Journal of Electronics and Electrical Engineering
https://ojs.wiserpub.com/index.php/JEEE/

Research Article

# EchoTrace: A 2D Echocardiography Deep Learning Approach for Left Ventricular Ejection Fraction Prediction

**Anup Kumar Paul*, Yakub Sadlil Bhuiyan**

Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh
E-mail: anuppaul@ewubd.edu

**Received:** 26 October 2023; **Revised:** 14 December 2023; **Accepted:** 22 December 2023

**Abstract:** A key indicator in the diagnosis, prognosis, and management of individuals with heart failure (HF) is the left ventricular ejection fraction (EF). The amount of blood that is forced out of the left ventricle with each contraction provides important details on how well the heart can circulate oxygen-rich blood across the human body. Echocardiography has long been the most commonly used imaging method for determining LVEF due to its availability and cost-effectiveness. This paper makes use of the EchoNet-Dynamic dataset, which has left ventricle coordination data. An organized data preprocessing pipeline is created to extract frames along with coordinates. The suggested model architecture incorporates pre-trained transfer learning models that are optimal for the task of localizing the left ventricle boundaries. By predicting coordinates with CNN regression-type models, we showed how a novel volume tracing method could be used to localize the left ventricle boundary and perhaps mitigate the drawbacks of segmentation-based methods. Based on predetermined thresholds, we divided the ejection fraction (EF) values into "normal", "mild", and "abnormal" categories to detect the patient's heart condition. The analysis revealed a high degree of sensitivity for the "normal" and "abnormal" classes but was lower in the "mild" class. We obtained a confusion matrix accuracy of 77%.

***Keywords***: ejection fraction, transfer learning, computer vision, deep learning, CNN

## 1. Introduction

An essential parameter in the diagnosis, treatment, and prognosis of individuals with heart failure (HF) is the left ventricular ejection fraction (EF). It contrasts the volume of blood pumped out with the total volume of blood in the heart's left ventricle. Based on their EF measurements, individuals with HF have historically been divided into two phenotypes: heart failure with preserved ejection fraction, which has an EF value equal to or greater than 50%, and heart failure with reduced ejection fraction, which has an EF value less than 50%. Heart failure with mid-range ejection fraction, which has an EF between 40 and 49% according to US standards and between 41 and 49% according to European guidelines, is a third borderline category that has been the subject of research in recent years [1, 2]. Recent research on patients with heart failure with mid-range ejection fraction revealed some significant differences between heart failure with reduced ejection fraction and heart failure with preserved ejection fraction patients [3]. Distinct pathologies have been identified for each of the three phenotypes [4, 5, 6]. In our paper, we categorized these three phenotypes as "Normal" when EF is greater than 50%, "Mild" from 40 to 49% and "Abnormal" when less than 40%.


DOI: https://doi.org/10.37256/jeee.3120243824

In terms of death and disability, cardiovascular diseases still dominate the world [7]. A precise assessment of cardiac function is essential for the diagnosis and treatment of various illnesses. Physicians may now obtain exact images of the heart because of recent advancements in medical imaging techniques, which have significantly increased their knowledge of the organ's structure and function. Among these imaging techniques, echocardiography stands out as a common and simple way to assess heart health [8]. Cardiovascular magnetic resonance imaging (CMR) has emerged as the standard for calculating left ventricular ejection fraction at the moment. But a cardiac MRI scan may cost us up to $2500, which is much higher than echocardiography [8]. However, manual inspection of the echocardiographic images is time-consuming, labor-intensive, and frequently sensitive to inter and intra-observer variability [9, 10, 11]. Automated echocardiographic image processing has shown encouraging results in terms of making it easier to diagnose and assess cardiac problems [12, 13].

Convolutional Neural Networks (CNN), a subset of deep learning algorithms, have achieved outstanding results in a range of computer vision applications, including segmentation, object identification, and picture categorization [14, 15, 16]. Transfer learning, which makes use of CNNs to their maximum potential, has proven to be a successful strategy for dealing with the problem of sparse training data for medical imaging applications. The main goal of this work was to create a CNN regression transfer learning model for the localization of the left ventricle (LV) border in cardiac images generated by 2D echocardiography. A significant stage in cardiac analysis is the localization of the LV borders, which provides vital assessments of cardiac function, including the ejection fraction (EF), a key marker of heart health. The efficiency and precision of echocardiographic analysis can be considerably improved by accurate and automated localization of LV borders, allowing physicians to make wise judgments about patient management and treatment plans. However, due to their poor accuracy in the presence of rhythm and structural alterations, such as left ventricular (LV) hypertrophy, dilatation, regional wall motion abnormalities, or unusual cardiac cycles, deep learning (DL) approaches for clinical application are difficult [17]. To accomplish this goal, we analyzed and assessed a variety of transfer learning models in order to determine which architecture performs the best in terms of sensitivity and validation loss.

Transfer learning is the process of using pretrained CNN models that have been optimized for our specific application and have been trained on ImageNet as initial weights [18]. With this approach, we are able to take advantage of the abundance of data that these models have gathered from performing a variety of visual tasks while also modifying them for the purpose of LV border localization. VGG16 and EfficientNetB7 demonstrated greater performance in localizing LV borders among the transfer learning models tested. Sensitivity, which indicates the percentage of properly localized LV borders, and validation loss, which measures the difference between predicted and actual boundary positions, are two metrics used to assess a model's performance.

The construction of an automatic and precise LV border localization model utilizing transfer learning and the thorough evaluation of several cutting-edge CNN architectures are the main achievements of this paper. The results of this study have important repercussions for improving the accuracy and efficiency of cardiac analysis and assisting physicians in making prompt and well-informed decisions for patient treatment.

The paper is structured as follows: A literature review is conducted in Section 2, followed by a thorough description of the methods used in Section 3. The results and discussion are presented in Section 4, and concluded in Section 5.

## 2. Literature review

With the advent of deep learning methods for medical image processing, the area of automated left ventricular ejection fraction (LVEF) has made great strides lately. In this review of the literature, we discuss the present status of research on automated LVEF and point out any gaps or restrictions that still need to be filled.

A deep learning-based technique for automatic endocardial boundary recognition and left ventricular functional evaluation in 2D echocardiographic videos was introduced by Ono et al. [19]. The effectiveness of various segmentation techniques was assessed through comparison, utilizing measures like mean Dice score and estimation error for echocardiographic indices. The global longitudinal strain (GLS), global circumferential strain (GCS), and left ventricular ejection fraction (LVEF) are segmented and estimated with the best degree of accuracy using the Unet++ model. The study

recognized the small number of test results from healthy patients and volunteers. Larger datasets and cross-validation should be used in future studies to evaluate the clinical utility of the suggested strategy and reduce bias. Only information gathered by experts utilizing certain ultrasound equipment was taken into account in the study.

In order to estimate the LVEF, Lagopoulos A. et al. devised a unique method that used geometric information taken from the left ventricular outlines [20]. They presented a thorough framework that includes contour segmentation, feature extraction, and LVEF estimation processes. An automated technique was used for contour segmentation to precisely identify the left ventricle in medical photographs. There is a lack of specific information in the publication regarding the size and variety of the validation dataset, which raises questions about how well the suggested strategy can be applied to other patient groups and imaging settings.

Research on the application of deep learning techniques for the segmentation of cardiac structures in 2D echocardiographic pictures was given by Leclerc S. et al. [13]. The authors investigate the extent to which the latest encoder-decoder deep convolutional neural network techniques can evaluate 2D echocardiographic images, specifically in terms of calculating clinical indices and segmenting heart structures, using a dataset. In order to facilitate echocardiographic evaluation, they unveiled the biggest completely annotated dataset for multi-structure ultrasound segmentation, known as the cardiac acquisitions dataset. The dataset includes 500 patients' two- and four-chamber acquisitions, together with reference measurements from three cardiologists on a fold of 50 patients and from one cardiologist on the entire dataset. With a mean correlation of 0.95 and an absolute mean error of 9.5 mL, the results demonstrated that encoder-decoder-based architectures beat the most advanced non-deep learning techniques and accurately recreated the expert analysis for the end-diastolic and end-systolic left ventricular volumes. The results are more contrasting, with a mean correlation coefficient of 0.80 and an absolute mean inaccuracy of 5.6% regarding the left ventricle's ejection percent. These results were marginally poorer than the intra-observer results, despite being below the inter-observer scores. Though deep learning techniques have produced encouraging results in several medical imaging applications, their use in echocardiographic picture segmentation is currently only sporadic. Only a few studies have used convolutional neural networks (CNNs) for this purpose. A possible research need might be filled by investigating and comparing various deep learning architectures and methods designed particularly for echocardiographic picture segmentation.

In order to accurately segment the LV in real time, Smistad E. et al. suggested a technique that made use of a 3D convolutional neural network (CNN) architecture, especially a modified U-Net model [21]. A sizable collection of cardiac ultrasound pictures and their related ground truth annotations is used to train the network. The network's parameters were optimized during training to reduce the discrepancy between the predicted LV segmentation and the actual segmentation. The main goal of the study is to assess the suggested strategy using a particular dataset. The technique might be tested on other datasets with changes in imaging protocols, picture quality, and patient groups in order to further evaluate its generalizability. The article shows that the deep learning-based methodology outperforms manual segmentation techniques, although a more thorough performance comparison may be made. Assessing the relative merits and drawbacks of the suggested methodology would be made easier by contrasting it with other cutting-edge deep learning techniques or other segmentation methods.

Moal O. et al. method for segmenting the left ventricle, locating the mitral valve, and estimating EF using the modified Simpson's rule made use of deep learning and statistical shape modelling [22]. On internal and external datasets, the technique achieves state-of-the-art performance with mean absolute errors of 6.10% and 5.39%, respectively. Additionally, it improves interpretability by giving doctors clear information at every stage of the examination. The method presents a potentially viable alternative to the laborious and unpredictable manual evaluation of EF in cardiac ultrasonography. Although the study lacks a direct comparison with manual evaluations to examine the accuracy and reliability of the completely automatic technique presented in the publication for left ventricular EF assessment, the study makes no mention of any drawbacks or difficulties that could arise when applying the suggested approach in actual clinical settings, such as changes in ultrasound technology, imaging techniques, or patient groups.

Asch F. et al. demonstrated that the automated calculation of LVEF was possible and had high consistency and excellent agreement with the reference values [23]. The algorithm showed sensitivity and specificity for identifying EF<=35% of 0.90 and 0.92, respectively, which was comparable to clinical readers' readings. To determine if the algorithm is a legitimate alternative to traditional measures, its performance should be compared to that of other imaging modalities,

such as cardiac magnetic resonance. In order to quantify LVEF without first determining the borders of the heart and measuring the volumes of the ventricles at the end of the diastole and end of the systole, the author evaluated the viability and accuracy of a novel totally automated machine learning approach. 99 subjects were tested, covering a broad spectrum of EF and image quality in relation to body habitus. They discovered that every patient in the test group could implement the novel technique, and the automated estimations proved to be quite accurate when compared to the reference standard of traditional measures made by a panel of experts. Crucially, the accuracy matched that of the traditional analysis performed by impartial clinical readers. Furthermore, when the automated analysis was performed on several pairs of apical 2- and 4-chamber views, the results were quite consistent. The authors also showed how, by virtually eliminating inter-technique bias and lowering limits of agreement, a straightforward mathematical de-trending correction based on parameters derived from conventional measurements might significantly improve the accuracy and consistency of the automated analysis.

Our work, in contrast to the articles mentioned, focuses on a unique method for localizing the left ventricle border in automated left ventricular ejection fraction (LVEF) calculation utilizing CNN regression-type models. We provide an alternate approach that has demonstrated potential benefits in some circumstances. By directly predicting the coordinates of the left ventricle border, our method seeks to increase accuracy and get around problems with segmentation-based approaches. Initial performance findings are encouraging, but more testing and comparison to other approaches are required to determine its overall efficacy.

A summary of various related works including the used machine learning or deep learning models along with their pros and cons are depicted in Table 1.

**Table 1.** Summary of various related works.

| Literature | Used Model | Pros | Cons |
|---|---|---|---|
| Ono et al. [19] | Unet++ | • Has the ability to facilitate examiners and enhance echocardiography process. | • Only information gathered by experts utilizing certain ultrasound equipment was taken into account.<br>• A small sample of test results from sick and healthy participants. |
| Lagopoulos A. et al. [20] | Machine Learning based Gradient Boosted Tree | • Simpler method than the state of the art.<br>• More explainable and competitive in terms of accuracy. | • Lack of size and variety of the validation dataset.<br>• Limited to other patient groups and imaging settings. |
| Leclerc S. et al. [13] | Encoder Decoder Deep CNN | • Achieves mean correlation of 0.95 and an absolute mean error of 9.5 mL. | • Results were marginally poorer than the intra-observer results. |
| Smistad E. et al. [21] | Modified Unet | • Highly efficient (17 ms on laptop GPU)<br>• Measures the left ventricle volume and ejection fraction in real time across several heartbeats in complete automation. | • Lacks generalizability.<br>• Lacks thorough performance comparison. |
| Moal O. et al. [22] | Deep Learning Model | • Achieves mean absolute error of 6.10% and 5.39% on internal and external datasets respectively.<br>• Improves interpretability by giving doctors clear information at every stage of the examination. | • Lacks a direct comparison with manual evaluations.<br>• Mention no drawbacks when applying the suggested approach in actual clinical settings. |
| Asch F. et al. [23] | Machine Learning Algorithms | • High consistency and with the reference values.<br>• Fast and fully automated nature. | • Tiny test group (only 99 patients).<br>• Validating the correctness of the automated EF estimates in the absence of endocardial boundaries is difficult.<br>• De-trending correlation, of the 99 test group patients. |

## 2.1 *Transfer learning*

The utilization of previously trained models and their expertise on new challenges is made possible by the potent deep learning approach known as transfer learning. By utilizing the recognized characteristics and patterns from previous tasks, it enables effective training of deep neural networks even with smaller input datasets. This method has important ramifications for data science since labeled data is frequently in short supply [24]. The basic concept of transfer learning is using the information learned from a model that has already been trained to enhance predictions on new, related tasks. A model can improve its comprehension and prediction abilities for the new challenge by making use of the insights gained during training. For instance, a classifier that has been taught to detect backpacks in photos might use that information to classify other items, such as sunglasses [25].

Figure 1 shows that transfer learning's main objective is to move previously acquired skills and information from one activity to another, making learning quicker and more precise. A pre-trained model's weights and parameters must be adjusted to make it more suitable. This method is particularly useful for jobs requiring significant computational capacity, such as computer vision. Saving time throughout the training process is one benefit that transfer learning delivers. Data scientists can use pre-trained models that have previously discovered common traits and trends as opposed to starting from scratch. Even with a small amount of labeled data, this not only quickens the training process but also enhances the model's performance on the new tasks. The weights that have already been trained are fixed, while the additional layers are fine-tuned during the training process. This aids in maintaining the broad information gained from the pre-trained model while allowing the model to adapt to the target job. The performance on the new dataset may be greatly enhanced by fine-tuning the model.

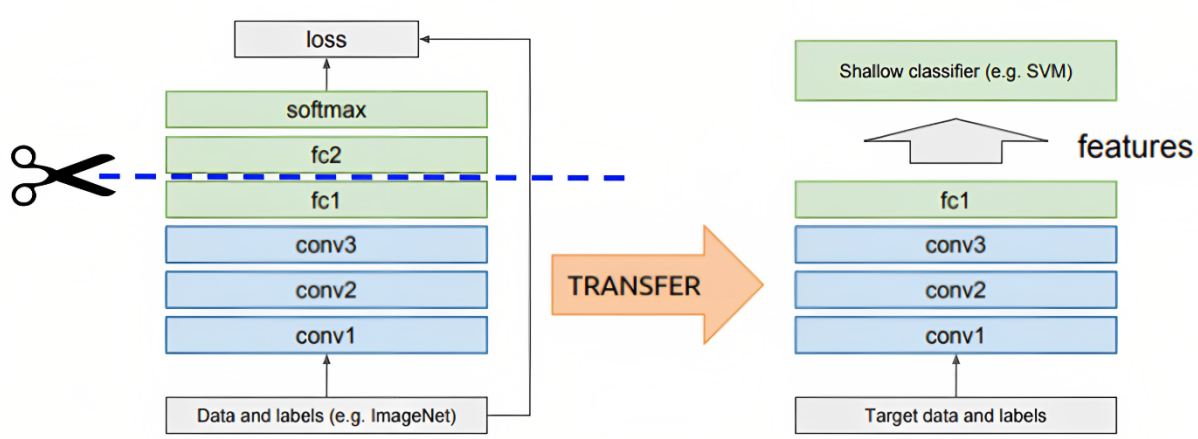


**Figure 1.** Transfer learning with pre-trained model as feature extractor.

## 2.2 *VGG*

Visual Geometry Group 16, or VGG16, is a deep convolutional neural network architecture that has become well known and well-liked in the field of computer vision. VGG16, created by the University of Oxford's (O Vision Geometric Group, is renowned for its elegance, simplicity, and superior performance in a range of image categorization tasks [26].

Figure 2 shows that the depth that defines the VGG16 design, which has 16 layers overall, is what makes it unique. Thirteen of these layers are convolutional, and the final three are fully connected. The fully connected layers carry out the final classification based on these retrieved characteristics after the convolutional layers have extracted significant information from the input image [27].

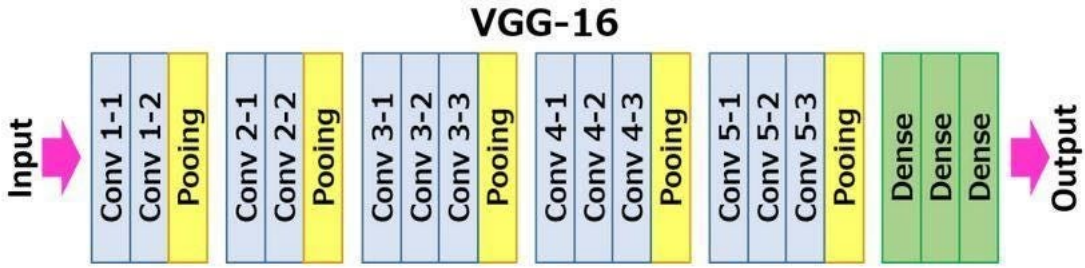


**Figure 2.** VGG-16 architecture

Its homogeneous structure makes VGG16 a very fascinating subject for research. The modest $3\times3$ receptive field size of each convolutional layer in the network corresponds to the size of the convolutional filter used to transform the input image. Using a stride of $2\times2$, a maximum pooling layer is used after each convolutional layer. Convolution and pooling layers are consistently repeated, resulting in a deep architecture that catches increasingly complex patterns and structures in the input images.

The early layers of VGG16 utilize narrow receptive fields to enable the network to gather local characteristics and fine-grained information. The receptive field grows as the network gets deeper, allowing for the recognition of larger and more abstract information. By gradually extracting hierarchical information from low-level edges and textures to high-level object ideas, this hierarchical learning process aids VGG16 in developing a deeper comprehension of complex visual concepts and things.

Because of its straightforward architecture, VGG16 is very easy to understand and analyze. The network's regular structure and uniform filter sizes make it easier to analyze the network's behavior and interpret its learned representations. VGG16 is frequently used by researchers as a starting point for creating more complex architectures or as a baseline model for a variety of computer vision tasks.

With regard to benchmark datasets like the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) dataset, VGG16 has repeatedly exhibited remarkable performance in image categorization tasks and attained state-of-the-art accuracy. The network's proficiency in learning discriminative features at various scales and levels of abstraction is a factor in how well it does image categorization. To discover complicated patterns and qualities in photographs, like a visual maestro, VGG16 gradually learns representations by studying the data at several layers.

It has a few constraints despite its outstanding performance. Since there are many characteristics as a result of the network's depth and homogeneity, it is memory-intensive. It can take some time to complete the training and inference procedures, especially on devices with limited resources. Furthermore, VGG16's homogeneous structure restricts its ability to simulate dependencies over long periods or capture geographical context.

However, it is impossible to emphasize how much VGG16 has contributed to the fields of deep learning and computer vision. Many technological advancements have been influenced by its beauty and simplicity. Researchers have experimented with alternative layer levels, filter sizes, or bypass connections in the VGG design, further testing the limits of performance and efficiency.

## 2.3 *MobileNetV2*

Google's MobileNetV2 is a cutting-edge deep learning architecture designed to provide powerful and adaptive models for mobile and embedded devices [28]. Machine learning models that can perform successfully in scenarios with low funding are becoming increasingly popular as mobile phones, tablets, and other forms of embedded devices increase. To address this issue, MobileNetV2 attempts to strike a careful balance between model size, computational efficiency, and accuracy.

As the successor to MobileNetV1, MobileNetV2 makes a number of significant improvements to its functionality and adaptability. MobileNetV2's usage of an inverted residue architecture is one of its unique features. Figure 3 shows that this

structure is made up of two types of blocks: blocks for residuals with a stride of one and blocks for reduction with a stride of two. Each block's three primary components are a $1 \times 1$ convolutional layer, a depth-wise separable convolutional layer, and another $1 \times 1$ convolutional layer [28].

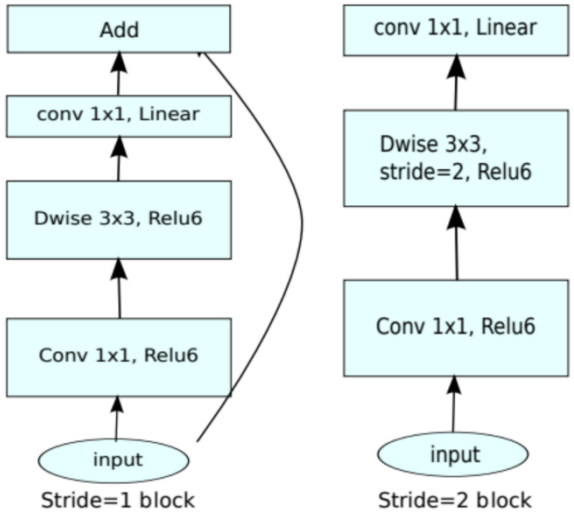


**Figure 3.** MobileNet-V2 architecture.

In MobileNetV2, the inverted residual structure is essential for lowering computational complexity while preserving model efficacy. To do this, depth-wise detachable convolution, which is a fundamental building component, is used. Convolutional layers often carry out a process called convolution on the full input volume, which can be computationally expensive. In contrast, the spatial and channel-wise convolutions are split into two distinct layers by depth-wise separable convolutions. A series of intermediate feature maps is produced after applying a depth-wise convolution separately to each input channel at the beginning. The final result is then created by combining the intermediate feature maps using a $1 \times 1$ pointwise convolution. MobileNetV2 greatly reduces the number of parameters and the amount of computing necessary by decoupling the spatial and channel-wise operations, resulting in a more lightweight system.

Down sampling the spatial dimensions of the feature maps is done via reduction blocks with a stride of 2. By removing less significant data, this down sampling technique aids in preserving and capturing the most crucial aspects. By lowering the spatial scale, MobileNetV2 can maintain a manageable model size and computational efficiency while still attaining acceptable accuracy across a range of workloads. Additionally, the reduction blocks enable effective information flow through the network by gradually decreasing the spatial dimensions, enabling the model to capture hierarchical representations of the input data.

Eliminating non-linearities from the thin layers is another noteworthy advancement in MobileNetV2. After each depth-wise separable convolution, non-linearities like ReLU activation functions were applied in the original MobileNet architecture. To improve information flow and the network's capacity for representation, these non-linearities are eliminated from the thin layers in MobileNetV2. With this modification, MobileNetV2 is able to make greater use of the computational resources, especially in situations where the devices' processing power is constrained.

MobileNetV2 reduces the number of parameters and the size of the overall model by excluding non-linearities in the thin layers. MobileNetV2 is more suited for implementation on embedded and mobile devices, where memory and processing power are frequently constrained, thanks to its reduced complexity. MobileNetV2's efficiency and compactness make it simple to integrate the network into real-time processing or offline operation applications. It makes it possible to deploy deep learning models on hardware, which lowers the frequency of network interactions and improves user privacy.

By using a combination of depth-wise separable convolutions, inverted residual structures, and tactical down sampling, MobileNetV2 successfully balances accuracy and efficiency. By using these methods, the model may efficiently capture and represent the salient aspects of the input data while incurring the least amount of computing overhead. Multiple computer vision tasks, including picture classification, object detection, and semantic segmentation, have shown the efficiency of MobileNetV2.

MobileNetV2 has a considerable influence on computer vision research. It has made it possible to create intelligent applications for embedded and mobile devices, creating new opportunities for real-time visual understanding and analysis. Numerous fields, such as driverless vehicles, security systems, augmented reality, and mobile healthcare, have discovered uses for MobileNetV2. It is the ideal option for developers and academics working on edge computing solutions due to its capacity to give accurate forecasts while operating effectively on platforms with limited resources.

## 2.4 *EfficientNet-B7*

The outstanding deep learning architecture EfficientNet-B7 shown in Figure 4 has attracted a lot of interest and praise in the field of computer vision. The EfficientNet-B7, created by the Google Brain team, is the apex of the EfficientNet series and offers unmatched performance and efficiency [29]. EfficientNet-B7 is now the preferred option for both researchers and practitioners, thanks to its cutting-edge design principles and avant-garde scaling strategies.

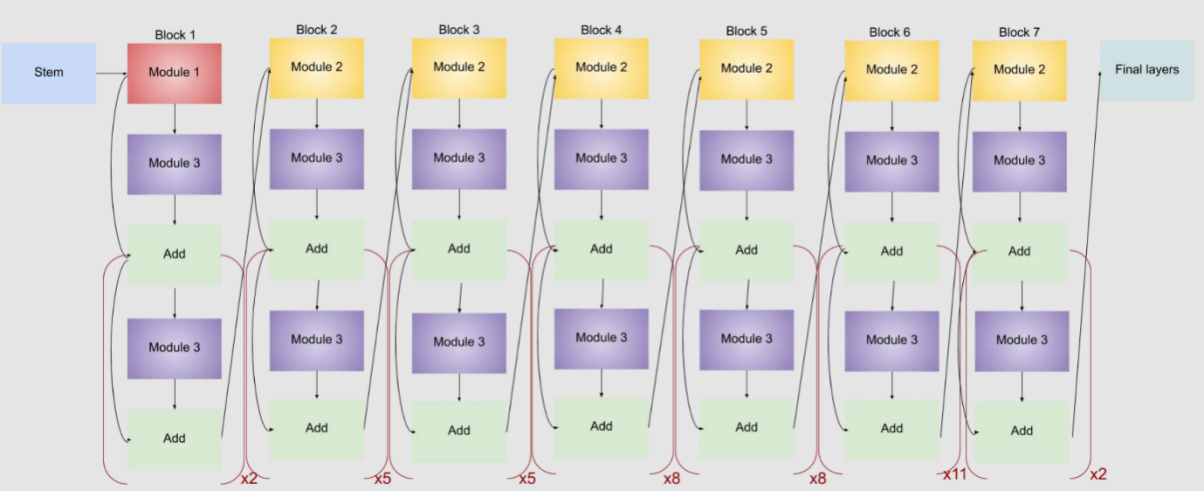


**Figure 4.** EfficientNet-B7 architecture.

EfficientNet-B7's success is due to its original model scaling strategy. EfficientNet-B7 uses a compound scaling strategy as opposed to conventional methods, which concentrate on growing just one component of the model, like depth or breadth. By ensuring that the model's depth, width, and resolution are all evenly scaled, this method creates an architecture that is more harmonious and efficient.

The dimensions of the model are heavily influenced by the compound scaling factor employed in EfficientNet-B7. The model achieves a harmonious balance between computing efficiency and accuracy by carefully choosing optimal scaling factors for depth, width, and resolution. EfficientNet-B7's bigger scaling factors produce a larger, more potent model that can capture complex patterns and characteristics in the input data.

The network's layer count is determined by the depth scaling coefficient of EfficientNet-B7. It can recognize both low-level and high-level features by capturing both low-level and high-level features by deepening the model and learning hierarchical representations of the input data. The model's capacity to comprehend intricate visual patterns and deliver precise forecasts is aided by this depth scaling. In EfficientNet-B7, the channel size of each layer is determined by the

width scaling coefficient. More information can be gathered and analyzed in each layer by widening the model. This larger network architecture enables it to gather more contextual data, improving feature learning and speed.

The input image resolution in EfficientNet-B7 is influenced by the resolution scaling coefficient. The model can distinguish between distinct classes better thanks to higher-resolution inputs that allow it to grasp the nuances and finer details in the data. It can extract more detailed and nuanced representations from higher-resolution images, which improves its performance on a variety of computer vision tasks.

EfficientNet-B7 uses a number of additional strategies to enhance its performance. Utilizing effective bottleneck structures is one of these methods. The model's parameters and level of computational complexity are reduced by these bottlenecks, which are $1 \times 1$ convolutions. By utilizing these bottlenecks, EfficientNet-B7 achieves a reasonable trade-off between model size and performance, making it extremely resource-efficient [30].

Utilizing massive datasets like ImageNet, which has millions of tagged images, is a key component of training EfficientNet-B7. It learns rich and discriminative representations that are applicable to a variety of tasks and datasets by training on such a wide range of diverse and substantial datasets. The model is able to develop a deep grasp of the visual environment thanks to this pre-training process, which lays the groundwork for later task-specific fine-tuning.

## 2.5 *ResNet-50*

ResNet-50 (Residual Network-50) is a notable achievement in deep learning and convolutional neural networks (CNNs). This architecture, created by Microsoft researchers, has transformed the way we construct and train deep neural networks.

The fundamental obstacle for deep neural networks is the degradation problem, in which the network's accuracy decreases as its depth grows. The vanishing gradient problem causes the gradients to become exceedingly small during backpropagation, making it impossible for the network to learn successfully. ResNet-50 tackles this issue by introducing skip connections, also known as residual connections or shortcuts, which allow the network to bypass certain levels and enable direct information flow [31].

The core principle behind ResNet-50 is to learn residual mappings by focusing on the difference between the desired output and the existing representation. This is accomplished by incorporating residual blocks, which consist of multiple convolutional layers, batch normalization, and rectified linear unit (ReLU) activations shown in Figure 5. These blocks allow the network to learn residual functions, allowing it to approximate the desired mapping more accurately.

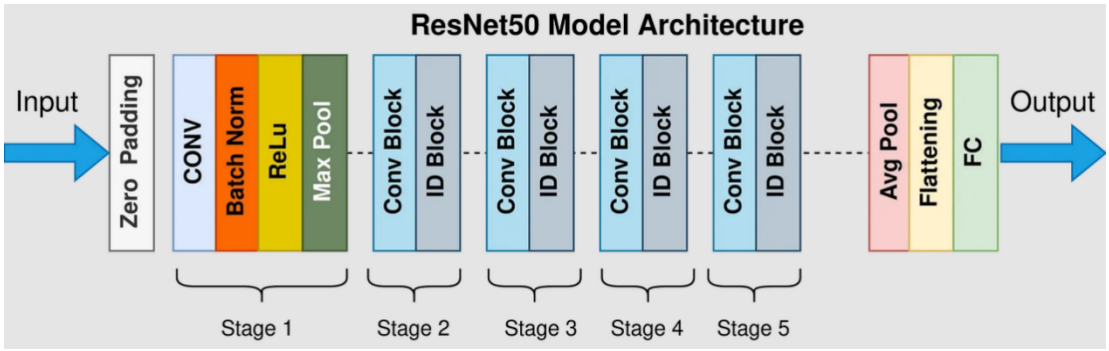


**Figure 5.** ResNet-50 architecture.

One of the key elements of ResNet-50 is the implementation of the bottleneck architecture. Each ResNet-50 residual block consists of three convolutional layers: a $1 \times 1$ layer, a $3 \times 3$ layer, and another $1 \times 1$ layer. The $1 \times 1$ convolutional layers are responsible for reducing and then restoring the dimensionality of the input feature maps, while the $3 \times 3$ convolutional layer is in charge of capturing spatial information. This bottleneck architecture reduces the computational cost of the network while maintaining its representational strength, allowing deeper networks to be trained more successfully.

ResNet-50's architecture is made up of 50 layers, hence the name. It starts with a convolutional layer and then moves on to a max-pooling layer. The layers that follow are separated into four stages, each of which contains many leftover blocks. As the number of filters at each level grows, the network is able to capture increasingly complicated information. After each convolutional layer, batch normalization is done to stabilize the training process and accelerate convergence.

It has demonstrated exceptional performance in a variety of computer vision tasks, including image identification issues. ResNet-50 outperformed humans in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2015, marking an important milestone in the field. Its outstanding performance can be attributed to its capacity to acquire highly discriminative features via skip connections, allowing the network to capture fine-grained information and patterns.

## 2.6 *Xception*

Xception, which stands for "Extreme Inception," is a convolutional neural network (CNN) architecture developed by François Chollet, the author of the Keras deep learning package. It is built on the Inception architecture, which is well known for making optimal use of computing resources and capturing multi-scale data [32]. By employing them as the main building blocks of the network, Xception takes the idea of depth-wise separable convolutions to an extreme.

The overall Architecture of Xception is shown in Figure 6. The fundamental idea behind Xception is to replace the conventional convolutional layers in classic CNNs with depth-wise separable convolutions. The depth-wise convolution and point-wise convolution processes make up a depth-wise separable convolution. Each input channel is convolved with a different filter in depth-wise convolution, producing a series of feature maps. The feature maps are then combined into the final output by using a $1 \times 1$ convolution in the point-wise convolution. In comparison to traditional convolutions, this separation of spatial and channel-wise convolutions greatly lowers the number of parameters and calculations. Xception delivers improved parameter efficiency and lowers network computational complexity by employing depth-wise separable convolutions. This is especially relevant in circumstances with low computing resources, such as mobile devices or embedded systems. Furthermore, Xception enables a finer-grained examination of the data, allowing the network to catch subtle patterns and features.

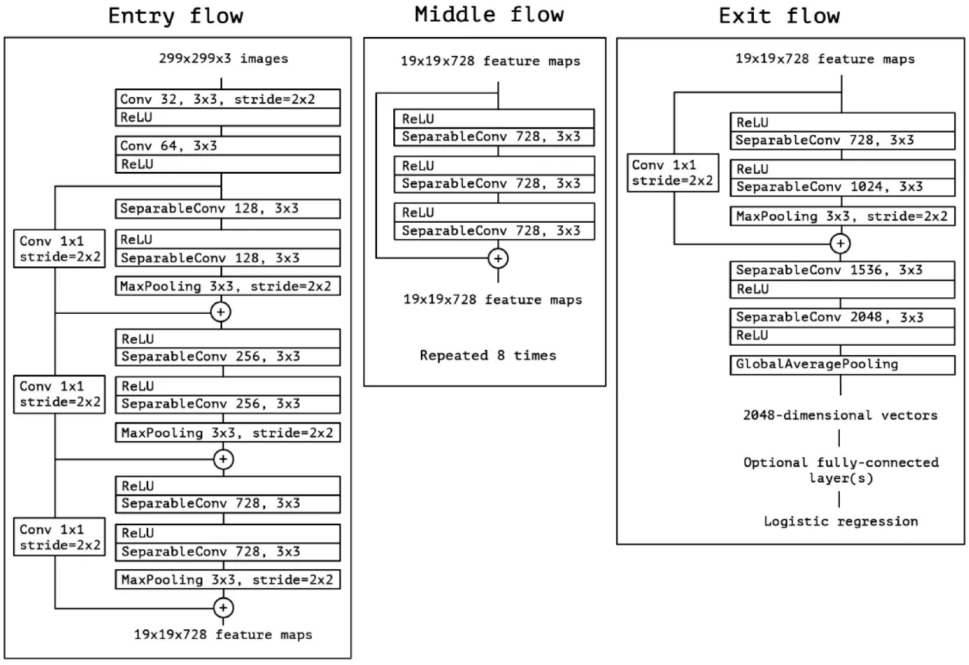


**Figure 6.** Xception architecture.

Xception's architecture is made up of a succession of depth-wise separable convolutional blocks. Each block is made up of three modules: a separable convolutional module, a linear projection module, and an identity module. The depth-wise

separable convolution is carried out by the separable convolutional module, which is followed by batch normalization and activation. By lowering the number of input channels, the linear projection module can match the number of output channels. By bypassing the convolutions and connecting the input and output directly, the identification module acts as a short-cut link that ensures minimum interference with information flow.

To address the vanishing gradient issue and improve data flow via the network, Xception also makes use of skip connections, which are similar to those found in ResNet design. The gradients can propagate more readily during training, which enhances the convergence of the training, thanks to the addition of these skip connections between some modules.

The use of global average pooling (GAP) rather than completely linked layers at the network's end is one distinguishing feature of Xception. In order to reduce the spatial dimensions to a single value per channel, GAP uses spatial averaging over the feature maps. By dramatically lowering the number of factors, this aggregation enables the network to collect global context. The final predictions are then made using the output feature vector from a SoftMax classifier.

With regard to a number of computer vision tasks, such as semantic segmentation, object identification, and picture classification, Xception has exhibited remarkably strong performance. It is a potent tool for research as well as practical applications due to its capacity to capture intricate features and intricate patterns, as well as its parameter efficiency. On large-scale datasets like ImageNet, Xception may be first trained using pre-learned weights, which offers a solid place to start for transfer learning. The necessity for lengthy training from scratch is minimized by fine-tuning domain-specific datasets that enable the network to adapt to certain tasks with little labeled input.

## 2.7 *Inception*

Christian Szegedy et al.'s 2014 introduction of the Inception architecture for deep learning and image categorization was a game-changer [33]. Since prior convolutional neural network (CNN) designs ran into problems with computing expense and overfitting as network depth increased, Inception was developed to find a happy medium between both of them.

The employment of inception modules is key to the Inception architecture as shown in Figure 7. The network can capture information at many scales and resolutions thanks to these modules' ability to run several convolutions with varying filter sizes in parallel. The network is taught to distinguish between items of varying sizes by combining filters of varying sizes ($1\times1$, $3\times3$, and $5\times5$) into a single module. The ultimate output of the module is a concatenation of the results of these convolutions. $1\times1$ convolutions, commonly referred to as "bottleneck layers," are used in Inception designs to further improve computational efficiency. Before using the computationally intensive bigger convolutions, these bottleneck layers are used to lower the number of input channels. The number of parameters and computational complexity can be drastically decreased while still preserving the network's capacity for representation by lowering the dimensionality of the input space.

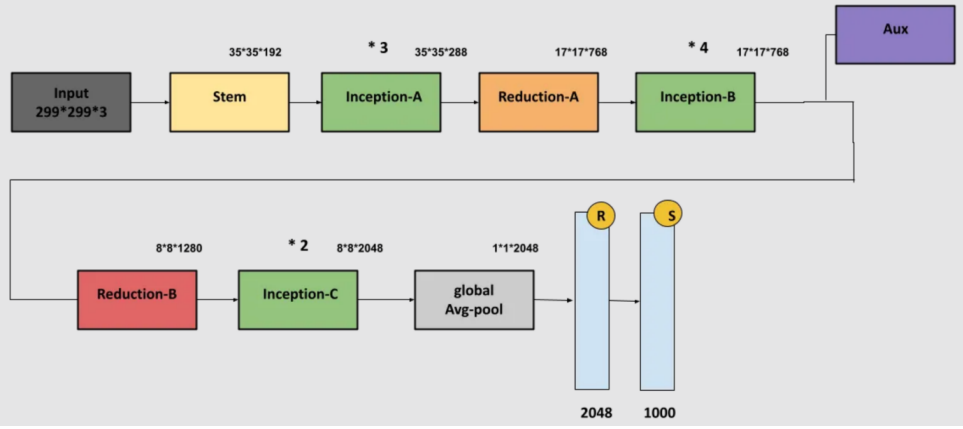


**Figure 7.** Inception architecture.

Additional upgrades were included in succeeding iterations of the Inception architecture, such as Inception-v3 and Inception-ResNet, in order to increase its performance. These iterations include Inception-v3. In Inception version 3, factorized 7×7 convolutions were included. These convolutions break a 7×7 convolution down into its component parts, a 1×7 convolution and a 7×1 convolution, respectively. The computational cost is decreased as a result of this factorization, while the efficiency of the convolution process is not affected in any way. In addition, the Inception modules and the residual connections from the ResNet architecture were brought together to form the Inception-ResNet combination. Incorporating residual connections helps ease the vanishing gradient problem and promotes smoother gradient flow during training, which ultimately leads to greater performance and quicker convergence. These benefits are achieved as a result of the inclusion of residual connections.

On a variety of picture classification benchmarks and contests, the Inception architecture has performed exceptionally well, achieving remarkable results. The Inception models routinely received top ranks in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC), demonstrating their capacity to handle complicated visual tasks and outperforming several earlier CNN architectures. This challenge was held by ImageNet.

In several image identification tasks, its capacity to collect both local and global characteristics has proved useful. The Inception architecture has continued to be a key milestone in the field of deep learning and computer vision, motivating other developments and pushing the limits of what is feasible in picture categorization and beyond, even with successive iterations and modifications.

## 2.8 *Comparison*

It's crucial to weigh the benefits and drawbacks of several deep learning networks like MobileNetV2, InceptionV3, Xception, ResNet50, VGG16, and EfficientNetB7 before settling on one. Figure 8 shows the comparison of performance based on ImageNet.

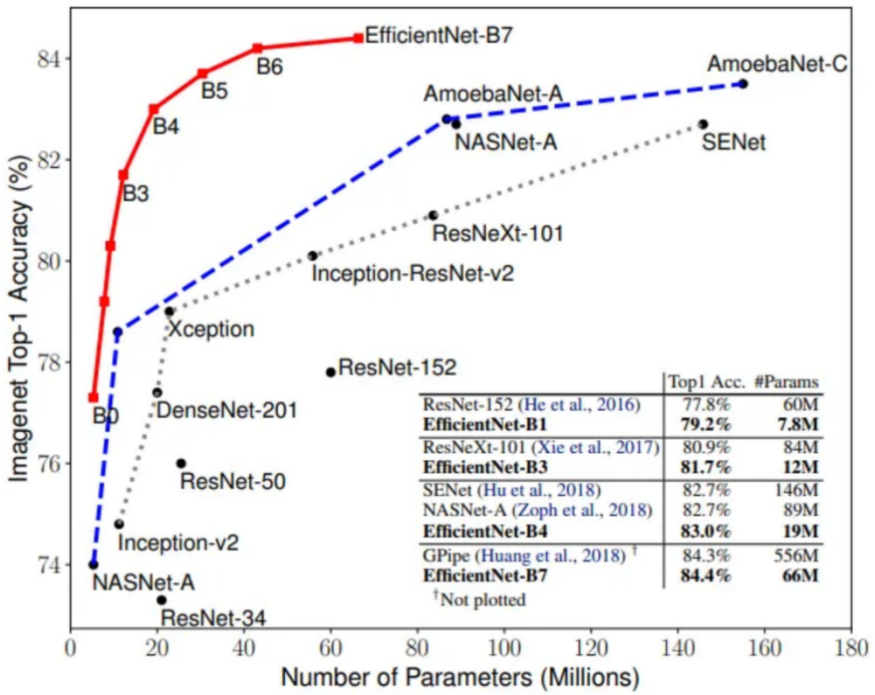


**Figure 8.** Model Size vs. Accuracy Comparison.

**MobileNetV2**

Pros:

- Efficient embedded and mobile device architecture.
- The complexity of calculation is reduced by inverted residual structure.
- Optimize resource consumption via the elimination of non-linearities from thin layers.

Cons:

- The model's accuracy might not be as high as that of other models.
- In situations where accuracy is critical, performance could be affected.

**InceptionV3**

Pros:

- Information is captured at various sizes by Inception modules, allowing for deeper feature extraction.
- Uses various filter sizes for better depiction.
- Uses $1 \times 1$ convolutions to reduce computational complexity.

Cons:

- For some applications, the depth and computing requirements might be difficult.
- A greater number of parameters in comparison to some other models.

**Xception**

Pros:

- Expands the capabilities of the Inception module by building on it.
- Efficiency is increased via depthwise separable convolutions.
- Produces competitive performance with less constraints.

Cons:

- Requires more computing power than some other models.
- A little bit more memory use.

**ResNet50**

Pros:

- Adds skip connections, allowing for the training of deeper networks.
- Deals with the vanishing gradient issue, improving convergence.
- Achieves cutting-edge outcomes in a variety of computer vision tasks.

Cons:

- Comparatively more parameters.
- Greater computational complexity in comparison to networks with fewer layers.

**VGG16**
Pros:

- A straightforward and understandable architecture.
- Performs astoundingly well in picture classification tasks.
- Absorbs hierarchical information gradually, enhancing comprehension of the visual environment.

Cons:

- Large number of parameters, making it computationally expensive.
- Memory-intensive, limiting deployment on resource-constrained devices.

**EfficientNetB7**
Pros:

- By using compound scaling, the model size, effectiveness, and accuracy are all balanced.
- Produces cutting-edge outcomes in picture classification challenges.
- offers a variety of models that are appropriate for various computational limitations.

Cons:

- The deployment of larger models may be constrained by the resources of certain devices.
- In comparison to certain other models, it may require greater computing power.

In conclusion, MobileNetV2 is perfect for low-powered devices since it trades off some precision for greater efficiency. Although InceptionV3 and Xception use more computing resources, they are superior at obtaining high-quality visual data. The state-of-the-art results that ResNet50 produces make it an excellent choice for many computer vision applications. While VGG16 is simple to use and achieves exceptional performance, doing so comes at the expense of greater computational complexity. Model size, efficiency, and accuracy are all considered by EfficientNetB7 while still accommodating a variety of computing requirements. Available resources, required precision, and the deployment platform all play a role in determining which architecture to choose. In order to choose the best architecture for their needs, researchers and practitioners must carefully weigh these benefits and drawbacks.

# 3. Materials and methods

## 3.1 *Dataset*

EchoNet-Dynamic is the name of the dataset that we utilized in the code [34]. It contains digital recordings of echocardiograms made at Stanford University Hospital, together with measurements, tracings, and computations made by human experts. The total number of videos in the EchoNet-Dynamic collection is 10,030, each annotated with corresponding left ventricular coordination data. For training, validation, and testing, these videos are separated into three groups with 75%, 12.5%, and 12.5% of the total videos in each group, respectively. Each video in the collection depicts either a standard, four-chamber apical view or one that has been zoomed in. The videos have a resolution of 112 by 112 pixels. For the three metrics of ejection percentage, end-systolic volume, and end-diastolic volume, the dataset offers coordination annotations. There are associated CSV files for each video, namely “FileList.csv” and “VolumeTracings.csv” which contain information about the videos and the corresponding volume tracings. There are 21 coordinates for each frame for localizing the left ventricle (1 for defining the long axis and 20 for the short axis).

## 3.2 *Data preprocessing*

Based on the data in the volume tracings CSV file, the code ran through each video file and extracted the diastolic and systolic frames. The output directory includes individual images representing the retrieved frames. The volume tracing CSV file was used by the code to get the locations for the diastolic and systolic frames. The coordinates are then saved in a separate CSV file along with the name of the image that corresponds and the split value for later analysis. As there are 21 coordinates for each frame and each coordinate has four x- and y-axis values, we now have $4 \times 21 = 84$ points. We need to predict those 84 coordinates with our model.

## 3.3 *Model building with transfer learning*

Our goal was to create a model for locating the left ventricle (LV) border in echo images. To determine the best-performing architectures based on sensitivity and validation loss, we tested a variety of transfer learning models, including MobileNetV2, InceptionV3, Xception, ResNet50, VGG16, and EfficientNetB7. We used pre-trained models with weights trained on the ImageNet dataset to take advantage of transfer learning. These pre-trained models were the foundational elements of our architecture, allowing us to take advantage of the skills acquired by the models in recognizing common image characteristics.

The process of creating a model using the EfficientNetB7 architecture is demonstrated in Figure 9. With ImageNet weights loaded and set to trainable, the EfficientNetB7 model was trained. For feature extraction and producing the final output, two additional conv2D layers were added on top of the pre-trained model to modify them for the LV border localization job. We followed the same structure for the rest of the transfer learning models. But we modified the kernel size of the additional layers to adjust with each model. The models were able to learn characteristics and patterns particular to LV border localization.

We experimented with several optimization techniques and learning rates in order to maximize the performance of the pre-trained models. We evaluated the learning rates of $10^{-3}$, $10^{-4}$, $10^{-5}$, and $10^{-6}$ and compared the optimization techniques with stochastic gradient descent (SGD), Adam, and RMSprop. According to our analysis, we discovered that Adam, when used with a learning rate of $10^{-4}$ had the best results for all the models in terms of sensitivity and validation loss. Early stopping avoided overfitting by keeping track of the models' performance on the validation set and ending training if no progress was shown after a certain number of epochs.

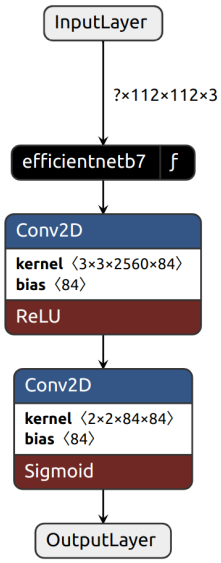


**Figure 9.** Model Summary of EfficientNet-B7 Model with Custom Top Layers.

### 3.4 *Measurements*

We calculated the mean absolute error (MAE) between the predicted coordinates and the actual coordinates for the validation and test datasets to evaluate the model's performance. Additionally, using the expected coordinates, we computed the ejection percent (EF) for each frame extracted. The left ventricle's measured volumes at the end-diastolic (ED) and end-systolic (ES) phases were used to calculate the EF. Using the measured coordinates, the volume of the LV was approximated, and the estimated volume was then scaled to the actual volume. Simpson's approach was then used to calculate the EF as a percentage [35].

$$EF = \frac{ED\ Volume - ES\ Volume}{ED\ Volume} \times 100 \tag{1}$$

The length of the long axis was calculated by computing the Euclidean distance between the two coordinates on the LV boundary. To estimate the height of a disk-shaped segment of the LV, this distance is divided by a scaling factor (20 in this case). The diameter of the LV was then computed for each short axis by determining the Euclidean distance between two sites on the short axis. The radius (r) was then calculated by dividing the diameter by two. Each disk-shaped section's area is computed using the formula: The overall volume of the LV is calculated by adding the areas of all the disk-shaped portions and multiplying this amount by the height of the LV disk. For categorizing EF, we have considered the patient's condition "normal" when EF is greater than 50%, "mild" from 40 to 49%, and "abnormal" when EF is less than 40% [35]. In addition, we created confusion matrices for the training and the test dataset to perform error analysis.

## 4. Results and discussion

For the proper diagnosis and treatment of cardiovascular disorders, the LVEF must be predicted with accuracy. The model used a variety of transfer learning architectures and was thoroughly evaluated using a test and validation dataset. The model's usefulness in clinical applications was evaluated using a variety of indicators, such as MAE, sensitivity analysis, and confusion matrices. Upon analyzing the results from Table 2, several key observations can be made:

**Table 2.** Performance Metrics for Various Transfer Learning Models.

| | | Confusion Matrix acc. | Sensitivity of Each Class | | | MAE Test Data |
|---|---|---|---|---|---|---|
| | | | Normal | Mild | Abnormal | |
| EfficientNetB7 | Train | 0.938 | 0.988 | 0.673 | 0.896 | 0.013 |
| | Test | 0.769 | 0.838 | 0.384 | 0.716 | |
| VGG 16 | Train | 0.932 | 0.954 | 0.739 | 0.927 | 0.014 |
| | Test | 0.71 | 0.744 | 0.442 | 0.791 | |
| Inception V3 | Train | 0.683 | 0.723 | 0.324 | 0.755 | 0.021 |
| | Test | 0.635 | 0.667 | 0.282 | 0.726 | |
| Xception | Train | 0.937 | 0.97 | 0.728 | 0.933 | 0.014 |
| | Test | 0.727 | 0.727 | 0.362 | 0.736 | |
| MobileNet V2 | Train | 0.799 | 0.869 | 0.385 | 0.77 | 0.017 |
| | Test | 0.719 | 0.787 | 0.297 | 0.691 | |
| ResNet 50 | Train | 0.907 | 0.959 | 0.609 | 0.879 | 0.015 |
| | Test | 0.721 | 0.78 | 0.275 | 0.756 | |

### 4.1 *Confusion matrix accuracy*

The classification performance of the model was evaluated using the overall confusion matrix accuracy. On the test data, EfficientNetB7 had the highest accuracy among the models, with a score of 0.748. Accuracy values of 0.635, 0.710, 0.727, 0.716, 0.719, and 0.721 were attained using InceptionV3, VGG16, Xception, MobileNetV2, and ResNet50, respectively. According to these findings, EfficientNetB7 fared comparably better at correctly categorizing the classes.

## 4.2 *Sensitivity of each class*

Sensitivity gauges how well a model can identify examples of a certain class. EfficientNetB7 showed greater sensitivity to the test data across all classes, with scores of 0.838 for "normal," 0.384 for "mild," and 0.716 for the "abnormal" class. The sensitivity values for the other models were often lower. According to these results, we got lower sensitivity for correctly detecting the "Mild" class.

## 4.3 *Mean Absolute Error (MAE)*

The MAE measures the average absolute deviation between the predicted and actual values. The model with the lowest MAE was EfficientNetB7 with 0.013, followed by VGG16 with 0.014. These results indicate that EfficientNetB7 and VGG16 produced the most accurate predictions relative to the actual values. EfficientNetB7 exhibited excellent overall performance, excelled in accuracy, and achieved a low MAE, indicating its efficacy in classifying and localizing LV boundaries with precision.

## 4.4 *Loss learning curve*

Analyzing the validation loss curve Figure 10 in the context of the transfer learning models studied in this paper enables evaluation of the model's generalizability to new data and detection of overfitting. The training and validation loss values both significantly decline throughout the early epochs, demonstrating that the EfficientNetB7 models are successfully learning from the training data. The model's performance on the training data keeps getting better as the training goes on, which causes the training loss to keep going down. Additionally, the validation loss shows how effectively the model generalizes to fresh data.

The validation loss initially tends to drop, showing that the model is getting better at generalizing. This mismatch shows that the model is getting too specialized in learning the training data and is not generalizing effectively to new, unknown data. However, at a certain point, the models start overfitting. We thus determined the ideal time to halt model training and avoid overfitting by carefully evaluating the validation loss curve. The validation loss often reaches its minimum value at this time, indicating the optimum trade-off between model complexity and generalization capacity.

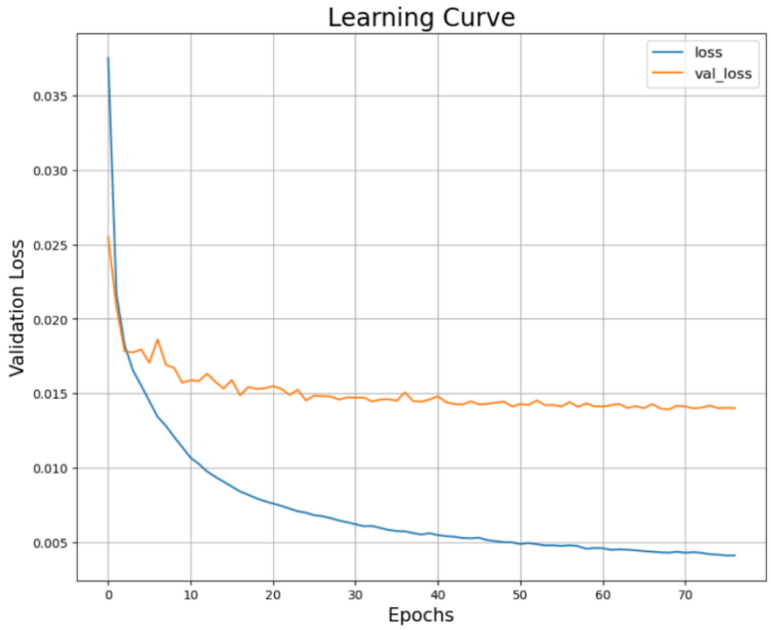


**Figure 10.** Loss Learning Curve of EfficientNet-B7.

## 4.5 *Discussion*

On the validation and test datasets, the model exhibits excellent performance using a variety of convolutional neural network configurations. Although EfficientNetB7 has more trainable layers, it outperforms the other models when it comes to performance and efficiency with customized top layers. Figure 11 shows how well EfficientNetB7 traced the left ventricle's location. The low MAE values for both the validation and test datasets indicate that the model can accurately predict LVEF.

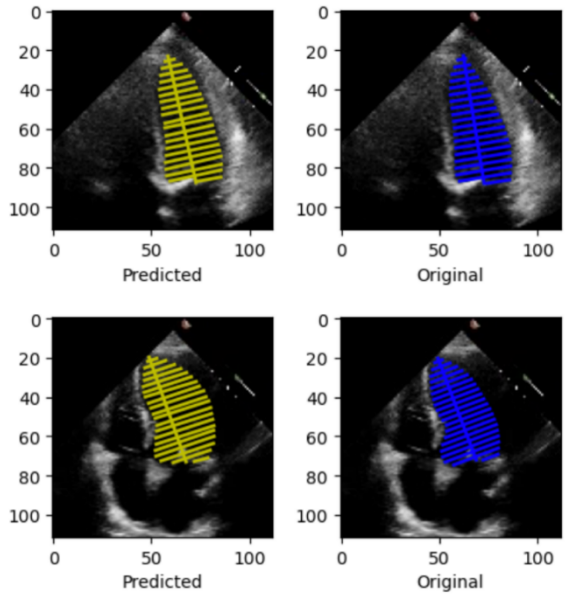


**Figure 11.** EfficientNet-B7 Prediction and Original Left Ventricle Position.

The research revealed varied levels of sensitivity in the "normal", "mild", and "abnormal" categories. The model is highly sensitive to the "normal" and "abnormal" classifications, correctly predicting the large majority of cases in these categories. This shows that the model is capable of classifying heart function examples as normal or significantly abnormal. The model's sensitivity for the "Mild" class, on the other hand, is moderate, indicating that it is difficult to reliably forecast LVEF levels when cardiac function is mild.

Several factors may account for this disparity. The dataset contains fewer instances representing the "Mild" class than the "Normal" and "Abnormal" classes, which is one possible cause. This imbalance may hinder the model's capacity to accurately learn and generalize patterns associated with the "Mild" class. Variations in image quality may also have an impact on the model's efficacy. Cardiac images may manifest variations in terms of resolution, noise, anomalies, and image acquisition techniques. These variations can make it difficult for the model to extract the pertinent characteristics and patterns required for an accurate LVEF prediction.

Numerous risk assessment instruments have been developed over time for clinical outcomes like hospitalization and death in HF patients [36, 37, 38]. Hemoglobin level, blood urea nitrogen, time since previous HF hospitalization, and health status were predictive of HF hospitalization, while blood urea nitrogen, BMI, and health status were predictive of death. Conversely, the current investigation focuses on EF value variations in patients. When compared to traditional clinical trials or registry-based investigations, the current study encounters some of the same difficulties associated with Electronic Health Record (EHR)-based cohort analysis [39]. Researchers must deal with challenges related to missing data items and inconsistent availability of data elements across systems when utilizing EHR data for clinical research. While EHR data was largely collected for clinical care, traditional clinical trials and registry-based cohort studies collect data intended to solve a specific research issue.

The same data may not be gathered consistently or made available to all users of EHR systems. It is commonly known that different EHR systems have different procedures for gathering and documenting data, and that these procedures can vary depending on patient demographics and local clinical practices. The data may not have been routinely collected in EHR systems, even if some of these characteristics have been recorded. As a result, when tested using patient data from other sites, a model trained on data from one site might not perform as predictively. As an extension of this study, we would like to observe how well deep learning (DL) models perform when trained on data from one site and tested on data that originated from a separate location.

Furthermore, if a patient's EF was assessed during an acute phase, a medical intervention or unfavorable event would have altered the findings within that brief time. The current study did not collect any data on the patients' HF treatment phase, such as acute or non-acute, and instead employed EF measures that were primarily taken from echocardiograms. A given outcome can be significantly influenced by features in DL algorithms for reasons other than biological relationship. When comparing characteristics with low prevalent or missing values, which may not necessarily be the result of a stronger biological link, features with high prevalence, low missingness, etc., in the total sample under analysis may have an excessive impact on the results. For a particular desired outcome, not every feature has the same statistical importance. It is difficult to identify significant biological associations between features and outcome variables unless we train DL models in a "controlled" feature environment. Conducting such training is extremely difficult due to the considerable diversity of EHR data across sites. While several features were found to be important contributors in the DL prediction of EF changes, we do not suggest that these features are the ones producing the change in patients' EF measures, nor do we know how much an influence these features have on a patient's EF measurement.

## 5. Conclusions and future work

This study compares sophisticated transfer learning models for predicting the LVEF from cardiac images. Future research could concentrate on resolving these issues by implementing preprocessing and exploratory data analysis to clean up the data. Beat-to-beat cardiac cycle detection is needed for measuring the average ejection fraction of every cycle, which we didn't perform in our work. In addition, the inherent difficulty of classifying cases as "mild" poses a significant obstacle. The line between normal and moderate abnormalities can be subjectively determined and may require expert interpretation. The model's ability to reliably categorize "mild" instances might be enhanced by including more clinical details and patient data. Several tactics may be investigated in order to improve the model's performance even further. Techniques for enhancing data can be used to broaden the variety and volume of training data. The capacity of the model to generalize to unknown changes may be improved by enhancing the dataset by performing modifications to the current images, such as rotation, translation, and scaling. The performance of the model can be further improved by training numerous deep learning models with various architectures or hyperparameters and integrating their predictions. In the context of LVEF prediction, it is crucial to consider the clinical consequences of the deep learning model. The model has the potential to simplify the assessment procedure, lessen human error, and deliver quick and reliable information for clinical decision-making because it is automated.

In summary, EffieceintNetB7 demonstrated excellent performance, high sensitivity for normal or abnormal cardiac function, and moderate sensitivity for mild abnormalities. Our CNN regression model predicts the left ventricle border directly from coordinates, eliminating the need for any time-consuming segmentations of data before training.

## Conflict of interest

There is no conflict of interest for this study.

## References


[1] P. A. Heidenreich et al., "2022 ACC/AHA/HFSA Guideline for the Management of Heart Failure," *J. Card. Fail.*, vol. 28, pp. e1–e167, 2022, https://doi.org/10.1016/j.cardfail.2022.02.010.

[2] C. S. Lam and S. D. Solomon, "The middle child in heart failure: heart failure with mid-range ejection fraction (40–50%)," *Eur. J. Hear. Fail.*, vol. 16, no. 9, pp. 1049–1055, 2014, https://doi.org/10.1002/ejhf.159.

[3] A. Rastogi, E. Novak, A. E. Platts, and D. L. Mann, "Epidemiology, pathophysiology and clinical outcomes for heart failure patients with a mid-range ejection fraction," *Eur. J. Hear. Fail.*, vol. 19, no. 12, pp. 1597–1605, 2017, https://doi.org/10.1002/ejhf.879.

[4] R. K. Cheng et al., "Outcomes in patients with heart failure with preserved, borderline, and reduced ejection fraction in the Medicare population," *Am. Hear. J.*, vol. 168, no. 5, pp. 721–730.e3, 2014, https://doi.org/10.1016/j.ahj.2014.07.008.

[5] A. A. Inamdar and A. C. Inamdar, "Heart Failure: Diagnosis, Management and Utilization," *J. Clin. Med.*, vol. 5, no. 7, p. 62, 2016, https://doi.org/10.3390/jcm5070062.

[6] D. S. Lee et al., "Relation of Disease Pathogenesis and Risk Factors to Heart Failure With Preserved or Reduced Ejection Fraction," *Circulation*, vol. 119, no. 24, pp. 3070–3077, 2009, https://doi.org/10.1161/circulationaha.108.815944.

[7] "Top ten causes of death," Accessed: Jul. 4, 2023. [Online.] Available: https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death.

[8] H. Li et al., "EchoEFNet: Multitask deep learning network for automatic calculation of left ventricular ejection fraction in 2D echocardiography," *Comput. Biol. Med.*, vol. 156, p. 106705, 2023, https://doi.org/10.1016/j.compbiomed.2023.106705.

[9] M. Cameli et al., "Echocardiographic assessment of left ventricular systolic function: from ejection fraction to torsion," *Hear. Fail. Rev.*, vol. 21, no. 1, pp. 77–94, 2015, https://doi.org/10.1007/s10741-015-9521-8.

[10] Y. Nagata et al., "Impact of image quality on reliability of the measurements of left ventricular systolic function and global longitudinal strain in 2D echocardiography," *Echo Res. Pract.*, vol. 5, no. 1, pp. 27–39, 2018, https://doi.org/10.1530/ERP-17-0047.

[11] Y. Guo, S. Green, L. Park, and L. Rispen, "Left ventricle volume measuring using echocardiography sequences," in *Proc. 2018 IEEE DICTA*, Canberra, ACT, Australia, Dec. 10-13, 2018, https://doi.org/10.1109/DICTA.2018.8615766.

[12] R. M. Abazid et al., "Visual versus fully automated assessment of left ventricular ejection fraction," *Avicenna J. Med.*, vol. 8, no. 2, pp. 41–45, 2018, https://doi.org/10.4103/ajm.ajm_209_17.

[13] S. Leclerc et al., "Deep Learning for Segmentation Using an Open Large-Scale Dataset in 2D Echocardiography," *IEEE Trans. Med. Imaging*, vol. 38, no. 9, pp. 2198–2210, 2019, https://doi.org/10.1109/tmi.2019.2900516.

[14] M. Xin and Y. Wang, "Research on image classification model based on deep convolution neural network," *EURASIP J. Image Video Process.*, vol. 2019, no. 1, p. 40, 2019, https://doi.org/10.1186/s13640-019-0417-8.

[15] J. Zhang, K. Yu, Z. Wen, X. Qi, and A. K. Paul, "3D Reconstruction for Motion Blurred Images Using Deep Learning-based Intelligent Systems," *Comput. Mater. Contin.*, vol. 66, no. 2, pp. 2087–2104, 2021, https://doi.org/10.32604/cmc.2020.014220.

[16] M. Arifuzzaman, R. Hasan, T. J. Toma, S. B. Hassan, and A. K. Paul, "An Advanced Decision Tree-Based Deep Neural Network in Nonlinear Data Classification," *Technologies*, vol. 11, no. 1, p. 24, 2023, https://doi.org/10.3390/technologies11010024.

[17] K. Kusunose et al., "A Deep Learning Approach for Assessment of Regional Wall Motion Abnormality From Echocardiographic Images," *JACC: Cardiovasc. Imaging*, vol. 13, no. 1, pp. 374–381, 2019, https://doi.org/10.1016/j.jcmg.2019.02.024.

[18] F. Zhuang et al., "A Comprehensive Survey on Transfer Learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, 2021, https://doi.org/10.1109/jproc.2020.3004555.

[19] S. Ono et al., "Automated Endocardial Border Detection and Left Ventricular Functional Assessment in Echocardiography Using Deep Learning," *Biomedicines*, vol. 10, no. 5, p. 1082, 2022, https://doi.org/10.3390/biomedicines10051082.

[20] A. Lagopoulos and D. Hristu-Varsakelis, “Measuring the Left Ventricular Ejection Fraction using Geometric Features,” in *Proc. 2022 IEEE CBMS*, Shenzen, China, Jul. 21-23, 2022, https://doi.org/10.1109/CBMS55023.2022.00008.

[21] E. Smistad, E. N. Steinsland, and L. Lovstakken, “Real-time 3D left ventricle segmentation and ejection fraction using deep learning,” in *Proc. 2021 IEEE IUS*, Xi’an, China, Sep. 11-16, 2021, https://doi.org/10.1109/IUS52206.2021.9593301.

[22] O. Moal et al., “Explicit and automatic ejection fraction assessment on 2D cardiac ultrasound with a deep learning-based approach,” *Comput. Biol. Med.*, vol. 146, p. 105637, 2022, https://doi.org/10.1016/j.compbiomed.2022.105637.

[23] F. M. Asch et al., “Automated Echocardiographic Quantification of Left Ventricular Ejection Fraction Without Volume Measurements Using a Machine Learning Algorithm Mimicking a Human Expert,” *Circ. Cardiovasc. Imaging*, vol. 12, p. e009303, 2019, https://doi.org/10.1161/circimaging.119.009303.

[24] P. Baheti, “A Newbie-Friendly Guide to Transfer Learning,” Accessed: Jul. 4, 2023. [Online]. Available: https://www.v7labs.com/blog/transfer-learning-guide.

[25] P. Sharma, “Understanding Transfer Learning for Deep Learning,” Accessed: Jul. 4, 2023. [Online]. Available: https://www.analyticsvidhya.com/blog/2021/10/understanding-transfer-learning-for-deep-learning/.

[26] “Everything you need to know about VGG16,” Accessed: Jul. 4, 2023. [Online]. Available: https://medium.com/@mygreatlearning/everything-you-need-to-know-about-vgg16-7315defb5918.

[27] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint, https://doi.org/10.48550/arXiv.1409.1556.

[28] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” in *Proc. IEEE CVPR*, Salt Lake City, UT, USA, Jun. 18-23, 2018, https://doi.org/10.48550/arXiv.1801.04381.

[29] M. Tan and Q. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in *Proc. ICML*, Long Beach, CA, USA, Jun. 9-15, 2019, pp. 6105-6114.

[30] “EfficientNet: Improving Accuracy and Efficiency through AutoML and Model Scaling,” Accessed: Jul. 5, 2023. [Online]. Available: https://blog.research.google/2019/05/efficientnet-improving-accuracy-and.html.

[31] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. IEEE CVPR*, Las Vegas, NV, USA, Jun. 27-30, 2016, pp. 770–778, https://doi.org/10.1109/CVPR.2016.90.

[32] F. Chollet, “Xception: Deep learning with depthwise separable convolutions,” in *Proc. 2017 IEEE CVPR*, Honolulu, HI, USA, Jul. 21-26, 2017.

[33] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception Architecture for Computer Vision,” in *Proc. IEEE CVPR*, Las Vegas, NV, USA, Jun. 27-30, 2016, pp. 2818–2826, https://doi.org/10.1109/cvpr.2016.308.

[34] D. Ouyang et al., “Video-based AI for beat-to-beat assessment of cardiac function,” *Nature*, vol. 580, no. 7804, pp. 252–256, 2020, https://doi.org/10.1038/s41586-020-2145-8.

[35] “What is Ejection Fraction?,” Accessed: Jul. 4, 2023. [Online]. Available: https://www.urmc.rochester.edu/encyclopedia/content.aspx?contenttypeid=56&contentid=DM14.

[36] S. Angraal et al., “Machine Learning Prediction of Mortality and Hospitalization in Heart Failure With Preserved Ejection Fraction,” *JACC: Heart Fail.*, vol. 8, no. 1, pp. 12–21, 2019, https://doi.org/10.1016/j.jchf.2019.06.013.

[37] W. Ouwerkerk, A. A. Voors, and A. H. Zwinderman, “Factors Influencing the Predictive Power of Models for Predicting Mortality and/or Heart Failure Hospitalization in Patients With Heart Failure,” *JACC: Heart Fail.*, vol. 2, no. 5, pp. 429–436, 2014, https://doi.org/10.1016/j.jchf.2014.04.006.

[38] R. J. Desai et al., “Comparison of Machine Learning Methods With Traditional Models for Use of Administrative Claims With Electronic Medical Records to Predict Heart Failure Outcomes,” *JAMA Netw. Open*, vol. 3, no. 1, p. e1918962, 2019, https://doi.org/10.1001/jamanetworkopen.2019.18962.

[39] J. Pathak, A. N. Kho, and J. C. Denny, “Electronic health records-driven phenotyping: challenges, recent advances, and perspectives,” *J. Am. Med. Inform. Assoc.*, vol. 20, no. e2, pp. e206–e211, 2013, https://doi.org/10.1136/amiajnl-2013-002428.