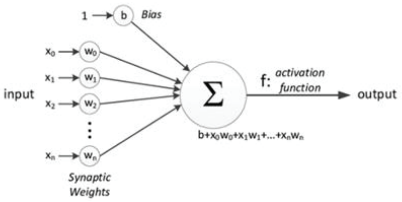

Fig. 1 Diagram of a simple Perceptron in ANN [3]

Convolutional Neural Network (CNN) is a popular deep learning neural network consisting many layers: convolutional layer, non-linearity layer, pooling layer and fully-connected layer [6]. The architecture of a CNN is comparable to the connectivity pattern of the Neurons in the human brain and was inspired by the organization of the Visual Cortex [7]. CNN can successfully capture the Spatial and Temporal dependencies of an image through relevant filters. It perform better in fitting to the image dataset due to the reduction in the number of parameters involved and reusability of weights[7]. The architecture of CNN is shown Fig. 2. In the next subsections, we will describe the four CNN models used in this study.

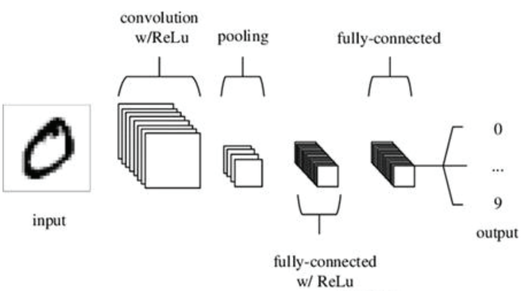

Fig. 2 Simple CNN architecture, comprised of just five layers [7]

B. VGG16 and VGG19

VGG-16 is a convolutional neural network with 16 layers which was proposed by K. Simonyan and A. Zisserman [9]. A VGG16 CNN uses very small 3×3 receptive fields throughout the whole net, which are convolved with the input at every pixel instead of large receptive fields [9].

The first and second layer have 64 channels of 3×3 filter size and same padding [9]. Next a max pool layer of stride (2, 2), two layers which have convolution layers of 256 filter size and filter size (3, 3) [9]. This is followed by a max pooling layer of stride (2, 2) which is same as the previous layer. Then there are 2 convolution layers of filter size (3, 3) and 256 filter. After that there are 2 sets of 3 convolution layer and a max pool layer. Each layers have 512 filters of (3, 3) size with same padding. This image is then passed to the stack of two convolution layers. Fig. 3 shows the architecture of VGG16 used by K. Simonyan and A. Zisserman [9]. Same as VGG16, VGG19 is also a model proposed by K. Simonyan and A. Zisserman. The difference between VGG16 and VGG19 is only on their number of weight layers. VGG16 consist of 16 layers while VGG19 consist of 19 layers. Fig. 4 shows the architecture of VGG19.

The VGGs model have been previously used for echocardiogram classification. The study by Ali Madani et. al [31] uses VGG16 followed by two fully connected layers of 1028 and 512 nodes. The model performed well on the echocardiogram, achieving 91.7% overall accuracy on low resolution echocardiogram.

Another study conducted by See Yee Tan et. al [32] use the VGG16 model followed by 500-500-8 fully connected layers for the echocardiogram classification. The model achieved 93.94% accuracy on the test set.

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Fig 3 VGG16 Architecture [9]

Layer name	#Filters	#Parameters	#Activations
input			150K
conv1_1	64	1.7K	3.2M
conv1_2	64	36K	3.2M
max pooling			802K
conv2_1	128	73K	1.6M
conv2_2	128	147K	1.6M
max pooling			401K
conv3_1	256	300K	802K
conv3_2	256	600K	802K
conv3_3	256	600K	802K
conv3_4	256	600K	802K
max pooling			200K
conv4_1	512	1.1M	401K
conv4_2	512	2.3M	401K
conv4_3	512	2.3M	401K
conv4_4	512	2.3M	401K
max pooling			100K
conv5_1	512	2.3M	100K
conv5_2	512	2.3M	100K
conv5_3	512	2.3M	100K
conv5_4	512	2.3M	100K
max pooling			25K
fc6		103M	4K
fc7		17M	4K
output		4M	1K

Fig. 4 VGG19 Architecture [9]

C. Inception

Inception is a CNN model by Google and got its start as a module of GoogleNet. The model that we use which is InceptionV3, is the 3rd edition of Google's Inception Convolutional Neural Network [26]. Inception is a deep neural network that is computational efficient for various use cases scenarios such as mobile vision and big-data scenarios [26]. InceptionV3 focuses on using less computational power by modifying the previous inception model by factorized convolutions, regularization, dimension reduction, and parallelized computations to loosen the constraint for easier model adaptation [27]. The architecture of InceptionV3 is built step by step progressively by first Factorized Convolution as it can improve the network efficiency. After that, the bigger convolution of the previous model is replaced by smaller model as it can speed up the training process [26]. Fig. 5 shows the architecture of InceptionV3 model.

In a study conducted by Wang Cheng et. al [34], an InceptionV3 model was used as a feature extractor along with Softmax, Logistic and SVM classifier to classify Pulmonary image. The model achieved 86.40% accuracy in classifying the images.

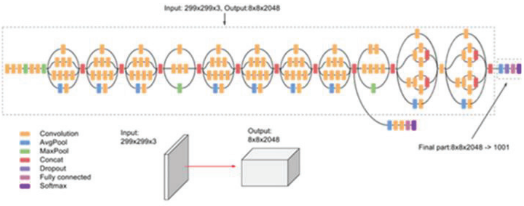

Fig. 5 InceptionV3 Architecture [30]

D. MobileNet

MobileNet is a CNN from google which use a depth-wise separable convolution to build light weight deep neural networks based on streamlined architecture [29]. MobileNet has 2 simple global hyper parameters as stated in [28], which has an efficiency trade-off between latency and accuracy. The 2 global hyper parameters allow to select the right size of the model based in the constraint of the problem [28]. The small size and lightweight model of MobileNet make it suitable for mobile and embedded vision which has limited computational power [29]. Fig. 6 show the overall architecture of MobileNet.

A research conducted by Ioannis et. al [31] uses MobileNet model to detect Covid-19 disease from X-ray images. The research are conducted by using 5 models which are VGG19, MobileNet v2, Inception, Xception and Inception ResNet v2. The MobileNet v2 and VGG19 model turned out to be the best model for this problem achieving 92.85% and 93.48% accuracy respectively.

Table 1. MobileNet Body Architecture

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32$ dw	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64$ dw	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128$ dw	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256$ dw	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
$5\times$ Conv dw / s1	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
$5\times$ Conv / s1	$1 \times 1 \times 512 \times 512$	$14 \times 14 \times 512$
Conv dw / s2	$3 \times 3 \times 512$ dw	$14 \times 14 \times 512$
Conv / s1	$1 \times 1 \times 512 \times 1024$	$7 \times 7 \times 512$
Conv dw / s2	$3 \times 3 \times 1024$ dw	$7 \times 7 \times 1024$
Conv / s1	$1 \times 1 \times 1024 \times 1024$	$7 \times 7 \times 1024$
Avg Pool / s1	Pool 7×7	$7 \times 7 \times 1024$
FC / s1	1024×1000	$1 \times 1 \times 1024$
Softmax / s1	Classifier	$1 \times 1 \times 1000$

Fig 6 MobileNet Architecture [33]

E. DICOM Images dataset

DICOM stands for Digital Imaging and Communications in Medicine. The DICOM image provides important information that can be used in interface specifications to enable network connectivity among a variety of vendors' products. This allow hospital to format and exchange medical images and associated information, both within the hospital and also outside the hospital [2]. In this study, all the DICOM images are acquired from a private hospital in Melaka, Malaysia and all the data of patients are removed for privacy purpose. There are total 9 classes for the images which are AP2, AP3, AP4, AP5, PLAX, PSAX-AP, PSAX-AV, PSAX-MID and PSAX-MV. Each class represent the different view of echocardiogram. The number of training image are 8997 images and about 900 images for each class. As for the validation dataset we will use 10% of the training dataset and 897 images as test data.

The training dataset will be used to train the CNN model and we will be using Categorical Crossentropy as the loss function. For the optimizer, we will be using the Adam optimizer with learning rate of 0.001 and learning rate decay of 0.0001. The Adam optimizer is chosen for this study as the Adam optimize r is appropriate for problems with very noisy or sparse gradients and it can make use of the average of the second moments of the gradients to adjust the learning rate parameter. The learning rate of 0.001 and learning rate decay of 0.0001 is used as they are the optimal choice for our model as stated by David Mack in [29]. Then we will test the accuracy of classification of each of these models using test datasets while validation datasets will useful to calculate the loss due to misclassification. After the result is obtained, the determination of the best deep learning model will be scored using their rank in accordance to their performance. Fig. 7 shows the echocardiogram views that were used in this study.

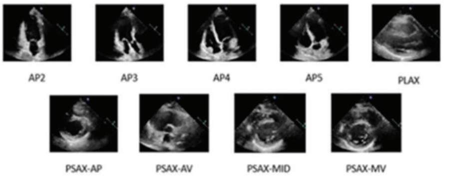

Fig. 7 Echocardiogram Views

III. IMPLEMENTATION

The implementation of this study was done using Python and has two main parts which are data pre-processing and model training. Before the data pre-processing process, all the required Python library will be imported. The overall flow of the study is shown in Fig.8.

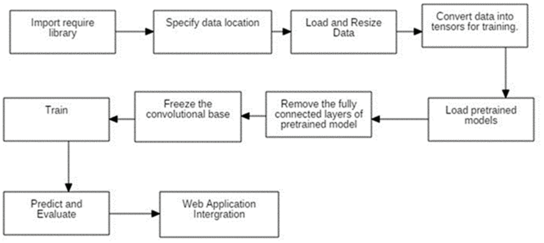

Fig.8 System Flow of Classification Software

A. Data preprocessing

In this stage, the data will be loaded into our system and is resized to 224 x 224. We reduce the size because it can lighten the computational power required to train the data. The data is feed into the neural network in 3 channels which is RGB (Red, Green and Blue). Before feeding the data into the neural network, the data will be convert into tensor. Tensors are multi-dimensional arrays with a uniform type (called a dtype). You can see all supported dtypes at tf.dtypes.DType. In this process the data will also be spilted.

B. Model Architecture and Training

After finishing the data pre-processing, the pre trained model will be load into our system. Instead of building a CNN from scratch, we will use pre trained model to train our data. This can speed up the training. The classification layer of the pre-trained model will be removed as we only need the convolutional layer for feature extraction. We will freeze all layer in convolutional base to prevent the weight from updating. It is important to freeze the convolutional base before you compile and train the model. Layer freezing means layer weights of a trained model are not changed when they are reused in a subsequent downstream task as they remain frozen, as shown in Fig. 10. Essentially when backpropagation is done during training these layers weights are untouched. So in transfer learning, we reuse by freezing or fine tuning the layers of a model. Since we will not be using the classifier from the pre trained model, we will build our own new classifier. Our classifier will consist of four 512 fully connected layers with ReLu activation, batch normalization and dropout followed by output layer with softmax activation. After we build our classifier, the pre trained model will be flatten and stack with our classifier. An early stopping criteria is implemented to stop the training if the validation loss is not improving for 3 epoch. The model is then trained until it achieves its highest accuracy. Fig.9 shows an example of VGG16 model architecture used in the study.

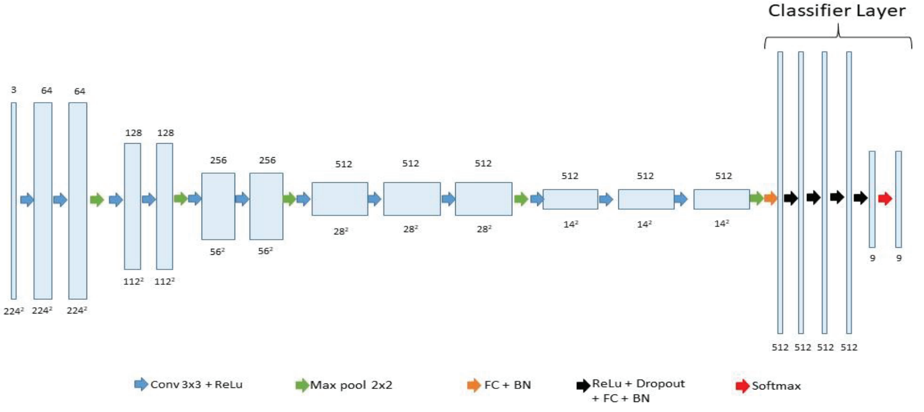

Fig. 9 An example of VGG16 architecture used in this study followed by 512-512-512-512 fully connected layers

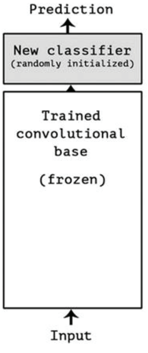

Fig.10 Freezing convolutional base and adding new classifier [11]

IV RESULT AND DISCUSSION

After the training, evaluation was done on each model to evaluate their performance. The performance is evaluated by using F1 Score and confidence of the models on each of the echocardiogram class.

From Fig. 11 and Table 1, it can be observed that VGG16 performed better than VGG19 while InceptionV3 and MobileNet has roughly the same performance. VGG16 model achieved lowest data performance on the PSAX-MV class. It has the lowest Recall and F1-Score which are 0.74 and 0.85 respectively. The VGG19 model has a lower performance compared to VGG16. The VGG19 model performs poorly on the PSAX-AP, PSAX-MID and PSAX-MV. From the accuracy perspective in Table 2, VGG16 has the best performance followed by InceptionV3, MobileNet and VGG19.

TABLE 1
F1 SCORE FOR EACH CLASS

	VGG16	VGG19	InceptionV3	MobileNet
AP2	0.96	0.95	1.00	1.00
AP3	0.96	1.00	1.00	1.00
AP4	0.92	0.95	0.99	0.92
AP5	0.94	0.97	0.99	0.91
PLAX	1.00	1.00	1.00	1.00
PSAX-AP	0.89	0.57	0.96	0.94
PSAX-AV	1.00	0.10	0.99	0.99
PSAX-MID	0.98	0.34	0.75	0.60
PSAX-MV	0.95	0.61	0.72	0.76

TABLE 2
ACCURACY FOR EACH MODEL

	VGG16	VGG19	InceptionV3	MobileNet
Accuracy	0.95	0.83	0.93	0.91

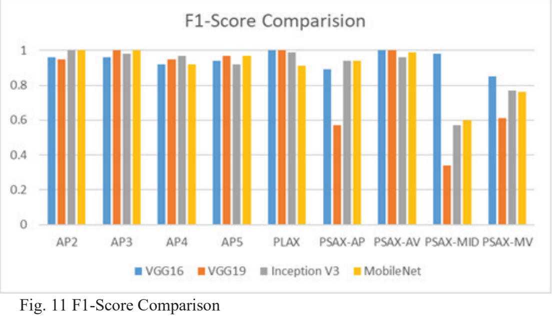

Fig. 11 F1-Score Comparison

In Table 3, VGG16 can classify most of the view well except for PSAX-MV which has the most misclassification, 24 misclassification followed by AP4 and AP3 and PSAX-MID which are 13, 7 and 1. VGG 19 model has a lower performance in classifying. From VGG19 confusion matrix, we can see that the model performs poorly on PSAX-AP and PSAX-MID where PSAX-AP has 60 misclassification and PSAX-MID has 74 misclassification. However, InceptionV3 and MobileNet have the roughly same performance on each class while both model perform poorly on PSAX-MID which InceptionV3 has 51 misclassifications and MobileNet has 50 misclassifications. Similar outcome can also be seen while comparing the F1 Score on Figure 9 which InceptionV3 and MobileNet perform poorly on PSAX-MID while VGG16 perform good across the board.

Confidence test is performed on a PSAX-AV image by using a high resolution image and low resolution image from the PSAX-AV class. From the confidence test in Fig. 12, we can see that VGG16, InceptionV3 and MobileNet has acceptable confidence on the high resolution PSAX-AV image while VGG19 has a low confidence score. From the test on a low resolution PSAX-AV image on Fig. 13, we can see that the problem on low confidence is more significant as the VGG16 and VGG19 have very low confidence score. However the InceptionV3 model has the same confidence score and MobileNet model have slightly lower confidence score on low resolution image. The poor performance of VGG16 and VGG19 model on low resolution image is because that the VGG model cannot extract very complex feature as it is a simple stack of convolutional and max-pooling layers followed by one another and finally fully connected layers compared to the deeper and complex architecture of InceptionV3 and MobileNet. This may cause VGG16 and VGG19 model unable to perform well on difficult task. Unlike the VGGs model, InceptionV3 consist of 1x1 filters also known as pointwise convolutions followed by convolutional layers with different filter sizes applied simultaneously [26][36]. MobileNet's depthwise separable convolution architecture also have comparable performance to Inception as stated in [30]. This allows Inception and MobileNet model to learn more complex features as it has more hidden layer than VGGs model. Thus making the InceptionV3 and MobileNet model more reliable on classifying echocardiogram.

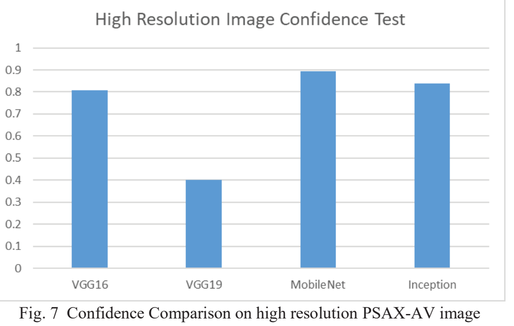

Fig. 7 Confidence Comparison on high resolution PSAX-AV image

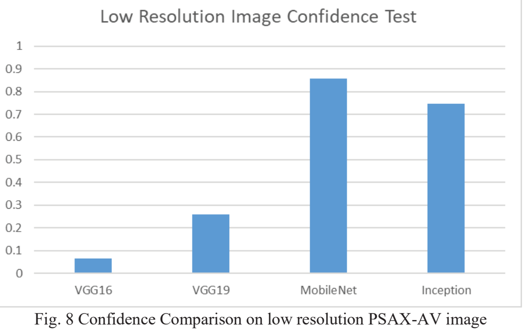

Fig. 8 Confidence Comparison on low resolution PSAX-AV image

TABLE 3
CONFUSION MATRIX

Model	Confusion Matrix	AP2	AP3	AP4	AP5	PLAX	PSAX-AP	PSAX-AV	PSAX-MID	PSAX-MV
VGG16	AP2	95	0	0	0	0	0	0	0	0
	AP3	7	88	0	0	0	0	0	0	0
	AP4	0	0	81	15	0	0	0	0	0
	AP5	0	0	0	100	0	0	0	0	0
	PLAX	0	0	0	0	100	0	0	0	0
	PSAX-AP	0	0	0	0	0	100	0	0	0
	PSAX-AV	0	0	0	0	0	0	97	0	0
	PSAX-MID	0	0	0	0	0	1	0	92	0
	PSAX-MV	0	0	0	0	0	24	0	3	77
VGG19		AP2	AP3	AP4	AP5	PLAX	PSAX-AP	PSAX-AV	PSAX-MID	PSAX-MV
	AP2	95	0	0	0	0	0	0	0	0
	AP3	0	95	0	0	0	0	0	0	0
	AP4	7	0	89	0	0	0	0	0	0
	AP5	1	0	5	94	0	0	0	0	0
	PLAX	0	0	0	0	100	0	0	0	0
	PSAX-AP	0	0	0	0	0	40	0	0	56
	PSAX-AV	0	0	0	0	0	0	97	0	0
	PSAX-MID	0	0	0	0	0	1	0	19	74
	PSAX-MV	0	0	0	0	0	1	0	3	103
InceptionV3		AP2	AP3	AP4	AP5	PLAX	PSAX-AP	PSAX-AV	PSAX-MID	PSAX-MV
	AP2	95	0	0	0	0	0	0	0	0
	AP3	0	95	0	0	0	0	0	0	0
	AP4	0	0	96	0	0	0	0	0	0
	AP5	0	0	5	88	0	0	1	0	6
	PLAX	0	0	0	0	98	0	0	0	2
	PSAX-AP	0	0	0	0	0	97	0	3	0
	PSAX-AV	0	0	0	0	0	6	91	0	0
	PSAX-MID	0	0	0	0	0	3	0	39	51
	PSAX-MV	0	0	0	0	0	1	0	1	102
MobileNet		AP2	AP3	AP4	AP5	PLAX	PSAX-AP	PSAX-AV	PSAX-MID	PSAX-MV
	AP2	95	0	0	0	0	0	0	0	0
	AP3	0	95	0	0	0	0	0	0	0
	AP4	0	0	96	0	0	0	0	0	0
	AP5	0	0	17	83	0	0	0	0	0
	PLAX	0	0	0	0	100	0	0	0	0
	PSAX-AP	0	0	0	0	0	100	0	0	0
	PSAX-AV	0	0	0	0	0	0	96	0	1
	PSAX-MID	0	0	0	0	0	3	0	40	50
	PSAX-MV	0	0	0	0	0	0	0	0	104

V. Conclusion

This study compares four deep learning models namely VGG16, VGG19, InceptionV3 and MobileNet to identify the views of echocardiogram. The observation from this study shows that VGG16 has the best performance by looking at the classification results. However while testing the confidence, InceptionV3 and MobileNet performs better than VGG16. The model using MobileNet has a high confidence even on a low-resolution image despite the low F1 score on PSAX-MID images. A high confidence score is required because we need the model to be able to classify a medical image correctly and consistently in the medical field. The low computational cost of MobileNet model will also give an edge on running on low computational device such as mobile devices.

Acknowledgment

The authors would like to thank Universiti Teknikal Malaysia Melaka for supporting this research.

References

[1] Weinhaus A.J., Roberts K.P. (2005) Anatomy of the Human Heart. In: Iaizzo P.A. (eds) Handbook of Cardiac Anatomy, Physiology, and Devices. Humana Press. https://doi.org/10.1007/978-1-59259-835-9_4

[2] Bidgood, W. D., Jr, Horii, S. C., Prior, F. W., & Van Syckle, D. E. (1997). Understanding and using DICOM, the data interchange standard for biomedical imaging. Journal of the American Medical Informatics Association: JAMIA, 4(3), 199–212. https://doi.org/10.1136/jamia.1997.0040199

[3] Benuwa, Ben & Zhan, Yong & Ghansah, Benjamin & wornyo, Dickson & Banaseka, Frank. (2016). A Review of Deep Machine Learning. International Journal of Engineering Research in Africa. 24. 124-136. 10.4028/www.scientific.net/JERA.24.124.

[4] Ethem Alpaydin. 2010. Introduction to Machine Learning (2nd. ed.). The MIT Press.

[5] Daniel G.G. (2013) Artificial Neural Network. In: Runehov A.L.C., Oviedo L. (eds) Encyclopedia of Sciences and Religions. Springer, Dordrecht. https://doi.org/10.1007/978-1-4020-8265-8_200980

[6] Albawi, Saad & Abed Mohammed, Tareq & ALZAWI, Saad. (2017). Understanding of a Convolutional Neural Network. 10.1109/ICEngTechnol.2017.8308186.

[7] O'Shea, Keiron & Nash, Ryan. (2015). An Introduction to Convolutional Neural Networks. ArXiv e-prints.

[8] Convolutional neural networks Jianxin Wu, LAMDA Group, National Key Lab for Novel Software Technology, Nanjing University, China, February 11, 2020

[9] Simonyan, Karen & Zisserman, Andrew. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv 1409.1556.

[10] Transfer learning with a pretrained ConvNet by Tensorflow from https://www.tensorflow.org/tutorials/images/transfer_learning#freeze_the_convolutional_base

[11] Precision vs Recall by Shruti Saxena May 12, 2018 from https://towardsdatascience.com/precision-vs-recall-386cf9f89488

[12] Lu, Lu & Shin, Yeonjong & Su, Yanhui & Karniadakis, George. (2019). Dying ReLU and Initialization: Theory and Numerical Examples.

[13] B. Hanin. Which neural net architectures give rise to exploding and vanishing gradients? In Advances in Neural Information Processing Systems, pages 580–589, 2018.

[14] A. F. Agarap. Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375, 2018.

[15] Deep Learning Essentials: Your hands-on guide to the fundamentals of deep learning and neural network modelling by Anurag Bhardwaj, Wei Di, Jianing Wei, Packt Publishing Ltd, 2018, ISBN 1785887777, 9781785887772

[16] Clevert, Djork-Arné & Unterthiner, Thomas & Hochreiter, Sepp. (2015). Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). Under Review of ICLR2016 (1997).

[17] Nwankpa, Chigozie & Ijomah, Winifred & Gachagan, Anthony & Marshall, Stephen. (2018). Activation Functions: Comparison of trends in Practice and Research for Deep Learning.

[18] Pedamonti, Dabal. (2018). Comparison of non-linear activation functions for deep neural networks on MNIST classification task.

[19] Ting K.M. (2017) Confusion Matrix. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning and Data Mining. Springer, Boston, MA

[20] Ting K.M. (2011) Precision and Recall. In: Sammut C., Webb G.I. (eds) Encyclopedia of Machine Learning. Springer, Boston, MA

[21] Goutte, C., & Gaussier, É. (2005). A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. ECIR.

[22] Janocha, K., & Czarnecki, W. (2017). On Loss Functions for Deep Neural Networks in Classification. ArXiv, abs/1702.05659.

[23] Kingma, Diederik & Ba, Jimmy. (2014). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations.

[24] Hochreiter, Sepp. (1998). The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems. 6. 107-116. 10.1142/S0218488598000094.

[25] How to pick the best learning rate for your machine learning project by David Mack, 16 April 2018 from https://www.freecodecamp.org/news/how-to-pick-the-best-learning-rate-for-your-machine-learning-project-9c28865039a8/

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

[27] A Review of Popular Deep Learning Architectures: ResNet, InceptionV3, and SqueezeNet by Vihar Kurama, 5 Jun 2020 from https://blog.paperspace.com/popular-deep-learning-architectures-resnet-inceptionv3-squeezenet/

[28] Howard, A. et al. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications." ArXiv abs/1704.04861 (2017): n. pag.

[29] Review: MobileNetV1 — Depthwise Separable Convolution (Light Weight Model) by Sik-Ho Tsang, Oct 14, 2018 from https://towardsdatascience.com/review-mobilenetv1-depthwise-separable-convolution-light-weight-model-a382df364b69

[30] N. Reddy, A. Rattani and R. Derakhshani, "Comparison of Deep Learning Models for Biometric-based Mobile User Authentication," 2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS), Redondo Beach, CA, USA, 2018, pp. 1-6, doi: 10.1109/BTAS.2018.8698586.

[31] Madani, A., Arnaout, R., Mofrad, M. *et al.* Fast and accurate view classification of echocardiograms using deep learning. *npj Digital Med* **1,** 6 (2018). https://doi.org/10.1038/s41746-017-0013-1

[32] Zhang J, Gajjala S, Agrawal P, Tison GH, Hallock LA, Beussink-Nelson L, Lassen MH, Fan E, Aras MA, Jordan C, Fleischmann KE, Melisko M, Qasim A, Shah SJ, Bajcsy R, Deo RC. Fully Automated Echocardiogram Interpretation in Clinical Practice. Circulation. 2018 Oct 16;138(16):1623-1635. doi: 10.1161/CIRCULATIONAHA.118.034338. PMID: 30354459; PMCID: PMC6200386.

[33] Apostolopoulos, I.D., Mpesiana, T.A. Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. Phys Eng Sci Med 43, 635–640 (2020). https://doi.org/10.1007/s13246-020-00865-4.

[34] Wang, Cheng & Chen, Delei & Lin, Hao & Liu, B. & Zeng, C. & Chen, D. & Zhang, E.. (2019). Pulmonary Image Classification Based on Inception-v3 Transfer Learning Model. IEEE Access. PP. 1-1. 10.1109/ACCESS.2019.2946000.

[35] Cardiovascular Diseases By World Health Organization from https://www.who.int/health-topics/cardiovascular-diseases/

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

Lee Qi Zian is pursuing his Bachelor of Computer Science, majoring in Artificial Intelligence at Universiti Teknikal Malaysia Melaka. His research interest includes Data Science and AI applications.

Yogan Jaya Kumar is a Senior Lecture in Universiti Teknikal Malaysia Melaka. He earned his bachelor degree and master degree from Universiti Sains Malaysia. He completed his PhD in 2014 in the field of Computer Science. His research involves in the field of Text Mining, Information Extraction and AI applications.

Goh Ong Sing is Professor at Universiti Teknikal Malaysia Melaka. His main research interest is in the development of intelligent agent, machine learning and speech technology, conversational robot and mobile services. He has led research grants funded by Malaysian Government's Intensified Research.

Zi Ye received bachelor degree in Mathematics & Statistical Science from University College London, UK in 2009, Master's degree in Applied Statistics from University of Oxford, UK in 2010. She is now pursuing her PhD in Universiti Teknikal Malaysia Melaka. Her research interests involve Artificial Intelligence & Machine Learning.