

Chapter 1

Introduction

1.1 Motivation

Humankind has always been struggling with different kinds of challenges that life puts through. Be it survival from natural calamities or man-made disasters, we always found a way to survive for generations. The innovative mindset will survive to make us innovate new techniques to overcome our limitations, either by creating something new from raw or mimicking some natural existence to solve our problems.

In a country with a vast population, the situation of a Pandemic like- COVID-19 becomes very dangerous and risks the lives of individuals at stake. Poor health infrastructure, less doctor-to-patient ratio, dense population, and lack of education, all result in the chaotic situation of panic among the people, black marketing of essential medical supplies, and whatnot. We ourselves have witnessed various unfortunate incidences, where someone is losing their dear ones just because of lack of proper diagnosis on time or wrong diagnosis of the disease.

Artificial intelligence[1] is one such area, which mimics human brain function to solve problems that are either really complex or absolutely impossible to solve with traditional computer algorithms by coding. For example- Computer Vision is possible because of the evolution of huge data along with high computational power availability, otherwise, it's almost impossible to code for all the facial features of one human or some other objects and use that to identify different subjects of choice.

This has been the main motivating factor to search for the tools that might assist in diagnoses at a rapid scale, cheaper in price as well and with greater accuracies of predictions. The goal is to assist in the diagnosis of COVID-19 using X-rays and CT, which are available almost in every part of the country. The only problem is there about the correct interpretation of the disease. In the initial phases of COVID-19, the CT score became very important for everyone to find the chest infection rate. But COVID-19 is not the only reason for a chest infection, as there are various types of Pneumonia as well (Viral Pneumonia and Bacterial Pneumonia), which also causes chest infection as it marks the presence of fluid in the lungs. During the panic situation, people went for a CT score and without even identifying it as Covid or Pneumonia, they started treatment for it assuming it was Covid as per CT score values, which made them use steroids and also stored

oxygen cylinders at home. Due to this, the supply chain of essentials to fight with Covid got interrupted and lot many people died because of the absence of oxygen and other medical supplies.

This thesis deals with one of the subsets of Artificial intelligence- Deep Transfer Learning, which utilizes pre-trained Deep neural network weights to retrain the model as per our problem statement. Here, our main focus is to utilize deep learning for medical image analysis to help doctors with better and more rapid diagnoses of diseases. To achieve that, we have utilized chest X-ray and CT images for building the classifier to identify them among different classes like- COVID-19, normal, Bacterial Pneumonia, And Viral Pneumonia. In the second approach, we tried to perform Cardiac MRI segmentation.

1.2 Scope of the research

Deep transfer learning is a machine learning technique that involves using the knowledge and weights learned from a pre-trained deep learning model on one task and applying it to a new task. This allows the new model to learn much faster and achieve higher performance than if it were trained from scratch on the new task.

The scope of deep transfer learning is broad and can be applied to a wide range of tasks, including image and video recognition, natural language processing, and speech recognition. Some examples of the use of deep transfer learning include:

1. **Image classification:** Transfer learning can be used to improve the performance of image classification tasks, such as identifying objects in images or diagnosing diseases from medical images. VGG-16 Net[2] is a very popular classification model, which is shown in Figure 1.1
2. **Object detection:** Transfer learning can be used to improve the performance of object detection tasks, such as identifying and locating objects in images or videos. Basic block diagram of object detection[3] is shown in Figure 1.2.
3. **Natural language processing:** Transfer learning can be used to improve the performance of natural languages processing tasks, such as sentiment analysis or machine translation.
4. **Speech recognition:** Transfer learning can be used to improve the performance of speech recognition tasks, such as transcribing audio recordings or identifying speakers.

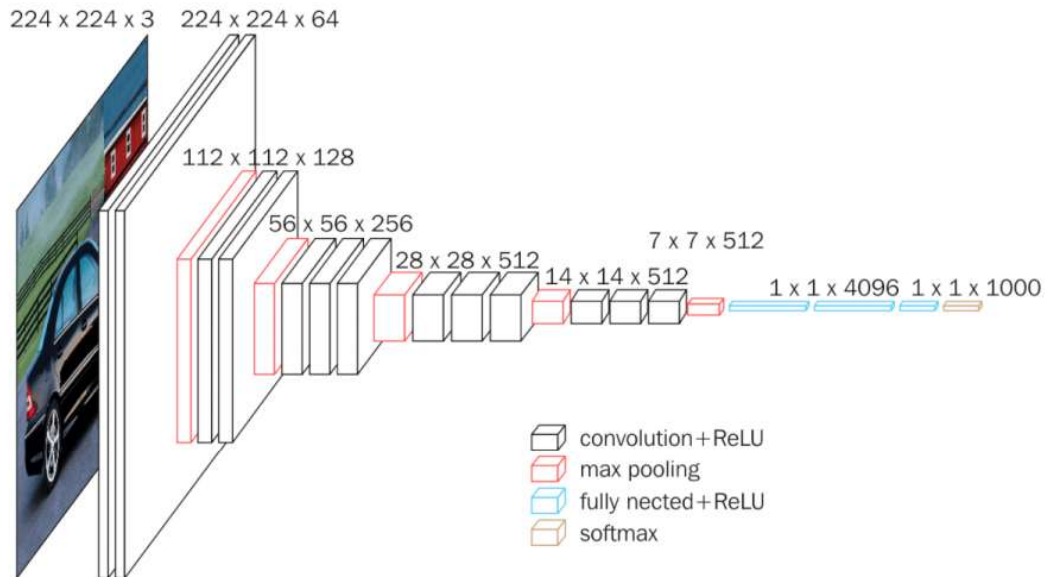


FIGURE 1.1: VGG-16 Net.

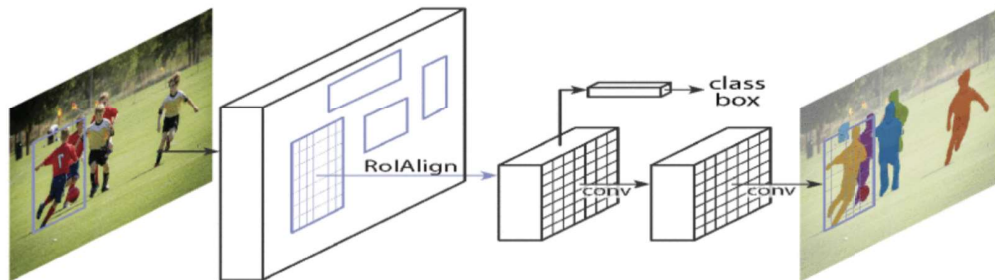


FIGURE 1.2: Object Detection

Overall, the scope of deep transfer learning is wide and it has the potential to significantly improve the performance of a wide range of machine learning tasks.

1.3 Challenges

Medical image classification and segmentation are closely related techniques that are used to analyze and understand medical images. Medical image classification involves using machine learning algorithms to classify images based on their content, while medical image segmentation involves dividing an image into distinct regions or objects based on their characteristics.

The scope of medical image classification and segmentation is wide and covers a variety of applications in the field of healthcare, including:

1. **Diagnosis:** Both medical image classification and segmentation can be used to assist in the diagnosis of diseases or conditions by identifying abnormalities in the images.
2. **Treatment planning:** Medical image segmentation can be used to identify and isolate specific structures in the image, which can be helpful for treatment planning, such as in radiation therapy or surgical planning.
3. **Monitoring treatment:** Both medical image classification and segmentation can be used to track the progression of a disease or the effectiveness of a treatment by comparing images over time.
4. **Screening:** Medical image classification can be used to automate the screening process for diseases or conditions by identifying abnormalities in images.
5. **Biomarker discovery:** Medical image classification can be used to identify patterns or features in images that may be indicative of a particular disease or condition, which can help to identify potential biomarkers.
6. **Image-guided procedures:** Medical image segmentation can be used to guide image-guided procedures, such as biopsies or interventional radiology procedures, by providing a detailed map of the relevant structures in the image.
7. **Research:** Both medical image classification and segmentation can be used to support research by enabling the automatic analysis of large datasets of medical images.

Overall, medical image classification and segmentation have a wide range of applications in healthcare and have the potential to improve the accuracy and efficiency of diagnosis, treatment planning, monitoring, screening, biomarker discovery, image-guided procedures, and research.

Deep transfer learning is a machine learning technique that involves using the knowledge and weights learned from a pre-trained deep learning model on one task and applying it to a new task. While deep transfer learning can be highly effective in many cases, it is not without its challenges. Some of the challenges of deep transfer learning include:

1. **Data availability:** In order to perform deep transfer learning, a large amount of labeled data is needed for both the pre-trained model and the new task. This can be a challenge if the data is not readily available or if it is expensive to collect and label.
2. **Domain shift:** If the new task is significantly different from the original task, the transferred knowledge may not be relevant or may need to be modified to fit the new domain. This is known as a domain shift and can be challenging to overcome.
3. **Overfitting:** Transfer learning involves fine-tuning the pre-trained model on the new task, which can lead to overfitting if the model is not properly regularized. Overfitting occurs when the model is too closely tailored to the training data and does not generalize well to new data.
4. **Computational requirements:** Transfer learning involves training a large, complex model, which can be computationally intensive and require specialized hardware and software.
5. **Human expertise:** Transfer learning requires a deep understanding of the pre-trained model, the new task, and how to adapt the transferred knowledge to fit the new task. This requires a high level of human expertise and may be a challenge in some cases.

1.4 Outline of Proposed Solution

1.4.1 Problems

1.4.1.1 Dying Neuron

The problem of dying neurons occurs in neural networks when some of the neurons in the network become inactive or "dead" during training. This can occur when the weights of the neurons are updated in such a way that the neuron's output is always zero or a constant value. When this happens, the neuron is unable to contribute to the overall output of the network and becomes ineffective.

Dying neurons can occur when the activation function used in the network is not well-suited to the task. For example, if the activation function is too linear, the weights of the neurons may be adjusted in such a way that the output is always zero or a constant value.

Dying neurons can also occur when the network is over-regularized, meaning that the constraints applied to the weights during training are too strong. This can cause the weights to be updated in such a way that the neuron's output is always zero or a constant value.

Dying neurons can be a problem because they can significantly reduce the performance of the network and make it difficult for the network to learn effectively. To prevent dying neurons, it is important to carefully select the activation function and choose appropriate regularization techniques to ensure that the weights are updated in a way that allows the neurons to contribute to the overall output of the network.

1.4.1.2 Vanishing Gradient

The vanishing gradient problem[4] is a common issue that can occur during the training of deep neural networks. It refers to the phenomenon where the gradients of the weights become very small or "vanish" as they are backpropagated through the network. This can make it difficult for the network to learn effectively, as the small gradients result in slow or minimal weight updates.

The vanishing gradient problem is often caused by the use of activation functions that saturate for large input values. Activation functions that saturate at high or low values produce gradients that are close to zero, which can make it difficult for the network to learn effectively. Activation functions that are prone to saturation include sigmoid, tanh, and some variants of ReLU.

The vanishing gradient problem can be addressed by using activation functions that do not saturate for large input values, such as leaky ReLU or SELU. It can also be addressed by using techniques such as batch normalization, which helps to stabilize the distribution of the activations and prevent the gradients from vanishing.

In summary, the vanishing gradient problem is a common issue that can occur during the training of deep neural networks and is caused by the use of activation functions that saturate for large input values. It can be addressed by using activation functions that do not saturate and by using techniques such as batch normalization to stabilize the distribution of the activations.

1.4.1.3 Computational Cost

The computational cost[5] of a deep learning model refers to the amount of computational resources, such as CPU time and memory, required to train and deploy the model. The computational cost

of a model can have a significant impact on its practicality and efficiency, as it determines how long the model will take to train and how much hardware is required to run it.

There are several factors that can influence the computational cost of a deep learning model, including:

1. **Model size:** Larger models with more parameters tend to have higher computational costs, as they require more CPU time and memory to train and run.
2. **Training data size:** The amount of training data can also impact the computational cost of a model, as larger datasets require more computational resources to process.
3. **Training and inference complexity:** The complexity of the model's architecture and the algorithms used for training and inference can also affect the computational cost. For example, models that use more complex activation functions or require multiple passes through the data during training may have higher computational costs.
4. **Hardware:** The hardware used to train and run the model can also impact the computational cost. For example, using more powerful hardware, such as a GPU, can reduce the computational cost of training a model.

Overall, the computational cost of a deep learning model can be a significant factor in its practicality and efficiency, and it is important to consider this when designing and deploying a model.

1.4.2 Solutions

1.4.2.1 Self-Normalization

Self-normalization is a technique[6] used in deep learning to ensure that the activations in a neural network have a stable distribution. This can help to prevent the vanishing or exploding gradient problem, which can occur during training when the gradients become too large or too small.

There are several ways to achieve self-normalization in a neural network. One approach is to use an activation function that has the property of self-normalizing, such as the self-normalizing exponential linear unit (SELU). SELU automatically adjusts the mean and variance of the activations to maintain a stable distribution.

Another approach is to use batch normalization, which involves normalizing the activations of each layer across a batch of data. This helps to stabilize the distribution of the activations and prevent the gradients from vanishing or exploding.

Self-normalization can be an effective technique for improving the training and performance of deep neural networks. It can help to prevent the vanishing or exploding gradient problem and allow the network to learn more effectively.

In summary, self-normalization is a technique used in deep learning to ensure that the activations in a neural network have a stable distribution. It can be achieved using activation functions that have the property of self-normalizing, such as SELU, or by using techniques such as batch normalization.

1.4.2.2 SELU over RELU

SELU (self-normalizing exponential linear unit) and RELU (rectified linear unit) are both types of activation functions[7] used in neural networks. Activation functions are mathematical functions that are applied to the output of a neuron to determine whether it should be activated or not.

RELU is a popular activation function that has a simple mathematical form and is easy to compute. It has the form $f(x) = \max(0, x)$, which means that it sets all negative values to zero and leaves positive values unchanged. RELU has been shown to be effective at improving the training speed and performance of deep neural networks.

$$\text{Relu}(x) = \begin{cases} 0 & \text{if } |x| \leq 0 \\ X & \text{if } |x| > 0 \end{cases} \quad (1.1)$$

SELU is another activation function that was designed to address some of the limitations of RELU. SELU has the form -

$$\text{Selu}(x) = \lambda \begin{cases} \alpha \exp\{x\} - \alpha & \text{if } |x| \leq 0 \\ X & \text{if } |x| > 0 \end{cases} \quad (1.2)$$

where lambda and alpha are constants. SELU has the property of self-normalizing, meaning that it can automatically adjust the mean and variance of the activations to maintain a stable distribution. This can help to prevent the vanishing or exploding gradient problem, which can occur during training when the gradients become too large or too small.

In summary, SELU is an activation function that was designed to address some of the limitations of RELU and has the property of self-normalizing. It may be more effective at preventing the vanishing or exploding gradient problem and maintaining a stable distribution of activations, but it is more complex to compute than RELU.

1.4.2.3 Inception Net

Inception Net[8] is a deep convolutional neural network architecture that was developed by Google for the ImageNet Large Scale Visual Recognition Challenge (ILSVRC). InceptionNet was designed to be computationally efficient and achieve high performance on image classification tasks.

InceptionNet is composed of modular blocks called "inception modules," which are designed to capture a wide range of features at different scales. These modules are connected in a "network-in-network" structure, where each module is a small network that processes the input data and passes it on to the next module. The output of the final module is fed into a global average pooling layer and a fully connected layer to produce the final output.

InceptionNet is known for its efficient use of computational resources and has been widely used for image classification tasks. It has been shown to achieve high performance on a variety of benchmarks and has been widely adopted in industry and research.

In summary, InceptionNet is a deep convolutional neural network architecture that was developed for image classification tasks and is known for its efficient use of computational resources and high performance. It is composed of modular blocks called "inception modules" and has a "network-in-network" structure.

InceptionNet has several advantages compared to other deep learning architectures, including:

1. **Efficient use of computational resources:** InceptionNet is designed to be computationally efficient, making it well-suited for deployment on devices with limited computational resources, such as smartphones or embedded systems.
2. **Improved performance:** InceptionNet has been shown to achieve high performance on image classification tasks, outperforming many other deep learning architectures.
3. **Modular design:** InceptionNet is composed of modular blocks called "inception modules," which can be easily combined and modified to suit the needs of a particular task. This allows for flexibility in the design of the network.

4. **Dimensionality reduction:** InceptionNet uses a technique called "dimensionality reduction" to reduce the number of parameters in the network, which helps to prevent overfitting and improve the generalization of the model.
5. **Transfer learning:** InceptionNet can be easily fine-tuned for new tasks using transfer learning, which allows the network to learn much faster and achieve higher performance than if it were trained from scratch.

Overall, InceptionNet is a highly effective deep learning architecture that has been widely used for image classification tasks and has several advantages, including computational efficiency, high performance

1.4.2.4 VGG-16 Net

VGG16 is a convolutional neural network architecture[2] that was developed by the Visual Geometry Group (VGG) at the University of Oxford. VGG16 is a 16-layer deep convolutional neural network that was trained on the ImageNet dataset, which is a large dataset of images used for image classification and object recognition tasks.

VGG16 is known for its simplicity and strong performance on image classification tasks. It consists of a series of convolutional and max-pooling layers, followed by a few fully connected layers. The convolutional layers extract features from the input images, and the fully connected layers use these features to make the final classification.

VGG16 is widely used in research and industry and has been used in a variety of applications, including image classification, object detection, and segmentation. It is often used as a baseline model to compare the performance of other deep learning architectures.

In summary, VGG16 is a convolutional neural network architecture that was developed for image classification tasks and is known for its simplicity and strong performance. It consists of a series of convolutional and fully connected layers and has been widely used in research and industry.

It has several advantages compared to other deep learning architectures:

1. **Simplicity:** VGG16 has a simple architecture, consisting of a series of convolutional and fully connected layers, which makes it easy to understand and implement.

2. **Strong performance:** VGG16 has been shown to achieve strong performance on image classification tasks, outperforming many other deep learning architectures.
3. **Widely used:** VGG16 is widely used in research and industry and has been used in a variety of applications, including image classification, object detection, and segmentation.
4. **Transfer learning:** VGG16 can be easily fine-tuned for new tasks using transfer learning, which allows the network to learn much faster and achieve higher performance than if it were trained from scratch.
5. **Baseline model:** VGG16 is often used as a baseline model to compare the performance of other deep learning architectures.

Overall, VGG16 is a simple and effective deep learning architecture that has been widely used for image classification tasks and has several advantages, including strong performance, simplicity, and wide adoption. It is often used as a baseline model to compare the performance of other architectures and can be easily fine-tuned for new tasks using transfer learning.

1.4.2.5 Autoencoder

An autoencoder is a type of neural network that is used for unsupervised learning. It consists of two parts: an encoder and a decoder. The encoder maps the input data to a lower-dimensional representation, known as the latent representation or bottleneck, and the decoder maps the latent representation back to the original input space.

The goal of an autoencoder is to learn a compressed representation of the input data that captures the important features or patterns in the data. This is done by training the autoencoder to reconstruct the original input from the latent representation. The encoder is trained to extract the important features from the input data, and the decoder is trained to reconstruct the original input from the latent representation.

Autoencoders can be used for a variety of tasks, including dimensionality reduction, data denoising, and data generation. They are often used as a pre-processing step for other machine learning tasks, such as classification or clustering.

Overall, autoencoders are a useful tool for unsupervised learning and can be used to learn a compressed representation of the input data that captures the important features or patterns in the data.

1.4.2.6 U-Net

UNet (U-Net) is a type of convolutional neural network (CNN) architecture that is commonly used for image segmentation tasks. It was developed by Olaf Ronneberger, Philipp Fischer, and Thomas Brox in 2015.

UNet consists of an encoder and a decoder, where the encoder is a traditional CNN that downsamples the input image to a lower-resolution feature map, and the decoder is a series of transposed convolutional layers that upsample the feature map back to the original image resolution. The encoder and decoder are connected by a series of skip connections, which concatenate the activations from the encoder layers with the upsampled activations from the decoder layers at corresponding resolutions.

UNet has several advantages, including:

1. **High accuracy:** UNet has achieved state-of-the-art performance on a variety of image segmentation tasks, including medical image segmentation.
2. **Efficient use of data:** The skip connections in UNet allow the model to use both low-level and high-level features to make predictions, which can improve the model's performance and allow it to make more efficient use of the data.
3. **Easy to implement:** UNet is relatively simple to implement and has a straightforward architecture, which makes it a popular choice for image segmentation tasks.

Overall, UNet is a powerful and widely-used CNN architecture for image segmentation tasks, and has achieved state-of-the-art performance on a variety of tasks. Its combination of skip connections and a downsampling and upsampling structure allows it to effectively use both low-level and high-level features to make predictions, which can improve its performance and make it more efficient.

1.4.2.7 Deep Transfer Learning

Deep transfer learning is a machine learning technique that involves using the knowledge and weights learned from a pre-trained deep learning model on one task and applying it to a new task. Deep transfer learning has several advantages over traditional deep learning, including:

1. **Faster training:** Transfer learning allows a new model to learn much faster than if it were trained from scratch on the new task, as it can start with the knowledge and weights learned from the pre-trained model.
2. **Higher performance:** Transfer learning can often result in higher performance on the new task compared to training a model from scratch, as the pre-trained model has already learned useful features and patterns from the original task.
3. **Reduced data requirements:** Transfer learning can reduce the amount of labeled data required for the new task, as the pre-trained model has already learned from a large amount of data. This can be particularly useful if the data for the new task is limited or expensive to obtain.
4. **Improved generalization:** Transfer learning can improve the generalization of the new model, as it has already learned from a diverse set of data and has a strong foundation of knowledge.

Overall, deep transfer learning can be an effective technique for improving the training and performance of deep learning models, particularly when data is limited or the new task is related to the original task.

1.5 Thesis Outline

Complete thesis has been organized into six chapters, which is shown in [Figure.1.3](#)-

In [Chapter1](#), we started with the motivation behind this research work, its scope, various challenges associated with it, and an outline of the proposed solutions, which deal with problems and solutions. At the end, it gives the thesis outline as a block diagram.

In [Chapter2](#), we have discussed about different medical imaging modalities like- MRI, X-ray, CT. We also discussed about Convolutional Neural Networks(CNN), Deep Transfer learning and about automated diagnosis of diseases with medical image classification and segmentation.

In [Chapter3](#), we discussed about our multiclass classification problem of classification among covid, normal, viral pneumonia and bacterial pneumonia. We have developed a transfer learning based model by utilizing the weights of pre-trained VGG-16 classifier along with Inception net to extract the features. In this, we performed case study based approach, which is divided into 3 cases

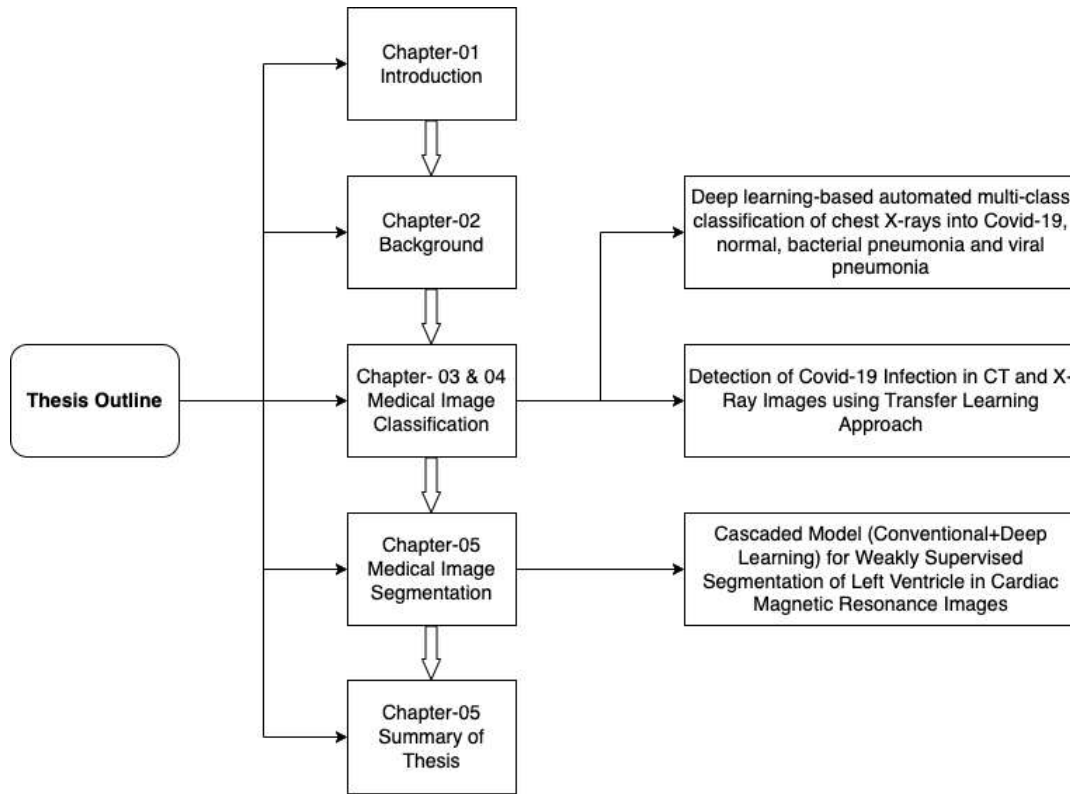


FIGURE 1.3: Thesis Outline

namely- four class classification, three class classification and two class classification. Based on the achieved accuracies, our model showed comparable performance to pre-existing methods like VGG-16, Res-Net, and Inception-Net.

In Chapter4, we discussed about Binary Class classification problem. Here we presented a study that presents an efficient classification methodology for the precise identification of infection caused by Covid-19 using CT and X-ray images. To achieve that, we have exploited the depthwise separable convolution-based model of MobileNet-V2 for feature extraction. The features of infection were supplied to the SVM classifier for training which produced accurate classification results.

In Chapter5, we have discussed about the cascaded model for weakly supervised segmentation of left ventricle in Cardiac MR Images. Here, we proposed to use a cascade of conventional and deep learning methods to obtain accurate segmentation. First, Seed Region growing, Random Walker, and K-Means clustering are used individually to roughly delineate the region of interest, and then, the generated masks are used to train the deep learning model to get the final segmentation. The proposed method is validated for the segmentation of the left ventricle in cardiac magnetic

resonance images (ACDC Dataset). The method is again cross-validated with another dataset on which the model is not previously trained.

In the last [Chapter6](#), we have summarised the thesis and gave the future directions from our proposed research work.

