ITE 4099

CAPSTONE THESIS REPORT

COMPARISON AND EFFICACY OF TRANSFER LEARNING MODELS FOR IMAGE SEGMENTATION ON OXFORD IIIT (CATS AND DOGS) DATASET

Submitted by:

SHREYA BIRTHARE

16BIT0196

Under the guidance of

PROFESSOR SWETA BHATTACHARYA

ACKNOWLEDGEMENT

I sincerely thank our Chancellor - Dr. G. Viswanathan, VIT University, for giving me the opportunity to pursue my engineering at VIT. I thank Prof. Sweta Bhattacharya – School of Information Technology and Engineering, VIT Vellore, for giving me the opportunity to do this project and guide me throughout. Lastly, I also thank the Dean and HOD of School of Information Technology and Engineering (SITE), for their continued support and encouragement.

DECLARATION BY THE CANDIDATE

I hereby declare that the project report entitled “COMPARISON AND EFFICACY OF TRANSFER LEARNING MODELS FOR IMAGE SEGMENTATION ON OXFORD IIIT (CATS AND DOGS) DATASET” submitted by me to VIT Vellore in partial fulfilment of the requirement for the award of the degree of B.Tech is a record of capstone project work carried out by me under the guidance of Prof Sweta Bhattacharya. I further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Shreya Birthare
16BIT0196
Place: Vellore
Date: May, 2020

ABSTRACT

The advancement in technology has increased day by day and the major root of this advancement is machine learning/deep Learning and data science. These fields help researchers and industry experts to create models to solve very complex problem from different domains. For example, it solves image processing, natural language processing, robotics, textual related, and many other complex problems. These fields get matured because a lot of data is now available to train the models. However, heterogeneity and complex homogeneity relation exist between data making it difficult to train the model fast and with good accuracy.

Image processing is one of the vital problems that are solved by deep learning. For example, deep learning help to diagnose disease from images helps to recognize objects available in images, image captioning and labeling and many more. However, the accuracy of these models is still needed to increase. The researchers integrate different techniques to increase the accuracy of such models. Image segmentation is one of the major techniques which allows models to perform object recognition automatically with more accuracy. The complex nature of images even with the same object but each image is being captured at different angle making it difficult for models to differentiate. In such scenarios, the number of use cases has grown, and different use cases require to train models according to that. However, it is not a good approach and requires a lot of resources. Thus, image processing requires some automatic techniques to handle use cases problem. The research aims to understand and analyze if pre- trained model can be used in form of transfer learning to solve such tasks faster and better. The research proposes to analyze and understand the difference between the state-of-the-art segmentation technique used for the Oxford-IIIT Pet Dataset versus the usage of transfer learning models with the combination of different parameters such as random initialization, fine-tuning, and so on. The goal of the study is to find out if the trained model can be used instead of spending a lot of time developing a new deep learning model for particular use cases. To evaluate the performance of the proposed model, this research will plan to employ the UNET benchmark which is a supervised learning-based study. In addition, this research will also be going to try different methodologies to compare with the benchmark which includes purely supervised image segmentation, semi-supervised image segmentation, transfer learning, and possible hybrid combinations of the above techniques. Finally, this research will provide some main insights by comparing all models with the benchmark. Lastly, this research will also conclude time evaluation when automating the process for different use cases.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATION

DL	Deep Learning
AI	Artificial Intelligence
ML	Machine Learning
CNN	Convolutional neural network
DICOM	Digital imaging and Communications inMedicine
GAN	Generative adversarial network
ROI	Region of interest
MS COCO	Microsoft Common objects in context
SSF	Scale Space Filtering
MRF	Markov Random Fields
CRF	Conditional random fields

LIST OF TABLES

CHAPTER 1.

INTRODUCTION

1.1. Background of the Study

This section will provide some explanation of the background of the research

1.1.1. Image Segmentation

In a visual object understanding system, image segmentation is a vital employed technique. In digital image processing or computer vision, image segmentation is the process of partitioning a digital image into various image segments (Kaur & Kaur, 2014). The objective of segmentation is to streamline or potentially change the portrayal of an image into something more significant and simpler to investigate for an algorithm. Image segmentation is normally used to find objects and boundaries (lines, bends, and so on) in pictures. Technically, image segmentation is the process of assigning a label to each pixel in an image to such an extent thatpixels with a similar label share specific qualities.

Many useful applications are existing that are based on image segmentation (Forsyth & Ponce, 2002). Examples, healthcare-related images (Forsyth & Ponce, 2002) (i.e., extracting exact boundaries from cancer images), geographic information systems (Gonzalez, 2009), video processing (Friedman & Russell, 2013), and many others. Moreover, researchers have introduced different techniques for image segmentation. Following are the essential sorts of image segmentation techniques:

Thresholding Segmentation (Otsu, 1979): In digital image processing, thresholding is the most basic technique for segmenting images. The process of thresholding involves the comparison of every value (pixel intensity) in an image with the specified threshold. If the value of pixel intensity in the original image (x,y) is greater than the threshold, then the value in the final image is set to be the threshold, otherwise, it is unaltered. If the pixel in the input image passes our threshold check, then the value is set to 255. To segment specific colors from an image, we must apply the thresholding methods to specific pixel intensity values in an image channel. The

first argument is our original image or the image on which we wish to do thresholding. From agrayscale image, we can create binary images using thresholding.

Edge-Based Segmentation (Brejl & Sonka, 2000): In edge-based detection, you aim for the partial segmentation minimum, in which all the local edges can be clustered together in one binary image. Edge-based segmentation relies on edges found within the image by edge detector operators. It is a tow step process. The first step is to find the approximate location of the region of interest (ROI). The second step is to make borders around ROI using techniques like Houghtransformation.

Region-Based Segmentation (Nock & Nielsen, 2004): Region-based segmentation techniques look for smaller or larger blocks within the input image to be segmented. Region-based segmentation methods are preferred to edge-based segmentation methods in the case of noisy images. Region-based segmentation techniques involve the use of algorithms to segment the image by breaking it down into different components which share similar pixel characteristics. We are growing regions by recursively including adjacent pixels which are similar to the starting pixel and are related. If neighboring regions are similar, we combine them into one region and keep going, so we will construct a segmented region over the entire image. Region- Based Segmentation is the separation of one or more regions or objects within an image according to some criterion such as discontinuity or similarity. To segment an object out of the image, however, you require the boundaries of a region to be closed. A region in an image can be defined either by its border (edge) or by its interior, and these two are equivalent representations. Region-growing techniques are preferred in noisy images, in which detecting edges is very hard. Edge-based segmentation algorithms work by detecting edges in the image, according to the different discontinuities of gray levels, colors, textures, luminance, saturation,contrast, etc.

Clustering-Based Segmentation Algorithms (Dhanachandra et al., 2015): There are a lot of different types of studies done on image segmentation using clusters. Hybrid algorithms for clustering have also been proposed by many researchers and used to unravel the segmentation of images. We follow a bottom-up approach, meaning that we are allocating pixels closest to clusters. We attempt to cluster pixels that are closest to each other. By segmenting an image, we can make use of only important segments to process. Using a max-connected-domain

algorithm, we successively map over the initial image and successively segment overall targetimage objects.

Neural Networks for Segmentation (Milletari et al., 2016): Previous automatic segmentation work using deep learning has mostly focused on a pixel-wise, patches-based classification, which relies on a convolutional neural network (CNN) to generate an affinity map. Some of the increasing applications include autonomous vehicles, human-computer interaction, VR, etc. With the increasing popularity of deep learning over the past years, many semantic segmentation problems are being solved using deeper architectures, more commonly Convolutional Neural Networks, that outperform the other approaches in terms of precision andefficiency.

active contours (Kass et al., 1988): Active contours are a kind of segmentation technique that can be defined as using power forces and constraints to separate pixels of interest from an image for further processing and analysis. The active contour turns image segmentation problems into an energy-minimizing problem, by minimizing the integration of a curve's internal energy with its external energy. Active contour models, also called snakes, are a computer vision framework.

Graph cuts (Boykov et al., 2001): Graph cuts offer a clever method of image segmentation, which relies on turning an energy minimization problem into one that defines a maximal stream, or minimal slice, over a weighted graph with edges. Many such problems in energy minimization can be approximated by solving the problem of the maximum flow on a graph (and then, according to the max-flow minimum-cut theorem, by defining a minimum cut of a graph). the algorithm of max-flow/min-cut can be used for finding the optimal cut, and hence, an optimal solution to the segmentation problem. So, the max-flow algorithm is executed over a network flow graph to find a min-cut that produces the optimal segmentation. For example, the max-flow/min-cut algorithms can be used for making photo-editing, and also segmenting medical images.

conditional and Markov random fields (Plath et al., 2009): A conditional random field is a discriminative statistical modeling technique used when class labels of various inputs are not independent. Conditional random fields (CRFs) are a class of statistical modeling methods often

applied to pattern recognition and machine learning and used to make structured predictions. One of the classic models of optimization in image segmentation is the well-known Markov random field model (MRF). A hybrid technique that combines Scale Space Filtering (SSF) andMarkov Random Fields (MRF) to segment the color images.

sparsity-based (Starck et al., 2005): The segmentation algorithm, based on Morphology, develops binary segmentation masks which precisely divide the composite image into various layers, like the background and the foreground layers. It is a sparsity-based method to remove additive white Gaussian noise from a given image. It removes additive white Gaussian noise. The foreground region of an image is divided into three clusters of different features, which are generated by the feature text, graphics, and overlapped clusters. This type of image segmentation is really good to work with images having texts.These techniques are being used by researchers and the industry for different tasks such as image enhancement, object recognition, and computer vision.

1.1.2. Deep Learning

Over the past few years, however, deep learning (DL) models have yielded a new generation of image segmentation models with remarkable performance improvements — often achieving the highest accuracy rates on popular benchmarks — resulting in a paradigm shift in the field. Deep learning is a subset of machine learning which gives the capacity to machines to perform human-like tasks without human contribution. It gives the capacity to an artificial intelligence specialist to impersonate the human brain. Deep learning is aiding to get more advancement in solving complex problems that were unable to solve by simple artificial intelligence. Deep learning is carried out through neural network engineering subsequently likewise called a deep neural organization. It is a very generic approach thus, can be applied to many areas of science and technology. For example, deep learning proposed such algorithms that beat existing AI/ML based-based image recognition and speech recognition algorithms (Tompson et al., 2014) (Farabet et al., 2012) (Sainath et al., 2013).

Deep learning is providing backbone support to various applications (Deng, 2014) including audio and speech-related applications, image, video, and multimodality applications, language modeling, information retrieval applications, and natural language processing applications. However, the drawback of deep learning is that it requires heaps of information with loads of computational power.

1.1.3. Transfer Learning

Machine learning and data mining already help researchers and developers in examining knowledge engineering with significant success in terms of classification, clustering, and pattern recognition(Wu et al., 2008). These algorithms are works without any flaw when the source of data extraction remains common. However, these algorithms perform inconsistently when the data extraction source changes. This requires developing a new algorithm according to the data source which is a hectic, expensive, and time-consuming approach (Pan & Yang, 2009). In such cases, transfer learning is an approach that can help map the existing model with a new source of data. Different applications of transfer learning include web document engineering (Fung et al., 2005), outdated data engineering (Pan et al., 2008), and sentiment engineering (Blitzer et al., 2007). Transfer learning is a machine learning strategy where a model created for an undertaking is reused as the beginning stage for a model on a subsequent task (Torrey & Shavlik, 2010). It is a well-known approach in deep learning where pre-trained models are utilized as the beginning stage of computer vision and natural language processing assignments given the tremendous figure and time assets expected to foster neural organization models on these issues.

1.2. Aim and Objectives

This research aims to attempt to review and use transfer learning for semantic image segmentation on the Oxford-IIIT Pet dataset. Further, we also hope to compare and review if and how learning and initialization affect performance across different tasks - to discover if and how sharing learned parameters across different but related tasks is beneficial.

Based on the aim of the study, the below objectives are formulated:

- Build a basic transfer learning neural network using RESNET50 and compare its efficacy with the supervised segmentation benchmark over Oxford-IIIT Pet Dataset
- Build a simple decoder transfer learning neural network using RESNET50 and compare its efficacy with the supervised segmentation benchmark over Oxford-IIIT Pet Dataset
- Build a deeper decoder transfer learning neural network using RESNET50 and compare its efficacy with the supervised segmentation benchmark over Oxford-IIIT Pet Dataset
- Build a transfer learning neural network using RESNET50 with the involvement

of RESNET50 in training and compare its efficacy with the supervised segmentation benchmark over Oxford-IIIT Pet Dataset

- Build a deep dense decoder transfer learning neural network using RESNET50 and compare its efficacy with the supervised segmentation benchmark over Oxford-IIIT PetDataset
- Build a deep dense decoder transfer learning neural network using RESNET50 with random initializations and compare its efficacy with the supervised segmentationbenchmark over Oxford-IIIT Pet Dataset
- Build a deep dense decode transfer learning neural network using RESNET150V2 and compare its efficacy with the supervised segmentation benchmark over Oxford-IIIT PetDataset

1.3. Research Questions

The below research questions will help us to deep dive into this research a little further

- Does a simple transfer learning model (resnet50) perform better than the supervised segmentation benchmark?
- Does a transfer learning model (RESNET50 + simple decoder) model perform better than a supervised segmentation benchmark?
- Does a transfer learning model (RESNET50+ deeper decoder) perform better than the supervised segmentation benchmark?
- Does a transfer learning model (RESNET50 + involvement of RESNET in Training) perform better than the supervised segmentation benchmark?
- Does a transfer learning model (RESNET50 + deeper dense decoder) perform better than the supervised segmentation benchmark?
- Does a transfer learning model (RESNET50 with random initializations + deep dense decoder) perform better than the supervised segmentation benchmark?
- Does a transfer learning model (RESNET150 V2 + deeper dense decoder) perform better than the supervised segmentation benchmark?
- If the proposed transfer learning models are near the supervised segmentation benchmark, then how faster/slower is the training process on average than the benchmark time?

1.4. Research significance

The proposed research will have significance in both Industry and Academia. However, it has more focus on academics. As presented research has provided a technique based on transfer learning, which will make the algorithms able to work on unseen problems. The

existing related research does not fully incorporate deep learning and transfer learning support. Whereas the proposed research will plan to use segmentation and other techniques. In addition, it will also utilize the ResNet-50 and ResNet-150 to will find out the extent of complexity required by the algorithm. This will open new doors of research. On the other hand, it will help the industry to take pictures of different fields (medical, detection, HCI, etc.) accordingly.

1.5. Scope of the Study

Due to the limitation of time frame, the scope of the research will be limited as below:

- The data from the research is taken from Kaggle. And directly imported without any interference.
- The research will include the development and evaluation of the RESNET neural network with some variations and its comparison with UNET. Ottransferssfer learning neural networks such as VGG, Inception, Mobile Net, etc are not considered due to lack of resource and time.

CHAPTER 2. LITERATURE REVIEW

2.1. Related Research

Deep learning is a subfield of machine learning and artificial intelligence. Deep learning-based algorithms solve complex problems (naturally related to humans) which are unable to solve by machine learning and artificial intelligence. Pramerdorfer and Kampel (Pramerdorfer & Kampel, 2016) have proposed a Convolutional Neural Network (CNN) based model which performs facial expression recognition. In addition, their study also provides insight into why existing related models perform inconsistently and what are the main factors and bottlenecks that impact the algorithm's performance. This study has utilized the FER2013 dataset (Goodfellow et al., 2013) which is 48x48 pixel grayscale 28,709 images of faces with annotated pixels. It also has a unique annotation of expression per image. The proposed model achieved significant performance. However, the research identified bottlenecks but did not provide solutions to overcome the bottlenecks.

Zhang et al. (Zhang et al., 2015) proposed a CNN-based framework to perform cross-scene crowd counting. Existing related solutions are scenes specific, which means; they perform significantly in the scene they are learned from. However, failed with the new and unseen scene and requires retraining according to this new scene. The proposed research performs a sort of fine-tuning on the existing CNN-based model. To train the model, a new dataset which is based on 108 different scenes and 200, 000 head annotations has been constructed. The image has 480x640 pixels at 3 channels images from a single camera taken from a video stream. Each image has a different number of people, and the number is annotated with the image. The result shows significant performance when they evaluate their model with three different datasets that

include WorldExpo'10 crowd counting, the UCSD pedestrian (Chan et al., 2008), and the UCFCC 50 dataset (Chan et al., 2008).

Shrestha et al. (Shrestha et al., 2015)proposed research that detects workers who have not applied safety measures on the job sites. Specifically, it works on helmet recognition during work and if anybody does not wear it then it annotates with speakers and mobile notification. The proposed research has utilized its developed dataset. Which is around 2000 hat images. The dataset is labeled with bounding box annotations The study has proposed a significant contribution to research. However, they only focus on helmets rather than all the safety measures that include boots, vests, gloves, etc.

Jetley et al. (Jetley et al., 2018) proposed an attention model which is based on end-to-end trainable for CNN-like architecture. It performs image classification by taking a 2D feature vector as an input and outputting a 2D matrices-based vector. The model has also been applied to pipelines at different stages of the Convolutional Neural Network. The research has used the CUB200-2011 dataset to train their model. To evaluate the model, the proposed research has utilized three unseen datasets and achieved significant results. However, segmentation-based results may not achieve the same accuracy on the non-segmented dataset.

Mamta Mittal et al. (Mamta Mittal et al., 2019) proposed a deep learning-based model for brain tumor detection which employed MRI (Magnetic Resonance Imaging) images as a dataset. In addition, the proposed research also uses image segmentation techniques to automate tumor detection. This research has used Elucidated image segmentation technique. The result of the research shows significant performance in terms of different evaluation measures.

Kim and Jong (Kim and Jong, 2019) proposed a loss function that is based on the SoftMax layer of the deep learning algorithm. This loss function aims to increase the performance of the image segmentation algorithm. The existing such works are mostly based on supervised learning. However, the proposed loss function is based on semi-supervised and unsupervised learning. The result shows the significant performance of the proposed work.

Van et al. (Van et al., 2019) proposed an image segmentation approach that is based on medical imaging. The existing approaches are mostly based on supervised learning, which requires a lot of labeling data to train the model. However, the proposed approach has

employed transfer learning methods to boost the performance of the model. The underlying study proposed 4 different classification methods with limited training data. All four algorithms show significant performance when tested with brain images by performing segmentation. In addition, the proposed work also compares these algorithms with each other to examine the outperformanceof the model.

Opbroek et al. (Opbroek et al., 2018) proposed a novel method for image weighting by using kernel learning. Kernel learning is a famous mechanism to reduce the difference between training and testing data and help to put the suitable values for kernel learning that are consequently used for weighting. The proposed work also employed Maximum Mean Discrepancy (MMD) with training data and testing data to get an efficient weighting of images. Maximum Mean Discrepancy also helps them to get test the model by joint optimizations of image weights along with the kernel. The result shows proposed technique has better segmentation when performing experiments on brain tissue images. In addition, the proposed method also performs better with heterogeneous data. Moreover, the proposed model also has significant performance in assigning weights.

Shaha and Pawar (Shaha and Pawar, 2018) proposed a transfer learning approach for object recognition or image classification which is based on the VGG19 pre-trained model. The proposed work integrated the Convolutional Neural Network (CNN) with VGG19 to make the transfer learning. The study employed GHIM10K and CaITech256 datasets to train the model. The study compared their work with (VGG19) with state-of-the-art approaches that include AlexNet and VGG16. Then in part of CNN architecture, the work performs a comparison with the Support Vector Machine (SVM) algorithm. The result shows significant performance (when using VGG19 with CNN) when measured with precision, recall, and F1 score.

CHAPTER 3. METHODOLOGY

The overall idea of the Methodology for this project is to implement transfer learning using the Resnet model in terms of transfer learning. The recent model has been extensively trained on the data set which gives additional superpower to the model that the model used to understand the basic features present in any image. When a new image that is not similar to the images used in training the data set is passed inside the model along with trainable parameters the model adapts itself and captures the features of the image and enforces a mechanism to train the new layers which work as a decoder. In this vein, the transfer learning model itself works as an encoder.

3.1. Dataset Description

The selected dataset for underlying research is named "The Oxford-IIIT Pet Dataset". It is a public dataset (Jawahar et al., 2012) whithat available on the Kaggle website. The dataset has a total of 7349 files of 37 diverse breeds of cats and dogs with 12 categories of cats and 25 categories of dogs. Each category has about 200 images. All these images were taken from a different angle with a different poses, different shapes, and different light intensities. This research has split the dataset into 3680 for training and 3669 for testing from a total of 7349 images.

Some sample images with their annotations are given in the figures below:

Figure 1: Example of Image and segmentation map present in Dataset

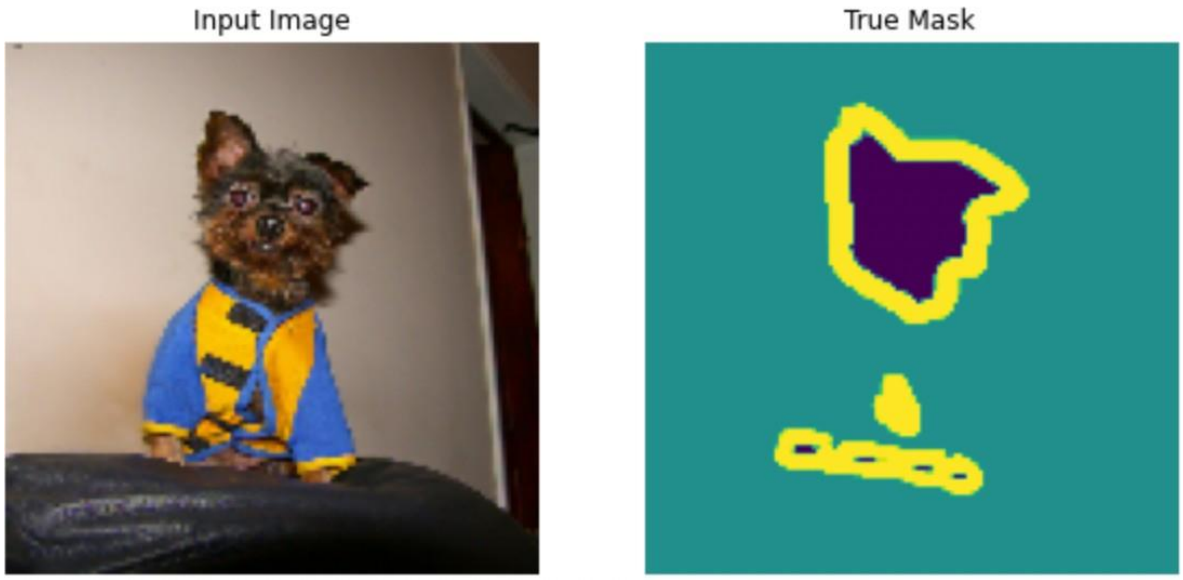

Figure 2: Example of Image and segmentation map present in Dataset

3.2. Data pre-processing

Data preprocessing is an essential part of data refinement. It is used to clean the data, deal with missing data, remove abnormalities, etc. This research has employed only one preprocessing technique that which is Normalization. The following sections contain some explanations aboutNormalization

Normalization is a preprocessing technique that can map or balance the range from existing diverse ranges. Especially, when need to perform some prediction we need to manage the large variations of output closer which is done by using the normalization technique (Patro & Sahu, 2015). This research has utilized normalization to balance the color range of each pixel into a specific range. It will help our model to understand the images easily. For example, the images that have 3 channels (RGB) ranging from 0–255pixel intensity. we divide all intensities by 255so that the values come between 0-1.

3.3. Data Augmentation

It is a data science technique that helps researchers to produce more data for their research. Data augmentation is the most employed technique when you have an unbalanced dataset. Data augmentation provides different techniques to manage unbalanced data, that includes Over

sampling, Undersampling, and SMOT (Synthetic Minority Oversampling Technique). The unbalanced dataset can lead the trained model towards the model overfit and model underfit. By considering these effects, this research aims to perform data augmentation by flipping the train images and masks so that we get more data to train on.

3.4. Adam Optimizer

Adam optimizer is one of the most usable optimizers in deep learning algorithms. It can help to deal with the sparse gradients on noisy data. This research will use this optimizer with a combination of 2 gradient descent methods, as stated below:

1) Momentum: This method of Adam considers the "exponential weighted average" whichallows the algorithm to move towards the minima at a faster pace.
2) Root Mean Square Propagation (RMPS): This method of Adam uses adaptive learning that tries to improve optimization using "exponential moving average" instead of cumulating the squared gradients.

3.5. Sparse Categorical Cross Entropy

This is the loss function used for many deep learning-based models in the research, this loss function is used when we have more than 2 label classes. For the segmentation problem, this is suitable as all images will be having consistent label maps so that we can have a loss function on every pixel-by-pixel mapping. Sparse Categorical Cross entropy is highly useful for segmentation tasks.

3.6. Evaluation Metrics

The evaluation metric is the standard accuracy based on the output of Sparse Categorical cross-entropy.

3.7. Model Components

Figure 2 illustrates the major components of the proposed model. The explanation of each component is given below.

3.8. Resnet-50

ResNet-50 is a residual network which is a convolutional neural network that consists of deep 50 layers. The simple transfer learning model will allow us to understand how far ResNet-50

can solve the problem without any additional layers. In underlying research, it will help us to measure how much complexity is required to solve the underlying problem using transfer learning.

Any function can be represented with a single-layer feedforward network. The layer could, however, be very large, and the network is prone to overfitting the data. As a result, there is a consensus among researchers that our network architecture has to be more complex.

Since AlexNet, the cutting-edge CNN architecture has become increasingly complex. The VGG network [3] and GoogleNet (also known as Inception v1) have 19 and 22 convolutional layersin comparison to AlexNet's meager 5 layers.

However, just adding layers on top of one another does not increase network depth. Because of the well-known vanishing gradient problem, it is challenging to train deep networks because, as the gradient is back-propagated to older layers, repetitive multiplication may lead the gradient to become infinitesimally small. As a result, the network's performance becomes saturated or even starts to decline quickly as it penetrates further.
Before ResNet, there had been several approaches to cope with the vanishing gradient problem, like adding an auxiliary loss in a middle layer as additional supervision, but none managed to address the issue permanently.

As seen in the accompanying picture, the fundamental concept behind ResNet is the introduction of an "identity shortcut connection" that omits one or more layers:

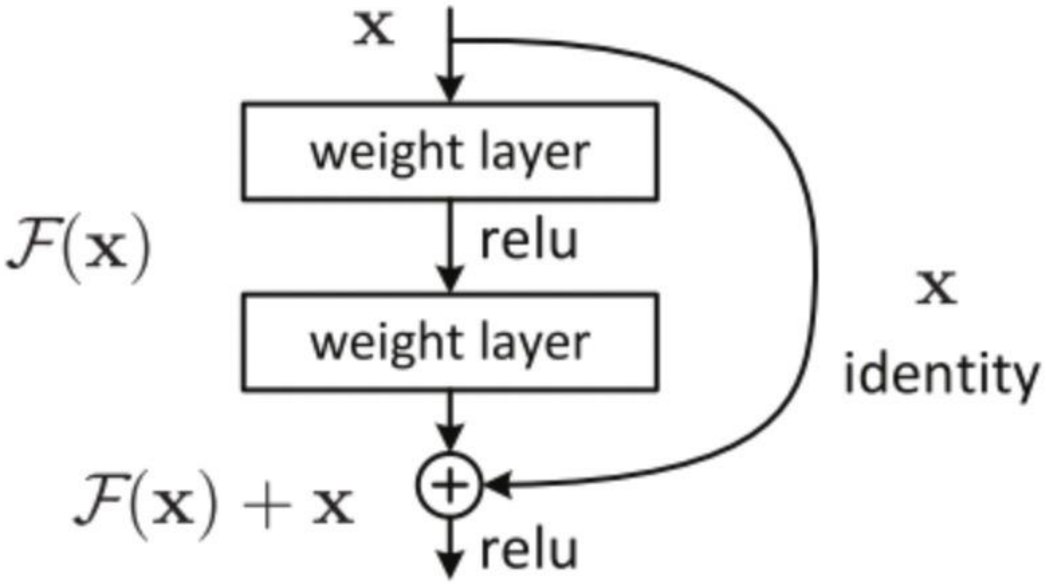

Figure 3: Image representation of skip connecting in ResNet

As long as identity mappings (a layer that accomplishes nothing) are stacked on top of the existing network, the performance of the final architecture shouldn't be affected, according to the authors of ResNet. This suggests that the training error of the deeper model shouldn't be greater than that of its shallower equivalents. They contend that it is simpler to allow the stacked layers to fit a residual mapping rather than the desired underlying mapping. And the remaining block mentioned above expressly permits it to do just that.

3.9. Encoder Network

Encoder decoder networks are sequential models which try to bottleneck the neural network and gain the maximum information because of the bottleneck. The encoder network vector is the input data in an embedded manner such that the embedded vector retains maximum information from the input data and converts this input into some different representation. Whereas, the decoder network is trained in such a way that it captures the necessary information from the embedded vector (or the specific representation converted by the encoder) and transforms the information into the desired output.

In the transfer learning model, the layers are pruned off from the end and the weights freeze when a new image is processed through the model the endless of the model behaves as encodersand the new dense layers which are trainable, behave as a decoder.

3.10. Decoder Networks

Decoder networks are used to solve the sequence-to-sequence problem in neural networks. In addition, it works as a bottleneck in a neural network which enforces the network to understand most of the data which is present in training. In the context of this research, a simple decoder network along with transfer learning will give us a little better understanding of how the transfer learning network will behave with additional complexity.

3.11. Deep Dense Decoder

The deep dense decoder helps us to understand the features of data and predicts the prostate by using the grouped convolution (To et al., 2018). This research will employ the deep dense decoder because the combination of the deep dense decoder and transfer learning will give the model additional complexity to understand the data in terms of generalization and customization for the problem.

3.12. Random Initialising

Generally, random initialization is the process of assigning random weights to machine learning algorithms to reduce the loss. Underlying research will initialize the transfer learning network with random weights which will allow us to check the additional use cases where the network can converge into some other minimum network. Which maybe be lesser than the minimums from present complexities. This will help us in making sure that all the models present in research are converging into a global minimum and are optimal. This will also allow us to understand if the ResNet-50 architecture is good for image segmentation or not.

3.13. ResNet-150

ResNet-150 is an upgraded model of ResNet-50 that consists of 150 deep layers of CNN. By using the ResNet-150 V2 model we can add additional complexity to ResNet 50. As the name suggests ResNet 150 V2 is three times more complex than ResNet-50. By using ResNet-150 V2 we will be able to understand if this problem requires more complexity in ResNet-50 or to what extent fine tuning requires in ResNet 50 models.

3.14. Transfer Learning

The goal of transfer learning is to transfer information from one domain to another. It is a machine learning process that entails applying features discovered while solving the source

problem to a different but related target problem. Transfer learning was first conceptualized in educational psychology. For instance, since both a bicycle and a motorcycle are two-wheelers and both require some common knowledge, someone who has mastered riding a bicycle will likely find learning to ride a motorcycle easier than others.

In Figure 2.2, some understandable illustrations of transfer learning are presented. The goal of transfer learning is to use the human capacity to transfer information from one subject to anotheras its source of inspiration.

In the idea of machine learning, independently and uniformly distributed (i.i.d). For training datasets, the presumption that the training and test sets of data must match is frequently made. Transfer learning relaxes this premise. To combat the issue of insufficient training data, this serves as motivation. The model in the target domain does not need to be trained from scratch because the training data and test data do not need to be i.i.d. for transfer learning. As a result, the need for large amounts of training data is greatly reduced, as is the cost of computing for training in the intended domain. This chapter discusses the premise and implementation of themodels.

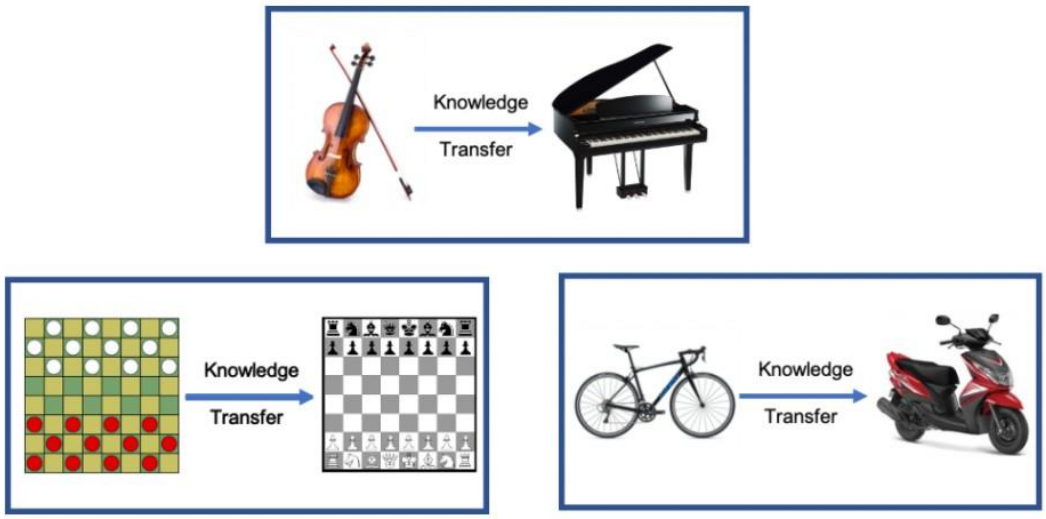

Figure 4:Idea of knowledge transfer

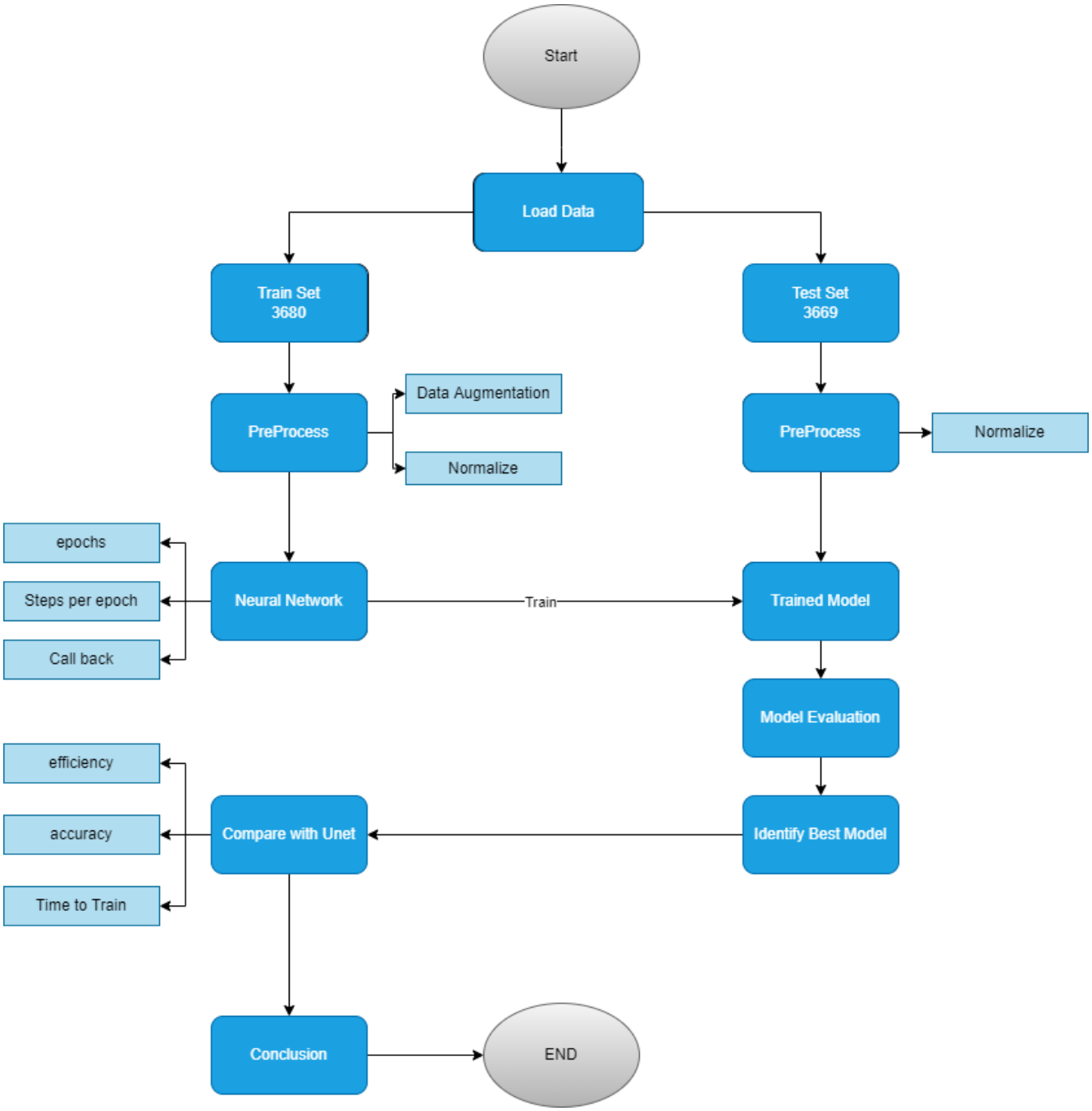

Figure 5:Architecture of the proposed

Figure 1 illustrates the complete architecture of the proposed research. For example, it explains the dataset division, type of preprocessing applied, parameters while training the model, modelevaluation, and identifies the best performing model among all.

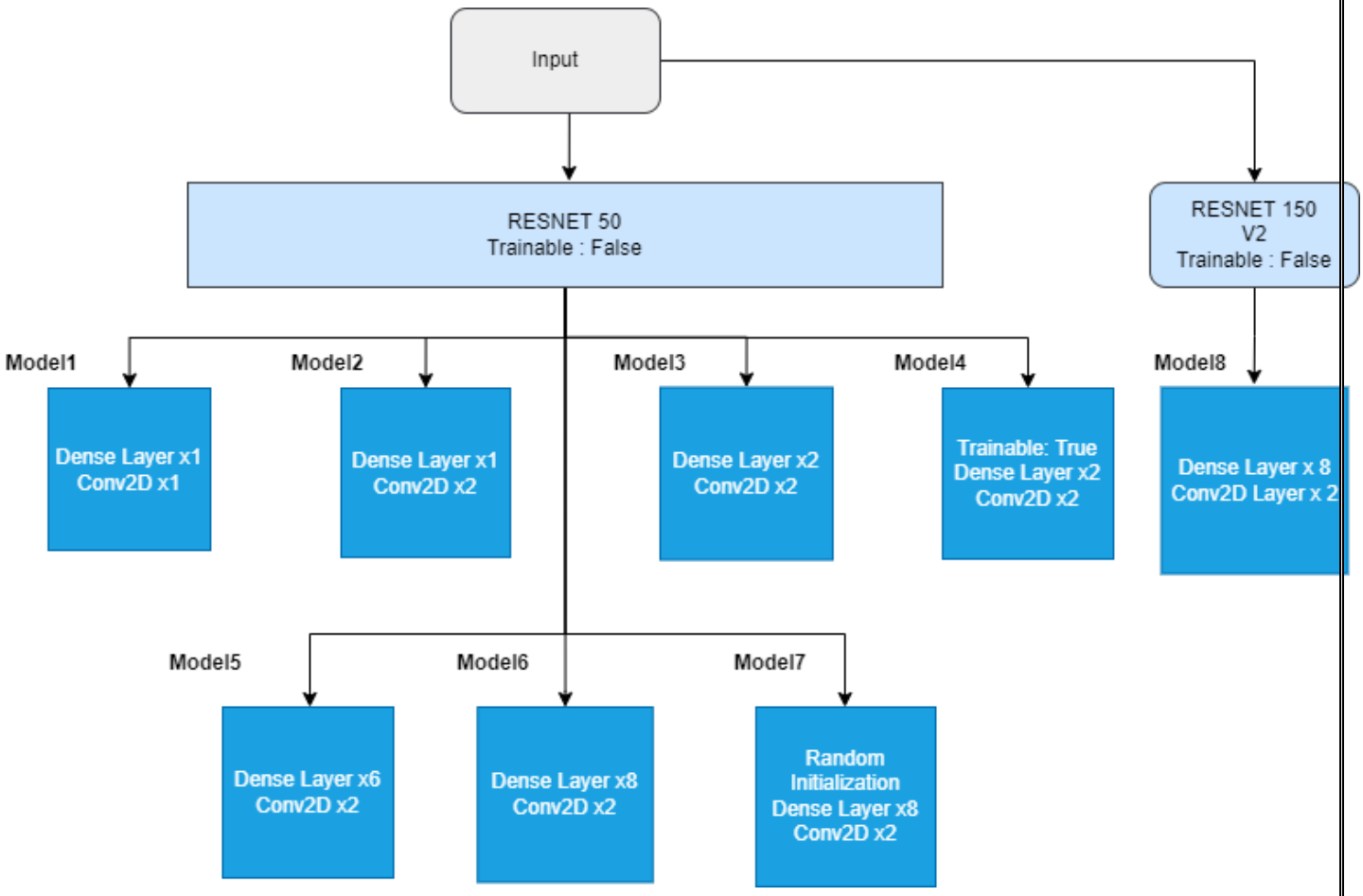

Figure 6: Neural network architectures

CHAPTER 4. IMPLEMENTATION

4.1. Introduction

The basic idea of transfer learning for the Premises is to set up a decoder network after the ResNet network. The idea here is to use the pre-trained weights of ResNet and the decoder willget trained in the training process to adapt the data set and update its weight.

The first few models are very simple, they are not meant to solve the problem of segmentation, but these models will help us to understand how much complexity is required to solve the segmentation problem and is transfer learning effective for solving the segmentation problem. The objective of the deeper and denser model is to optimize and decode the outputs from the pre-trained ResNet, the following network will always be ended by a convolution neural net which will be the last layers of the decoder so that we get a segmentation map for the inputs.

The data set to use to pre-train the ResNet is image net which is very much similar to the Oxford-IIIT pet dataset. This means that the output from the ResNet is very much beneficial for the model and the decoders will decode the output using backpropagation and try to convertthe ResNet output into a segmentation map.

Where using decoder models from model-1 to model-8, linearly increases their complexity that is adding more layers. The idea of implementing such a thing is to understand which network is underfitting and which network is overfitting which intern will give us the best model for thisparticular problem.

In the linear progression of complexities of the model we have also added some combinations where we will be training the ResNet network along with the decoder network, not training the ResNet network at all, and initializing the neural network with random weights. The last model will use ResNet 150 instead of ResNet 50 to increase the complexity. If the results from ResNet 50 and ResNet 150 are the same that means the complexity of the ResNet 50 model is enough to solve this problem. There can be some chances where ResNet 150 might reduce the validation and testing accuracy due to overfitting. All the models except model-8 use ResNet 50.

For each model, the input is an image in 3 channels with dimensions (128,128,3) and the output is a segmentation map of the same dimensions (128,128,3). ResNet model will flatten the input into a long vector of length 2048 (both ResNet 50 and ResNet 150) this part of the neural network act as the encoder. Then there are combinations of our Different types of layers, Dense layer(s) and convolutional neural network layer(s). The combination part where it expands the vector of length 2048 to a 3D vector of dimension (128,128,3) is the encoder of our network.

The above-mentioned premise makes all of our models Encoder-Decoder neural networks which are using Transfer Learning technique to solve the Image Segmentation problem.

4.2. Model-1

Model-1 is a very basic model with one dense layer and one convolution layer. The idea behind creating this model is to understand and verify if the model is under-fitted. Model giving good accuracy employees the pretraintrain ResNet is sufficient for segmentation problem and only requires additional fine-tuning. Having bad accuracy implies the model is underfitting and can be improvised by adding complexity to the decoder network. We also will be able to understand the complexity of ResNet concerning the segmentation problem for the Oxford-IIIT pet dataset.

The architecture of the model is given in Fig :

The model is sequential after the ResNet network and comprises one fully connected dense layer and one convolutional layer to get output as an image (segmentation map), the ResNet layer is not trainable

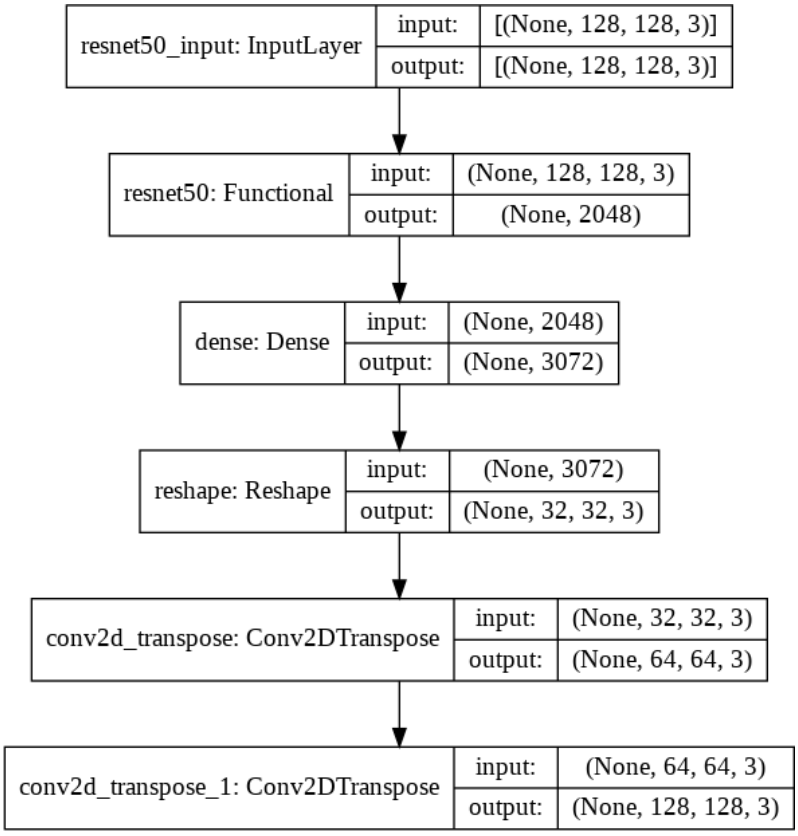

Figure 7: Architecture of Model-1

4.3. Model-2

Model-2 adds an extra step of complexity to model-1 by adding a convolutional neural network layer(with more number of nodes). The idea behind model-2 is to increase complexity and verify the behavior of the results to understand the nature of our decoder.

The architecture of the model is given in Fig :

The model is sequential after the ResNet network and comprises one fully connected dense layer and two convolutional layers to get output as an image (segmentation map), the ResNet layer is not trainable

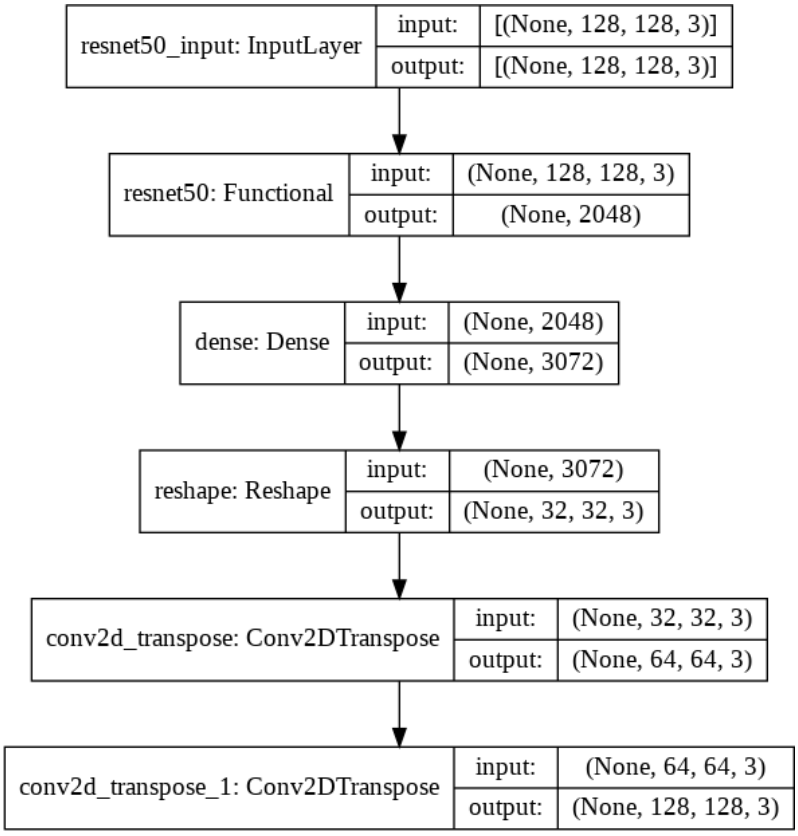

Figure 8: Architecture of Model-2

4.4. Model-3

Model-3 adds an extra step of complexity to model-2 by adding a fully connected dense layer (with more number of nodes) to the encoder.

The architecture of the model is given in Fig :

The model is sequential after the ResNet network and comprises two fully connected dense layer layers and two convolutional layers to get output as an image (segmentation map), the ResNet layer is not trainable

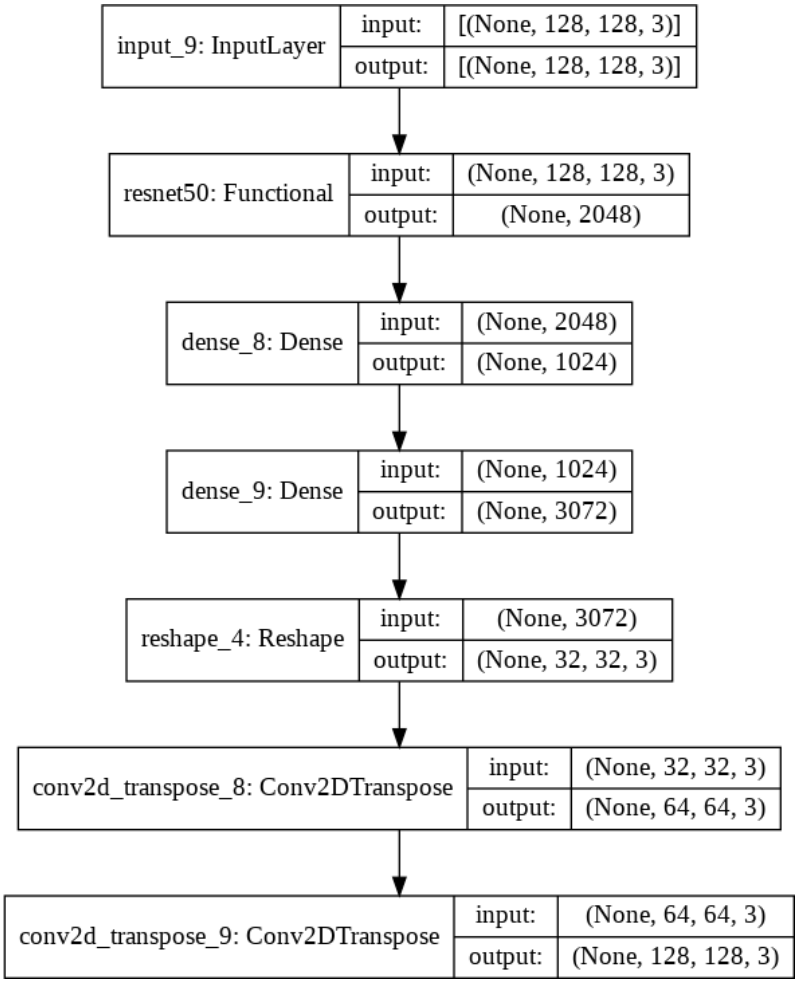

Figure 9:Architecture of Model-3

4.5. Model-4

Model-4 adds an extra step of complexity to model-3 by adding a fully connected dense layer (with more number of nodes) to the encoder. And this model also trains the Resnet further from the pretrained Weights. Training the Resnet can give us some hints about the Efficacy of ResNetarchitecture for the image segmentation problem.

The architecture of the model is given in Fig :

The model is sequential after the ResNet network and comprises o fully connected dense layer layers and two convolutional layers to get output as ian mage (segmentation map), the ResNetlayer is trainable.

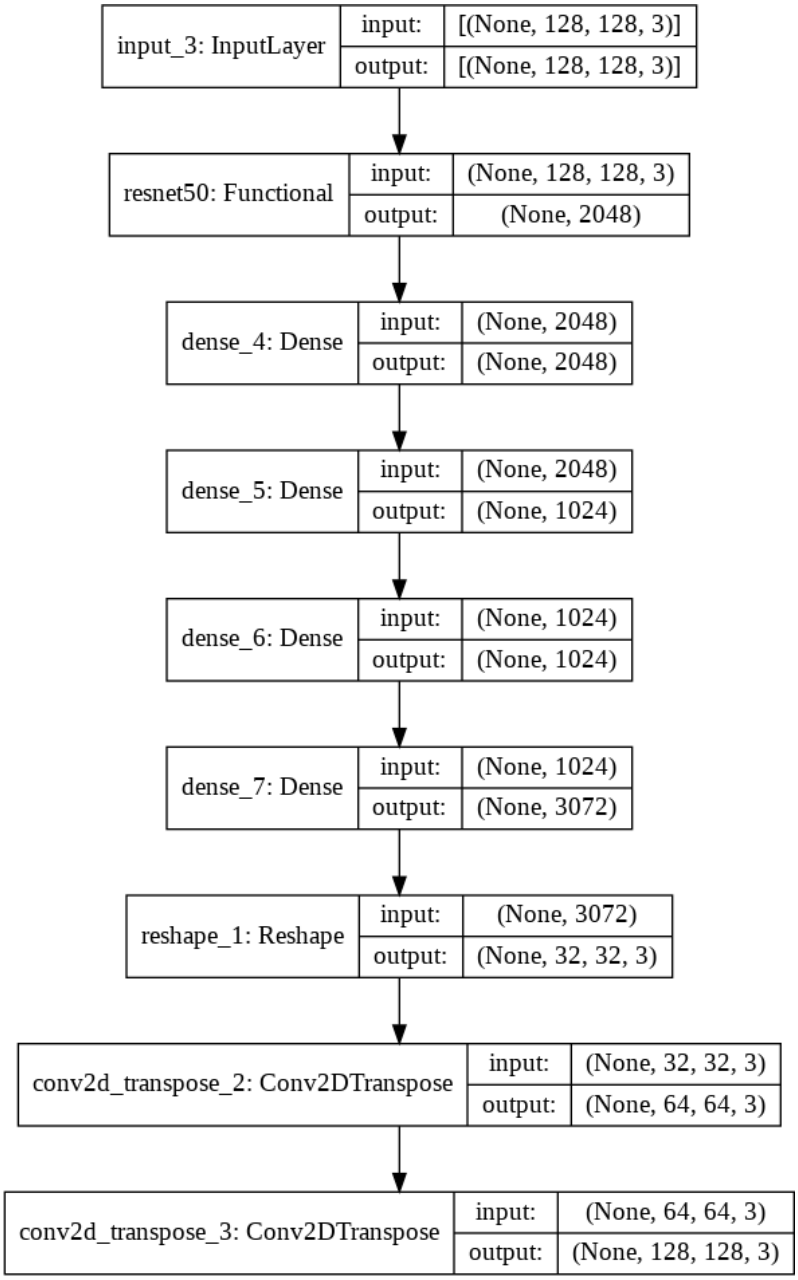

Figure 10: Architecture of Model-4

4.6. Model-5

Model-5 has six fully connected Dense layers (with an incremental number of nodes in each layer) followed by two convolutional neural net layers. The encoder used for transfer learning here is ResNet 50.

The architecture of the model is given in Fig :

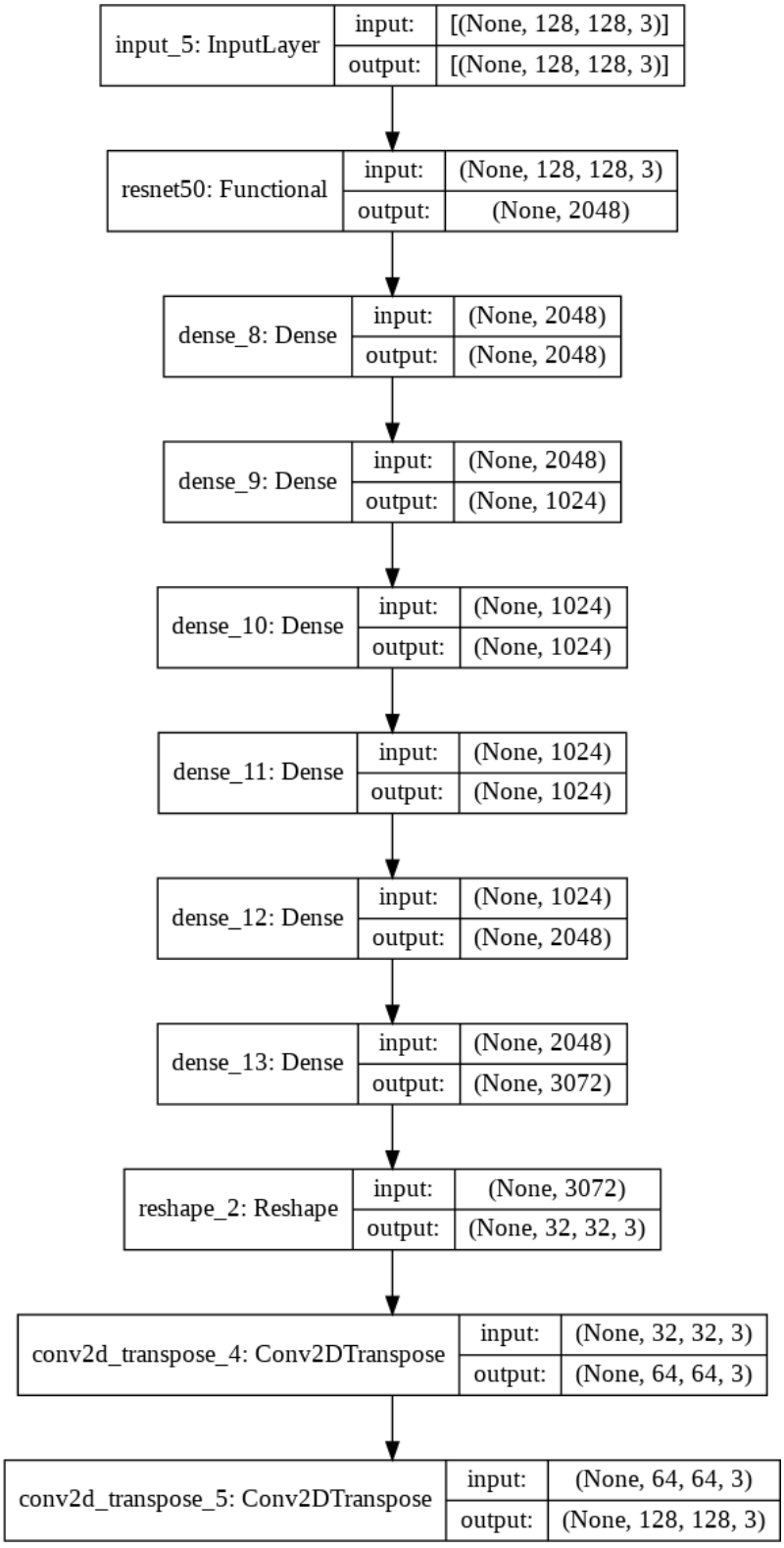

Figure 11: Architecture of Model-5

4.7. Model-6

Model-6 has eight fully connected Dense layers (with an incremental number of nodes in each layer) followed by two convolutional neural net layers. The encoder used for transfer learning here is ResNet 50.

The architecture of the model is given in Fig :

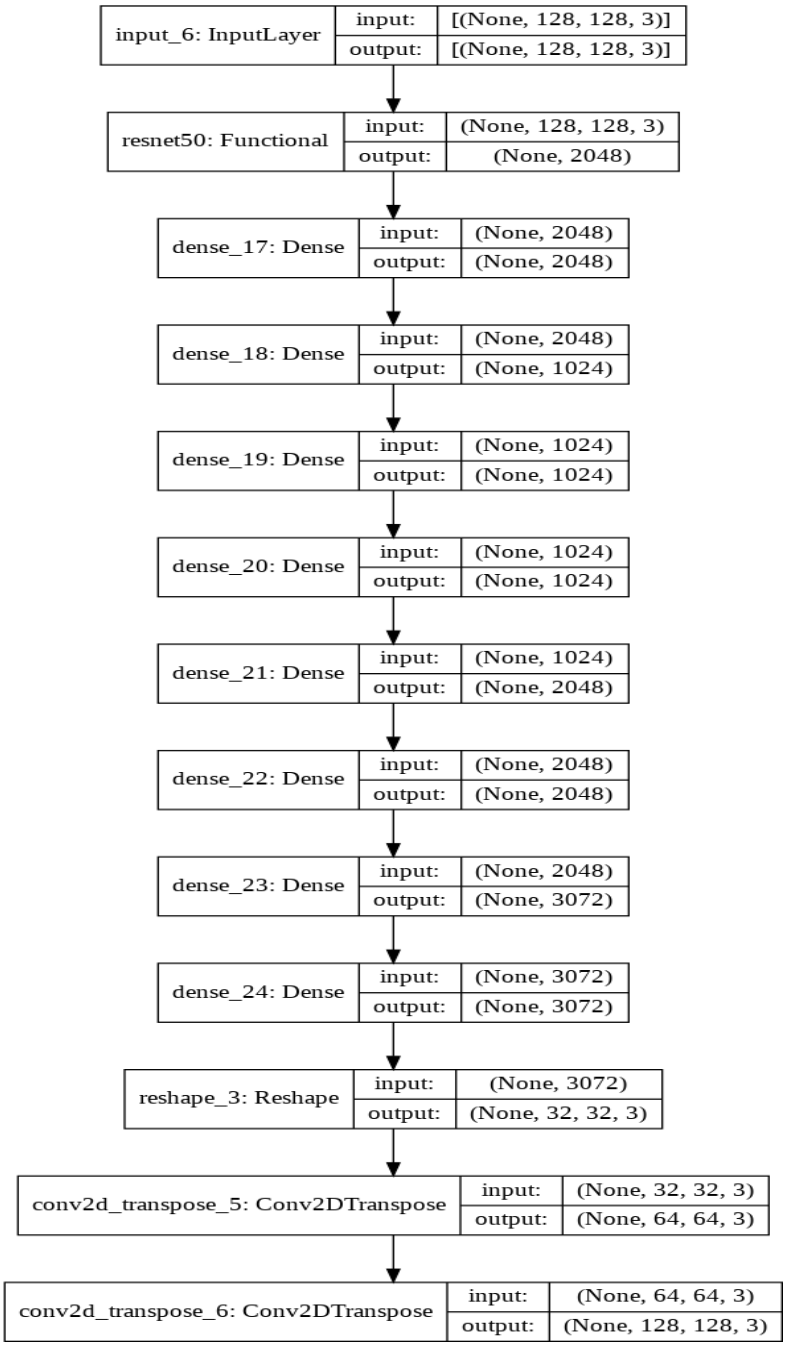

Figure 12: Architecture of Model-6

4.8. Model-7

Model-7 has eight fully connected Dense layers (with an incremental number of nodes in each layer) followed by two convolutional neural net layers. The encoder used for transfer learning here is ResNet 50.

The architecture of the model is given in Fig :

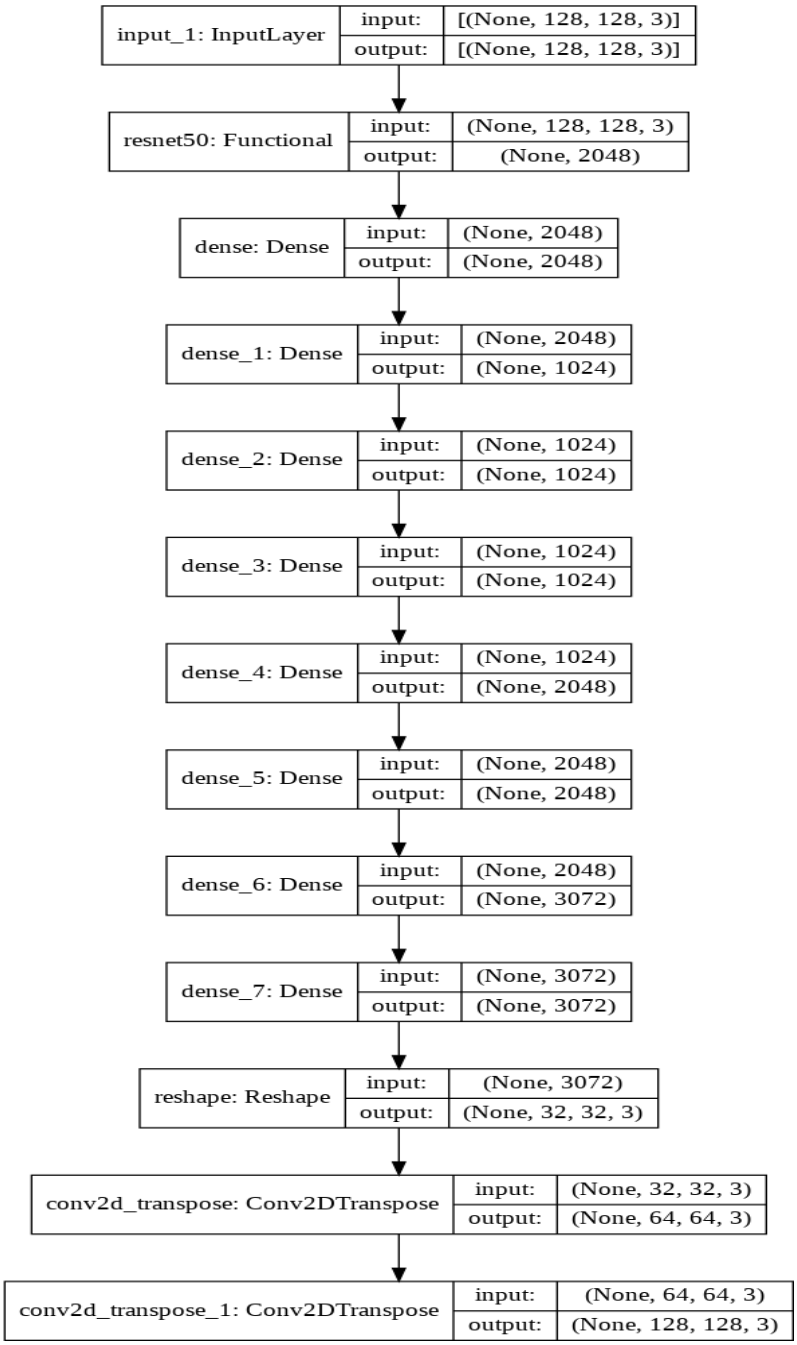

Figure 13: Architecture of Model-7

4.9. Model-8

Model-8 has eight fully connected Dense layers (with an incremental number of nodes in each layer) followed by two convolutional neural net layers. The encoder used for transfer learning here is ResNet 150.

The architecture of the model is given in Fig :

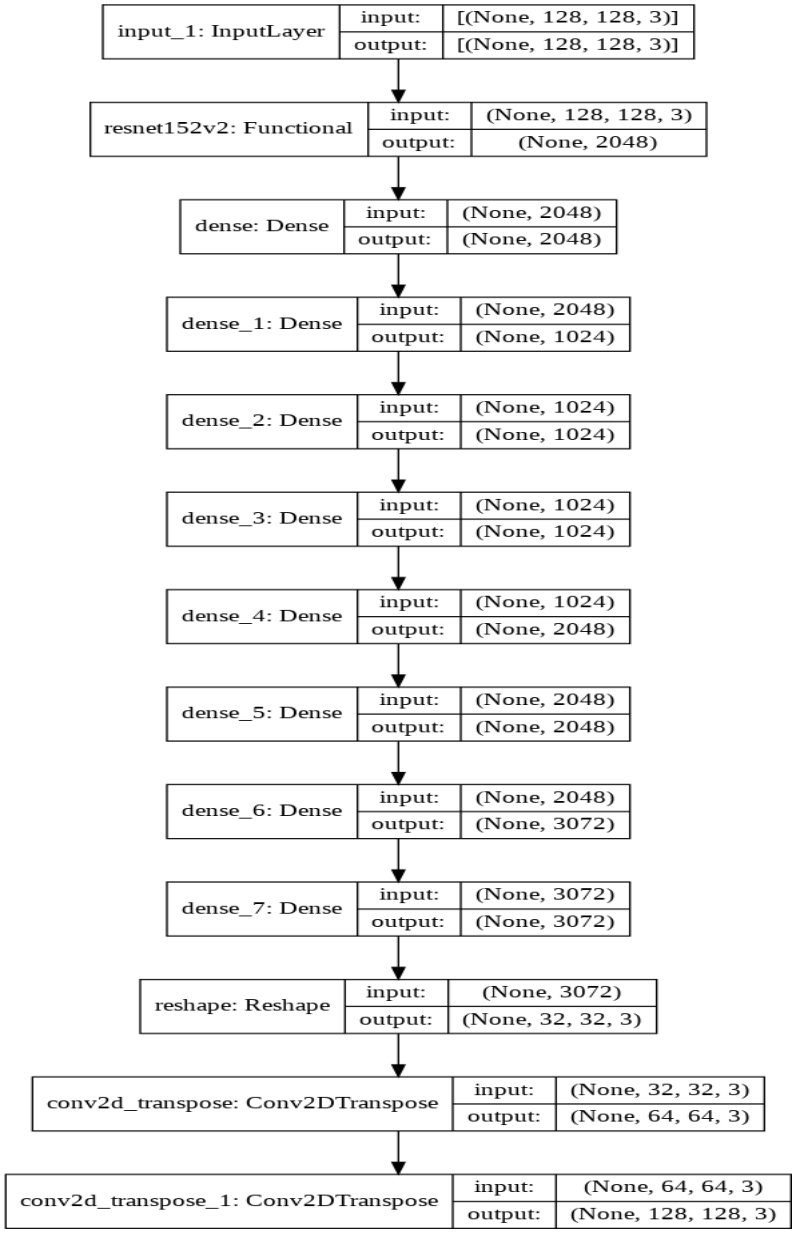

Figure 14: Architecture of Model-8

4.10. Summary

The baseline model from which we will be comparing our Transfer Learning models is UNet with the last layers giving output in dimensions of (128,128,3). For the training of the UNet model, there are no restrictions on the number of epochs (training was stopped when the difference between validation accuracy and training accuracy became insignificant), always the best weights with the best accuracy are saved, and all the layers are trained, no layer is kept frozen while training the network, except using usual dropout layer which randomizes nodes freezing in a layer.

We will take a look at results discussions and analysis of all the 8 mentioned modes and comparison of UNet (The baseline model) in terms of accuracy, efficacy, and training time in the upcoming Chapter 5 and Chapter 6.

CHAPTER 5. RESULT AND DISCUSSIONS

5.1. Introduction

The final testing accuracy of the baseline neural network UNet is 90.8%. the validation accuracy started from 59.42% in the first epoch and end it up at 87.58%. the validation loss went from 0.8323 to 0.3209 in 176 epochs. After the 176 epoch, the change in validation loss was insignificant. The training accuracy started from 53.47% and went up to 88.61%. as 88.61 is closer to 90.8 the model didn't overfit.

The number of epochs recorded for the transfer learning model is 20 because all the models roughly after 20 epochs started overfitting.

Model	Training Accuracy after first epoch in %	Training Accuracy after 20 epochs in %	Validation Accuracy after first epoch in %	Validation Accuracy after 20 epochs in %	Validation loss after first epoch	Validation loss after 20 epochs	Testing Accuracy in %
UNet	53.47	epoch number 176 - 88.61%	59.42	epoch number 176 - 87.58 %	0.8323	0.3209	90.8
Model-1	38.46	67.14	48.24	66.04	1.0933	0.8146	62.09
Model-2	33.96	62.15	41.01	63.31	1.0964	0.8185	65.82
Model-3	40.5	70.83	50.26	70.08	1.0669	0.7083	69.45
Model-4	39.11	86.55	47.55	84.1	1.0627	0.2778	82.38
Model-5	64.66	87.57	57.28	86.56	1.7139	0.3681	86.11
Model-6	40.2	90.63	46.13	86.68	0.9987	0.3799	90.26
Model-7	48.01	79.32	56.92	74.93	0.9713	0.7774	
Model-8	50.11	87.68	60.53	86.54	1	0.3987	88.49

Table 1: Comparison of different metrics of the models

Model	Time per epoch in seconds	Total Training time in seconds
UNet	521s	91696s
Model-1	0.720s	14.4s
Model-2	0.883s	17.7s
Model-3	0.943s	18.86s
Model-4	183s	3660s
Model-5	1.32s	26.4s
Model-6	1.64s	32.8s
Model-7	1.72s	34.4s
Model-8	3.48s	69.6s

Table 2 : Training time of models

5.2. Results from Model-1

The final testing accuracy of model-1 is 62.09% after 20 epochs. The validation accuracy started from 48.24% and went up to 66.04%. This model didn't have a significant result but by doing this experiment we came to know that using a simple decoder is not going to solve the segmentation problem. It is very clear from the difference in validation loss from epoch 1 to epoch 20 that the model was trying to fit the data but couldn't fit the data in 20 epochs. This indicates that to solve this problem we need to add a little bit of extra complexity. The model didn't overfit but it could be implied that it under fitted.

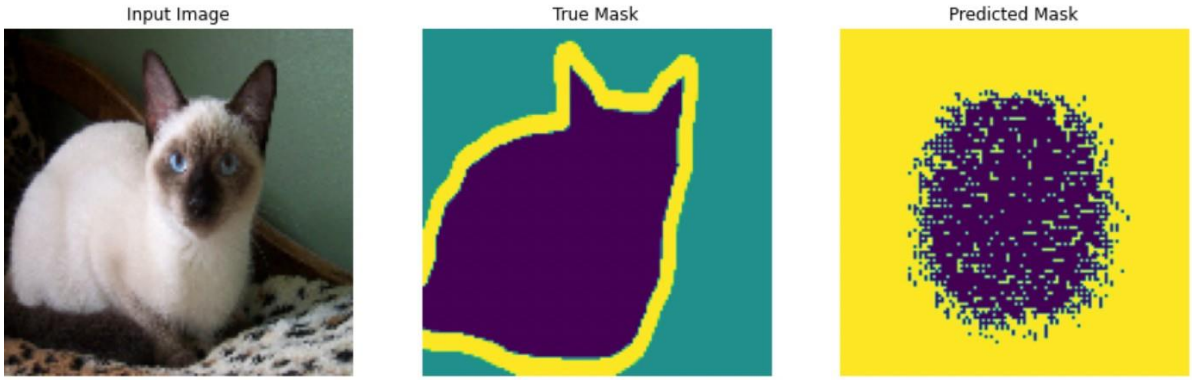

Figure 15: Sample output of Model-1

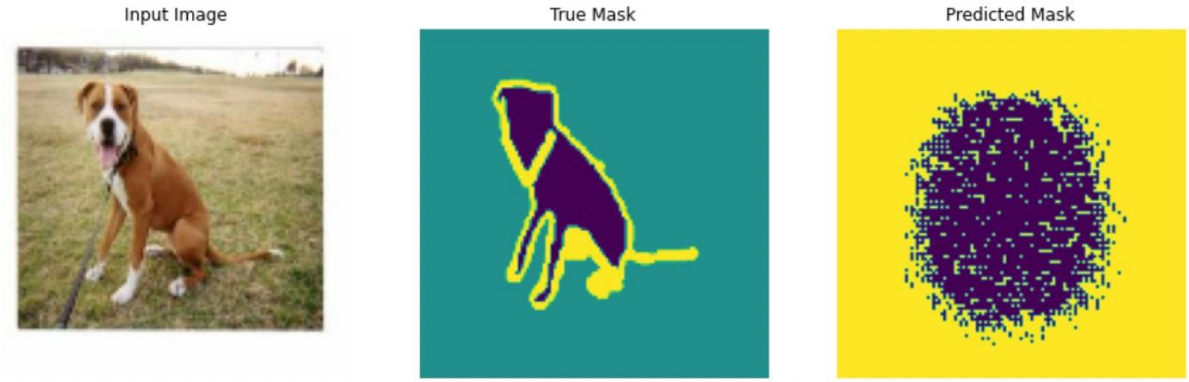

Figure 16: Sample output of Model-1

5.3. Results from Model-2

The final testing accuracy of model-2 is 65.82% after 20 epochs. The validation accuracy started with 41.01% and ended up at 63.31% surprisingly doing worse than model-1 on validation accuracy, but it generalized the data better than model-1 and had slightly higher testing accuracy. This model was also trying to fit the data but couldn't.

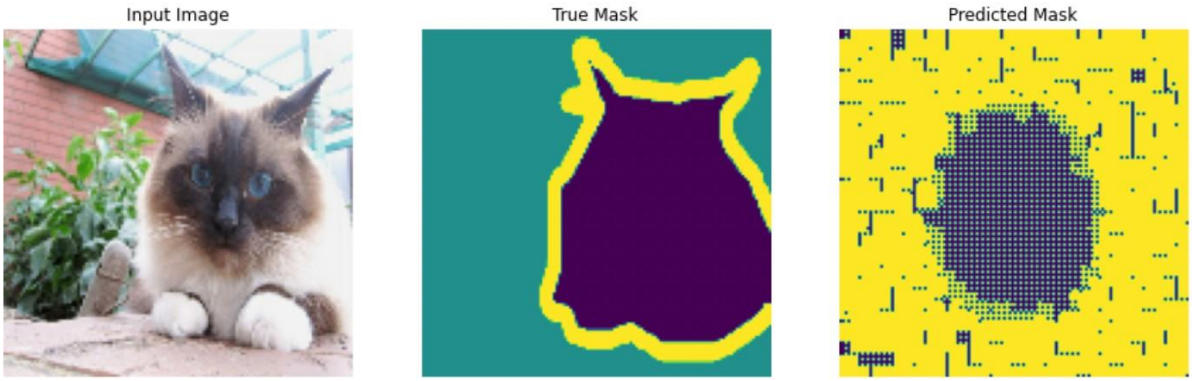

Figure 17:Sample output of Model-2

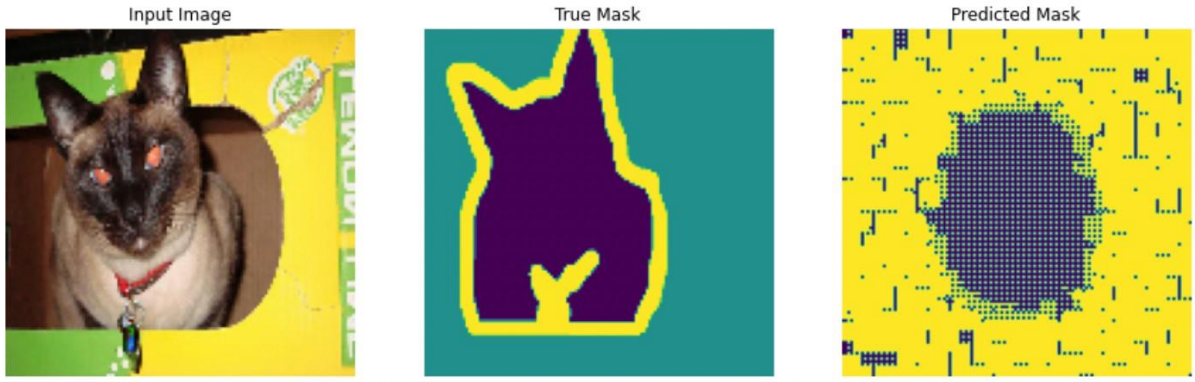

Figure 18:Sample output of Model-2

5.4. Results from Model-3

Model-3 is the transfer learning model where we introduced two dense layers and two convolution neural network layers to the encoder. The final accuracy of model-3 is 69.45%. The validation law started from 1.0669 and ended up at 0.7083 which indicates that the model was fitting properly. The validation accuracy started from 50.26% and ended up at 70.08% which is better than the previous two models. Training accuracy started from 40.5% at ended up at 70.83%. the model didn't overfit because the difference between training accuracy and validation accuracy is not significant. By the linear progression of increasing the density of decoders, we can observe that the testing accuracy is increasing as we increase the number of layers.

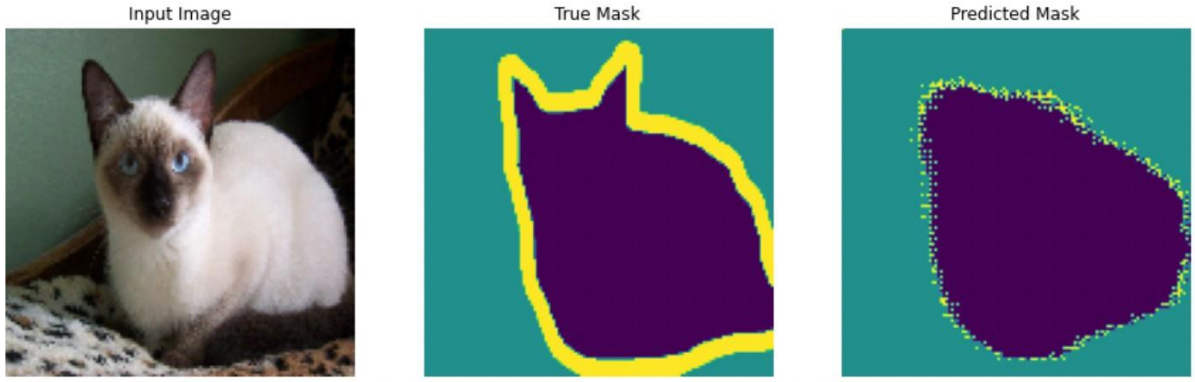

Figure 19:Sample output of Model-3

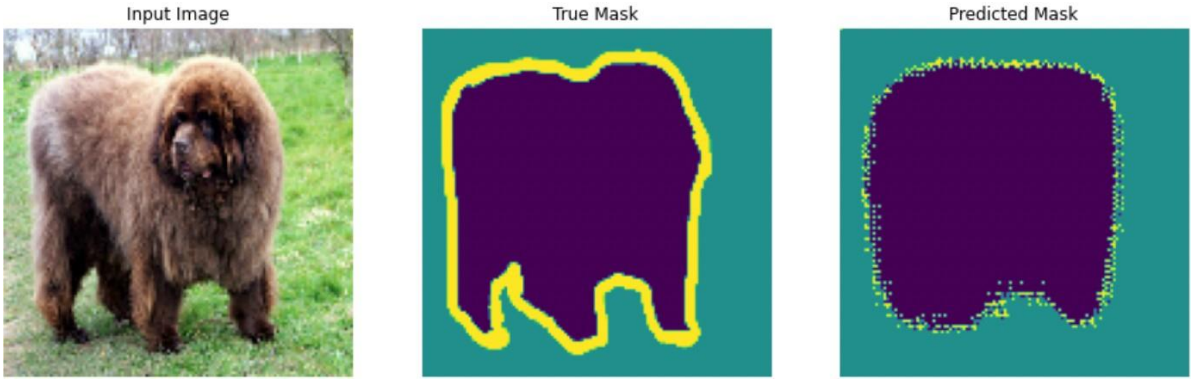

Figure 20:Sample output of Model-3

5.5. Results from Model-4

Model-4 has four dense layers and two convolution neural network layers. For this model, we have also trained the ResNet to see if we get a significant improvement in the testing accuracy. The idea here is to fi the data to the encoder as well as the encoder. By doing so we will be able to postulate whether the architecture itself can update the encoder weights and give us better results. Testing accuracy of model-4 is 82.3 8%. The validation loss dropped from 1.0627 to 0.2778, which indicates the weights of the encoder were changing rapidly using the advantage of the back-propagation algorithm. This model is also not an overfitting model overfitting because the training accuracy was around 86% and validation accuracy was around 84% which does not have a huge difference. Adapting the weights of the encoder increased the time of training exponentially as shown in Table 2 as all the weights were getting updated for the encoder using back-propagation and gradient descent, the process is computationally expensive. As the results suggest the model did quite a good job in generalizing the data but took a lot of time to train itself.

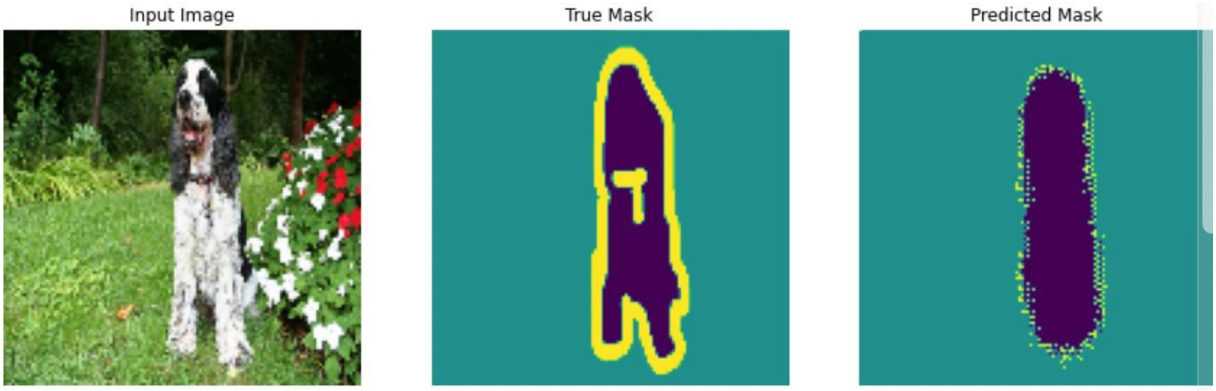

Figure 21:Sample output of Model-4

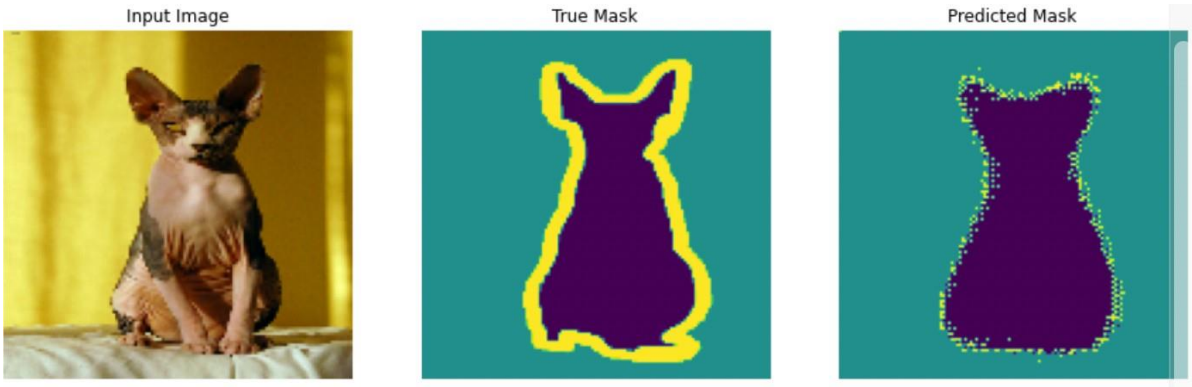

Figure 22:Sample output of Model-4

5.6. Results from Model-5

The number of layers in model-5 has been increased by two layers with linear incremented notes from model-4, which implies that model-5 is more complex than model-4 and previous models. This pattern can also be observed via the incremental time of training models per eepochexcept t force model-4 (as we did not freeze the ResNet architecture and it was also part of training), as we increase the complexity of the model the time taken to train the model also get increased as the model will have to perform more number of computationally intensive operations to fit the data. The ResNet architecture in model-5 has frozen weights was were not retrained. The final accuracy of model-5 is 86.1% doing better than all the previous models. The validation loss dropped from 1.7139 to 0.3681. No traces of overfitting can be observed here as the training accuracy is 87.57% and the validation accuracy is 86.56% which is quite near the training accuracy.

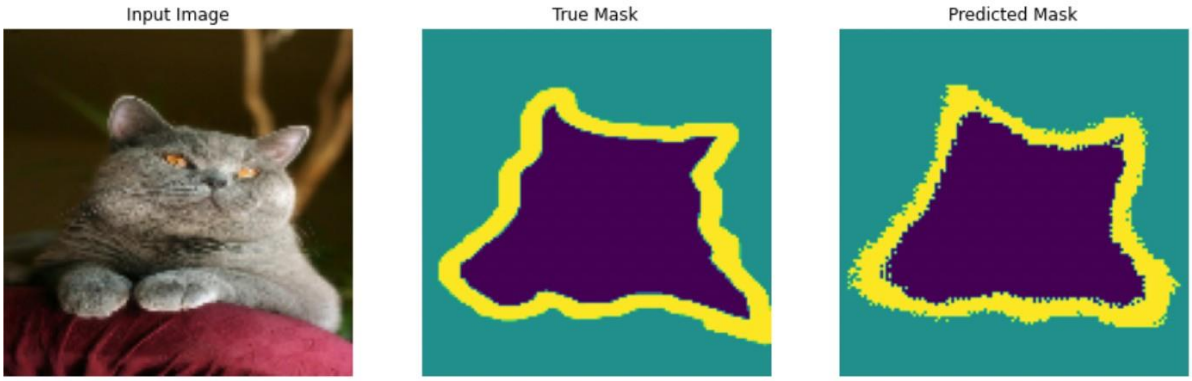

Figure 23:Sample output of Model-5

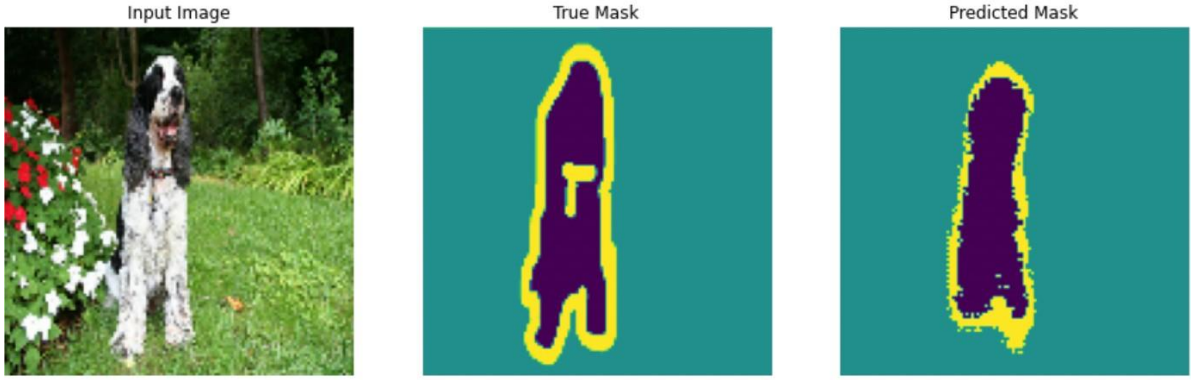

Figure 24:Sample output of Model-5

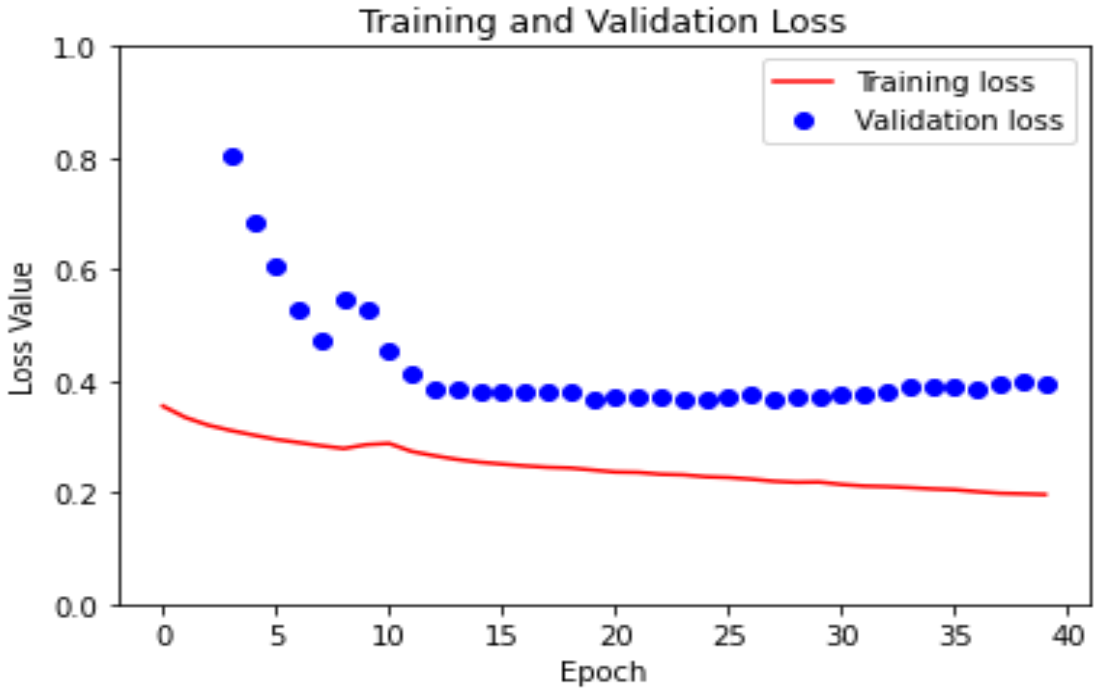

Figure 25:Training and Validation Loss vs Loss value curve for Model-5

5.7. Results from Model-6

Model 6 has 8 dense layers and two convolution neural network layers. The testing accuracy of this model is 90.26% which is very much close to the baseline (UNet) which is 90.8%. This indicates at model 6 has successfully achieved the learning capacity which is equivalent to the learning of the baseline model (UNet). This model has not at all overfitted the data as there is a very little difference between training and testing accuracy little but not to a very big extent even with that, it has generalized the test data correctly. The training accuracy started at 48.01% and ended up at 90.63%. The time taken to train all data with 20 epochs in the model is 34.4 seconds which is approximately 99% faster than the baseline model UNet.

We do furthermore tests using model-7 where we initialize the first layer weights randomly and freeze them to see if the pretrained weights of ResNet have an impact on the learning of data orif are we getting results based on some random fluke.

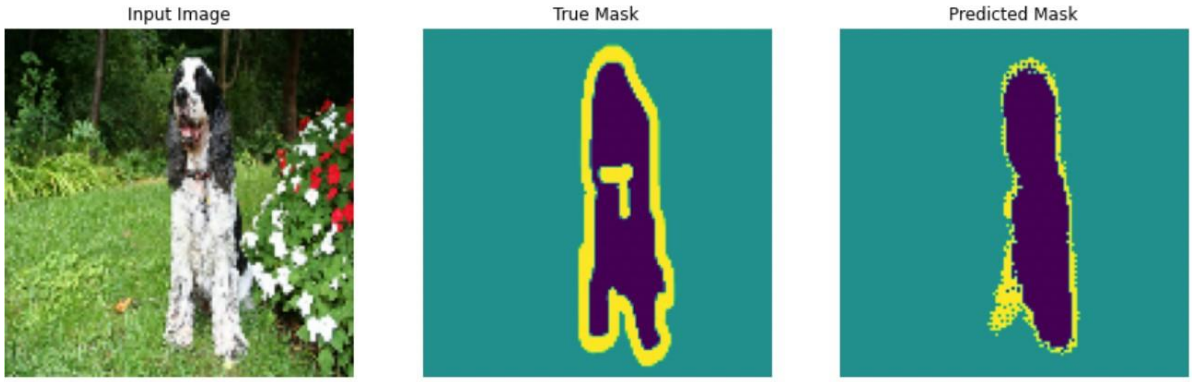

Figure 26:Sample output of Model-6

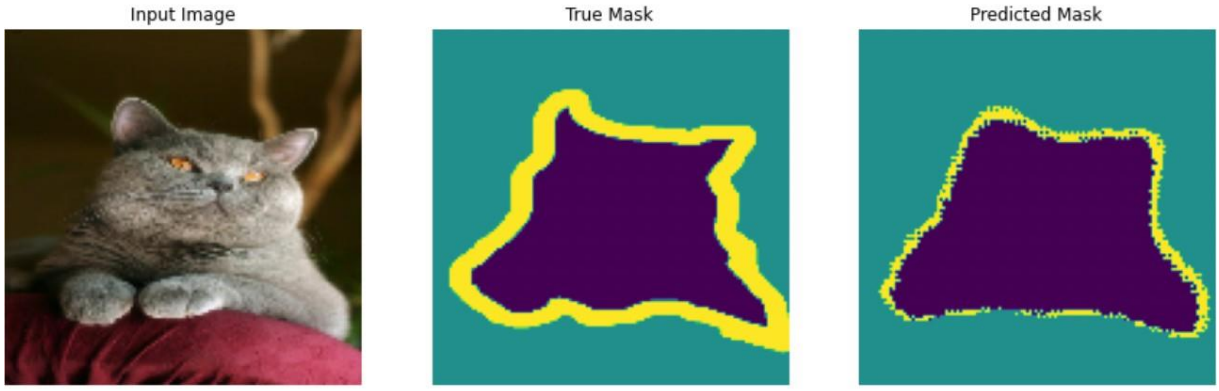

Figure 27:Sample output of Model-6

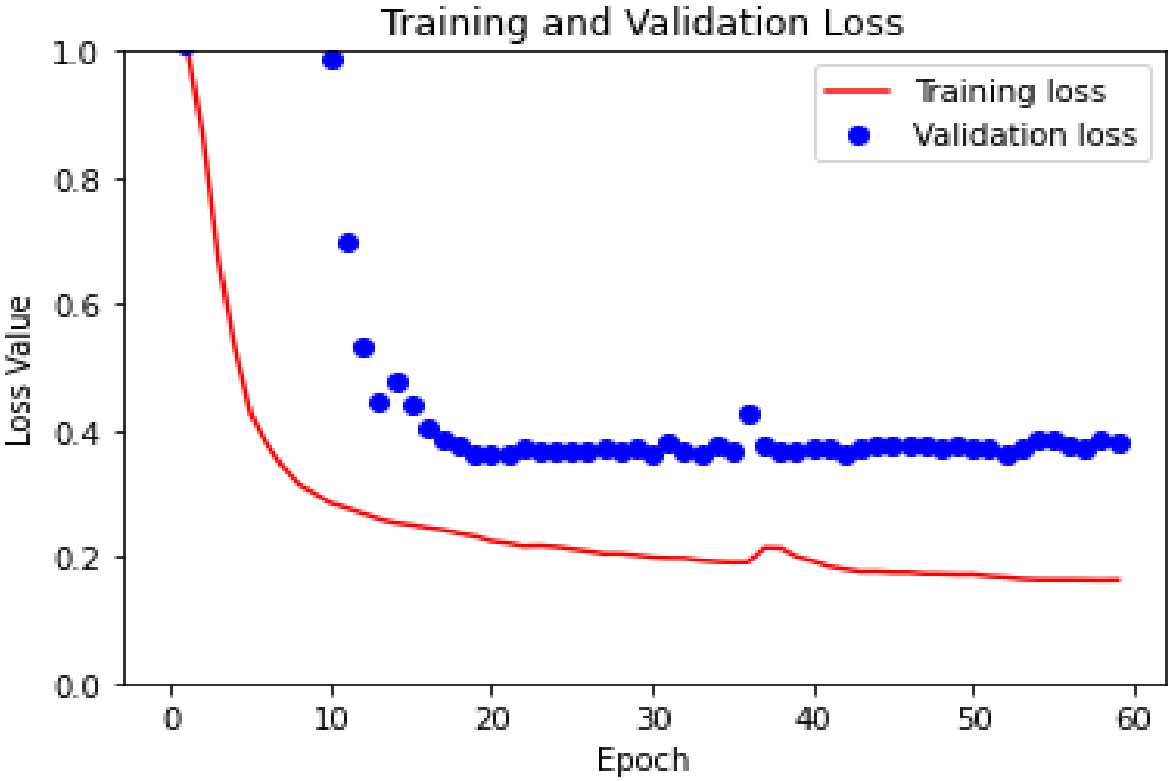

Figure 28:Training and Validation Loss vs Loss value curve for Model-6

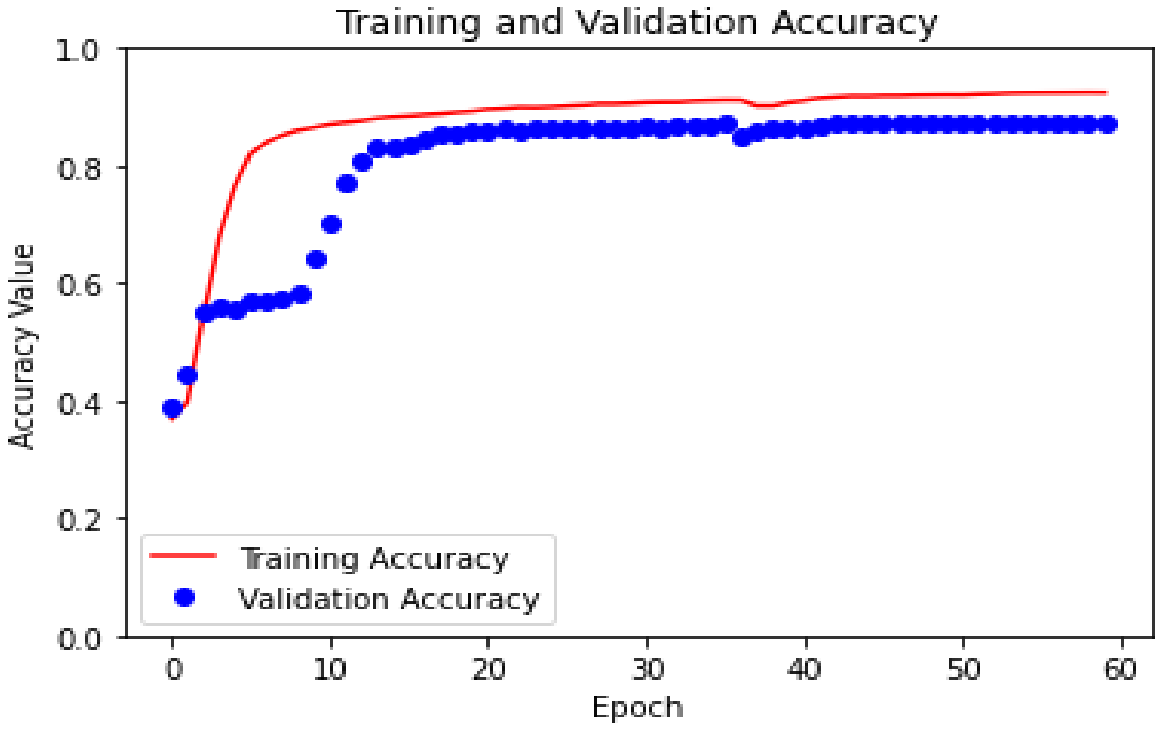

Figure 29:Training and Validation Loss vs Accuracy value curve for Model-6

5.8. Results from Model-7

The architecture of model-7 is the same as the architecture of model-6 the only difference is that we initialize the first layer weights of model-7 randomly and freeze them. By doing so model-7 gives less accuracy than model-6, which implies the pre-trained weights of model-6 are important to achieve baseline accuracy, after tampering with even one layer of weights theaccuracy got depreciated by approximately 10%.

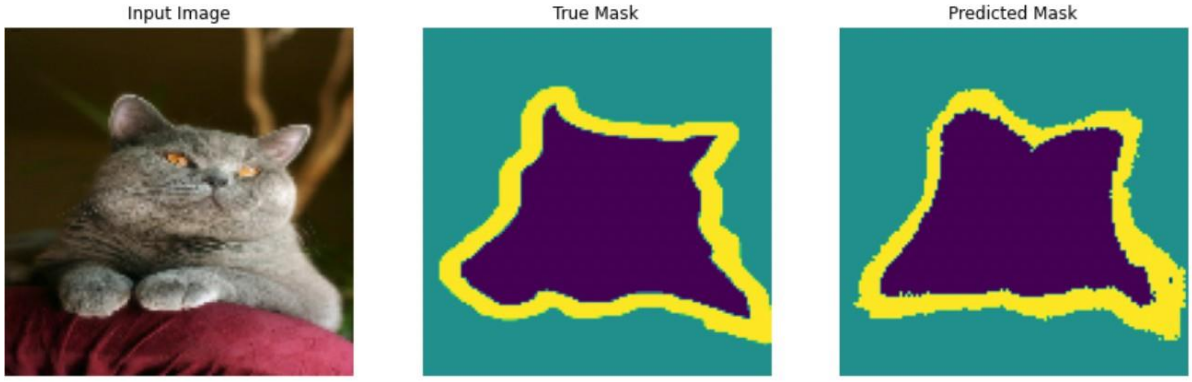

Figure 30:Sample output of Model-7

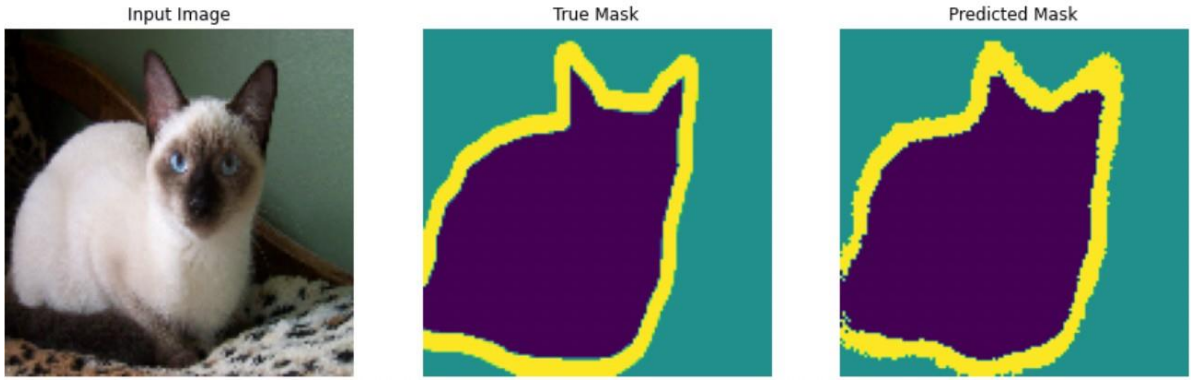

Figure 31:Sample output of Model-7

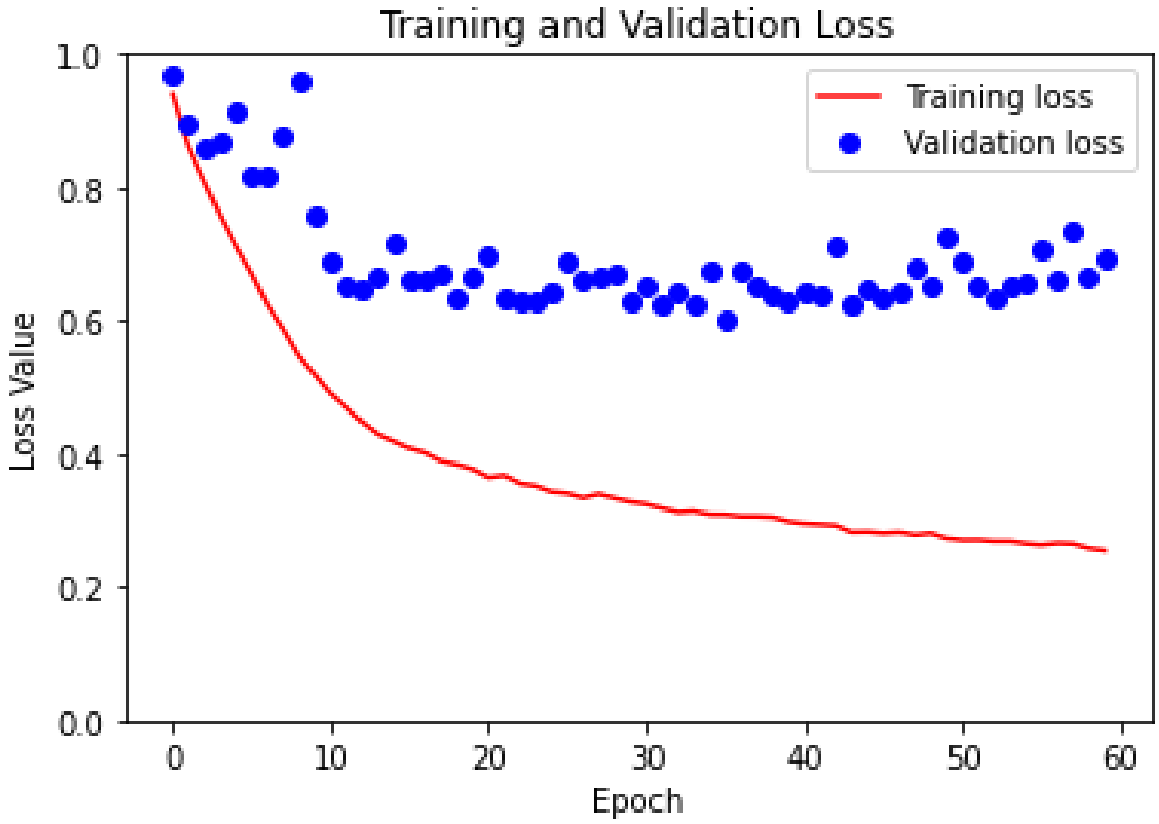

Figure 32:Training and Validation Loss vs Loss value curve for Model-7

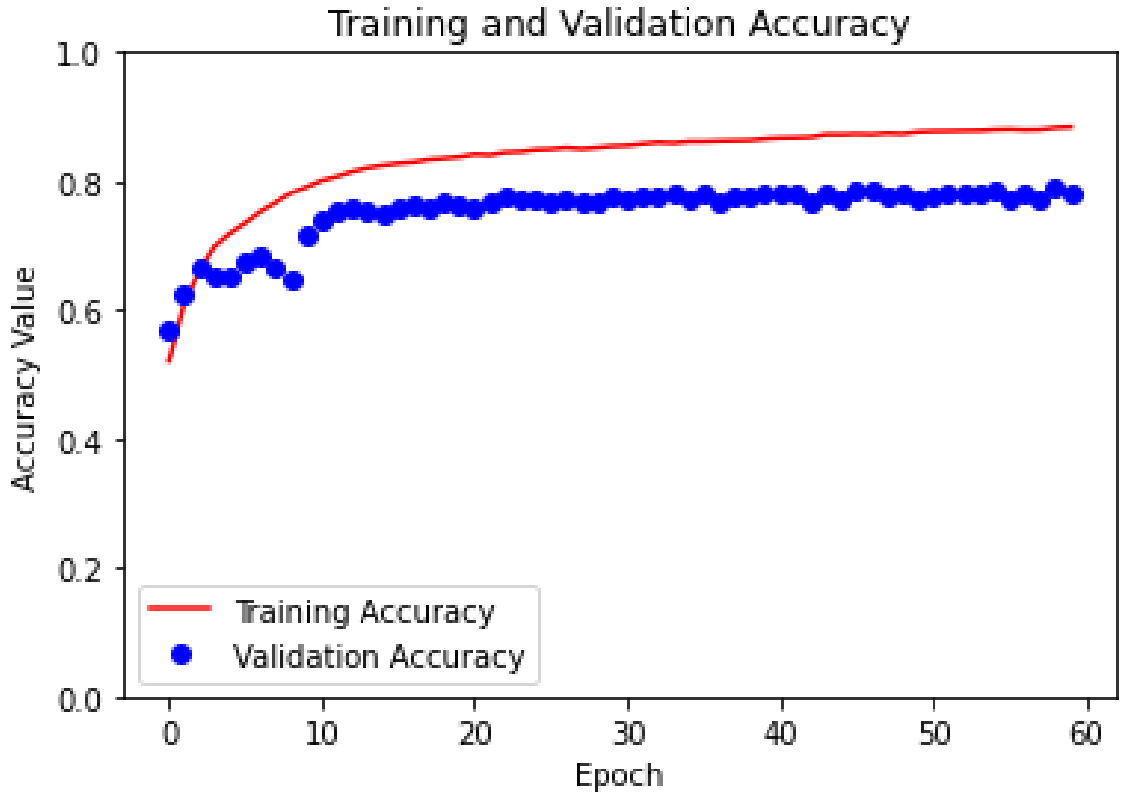

Figure 33:Training and Validation Loss vs Accuracy value curve for Model-7

5.9. Results from Model-8

Model-8 uses ResNet 150 instead of ResNet 50 as its encoder, here also we freeze the weights of the ncoderencoder and we have 8 dense layers and two convolutional neural network layers in the decoder. As expected ResNet 150 is more complex than ResNet 50, as the time taken to propagate the data from input to output will computationally take more time. In this case, it is more than twice. Even with the additional complexity of ResNet 150, the model-8 is almost similar to the baseline model that means even after increasing the complexity we cannot get a better generalization of the testing data the model-8 is saturated.

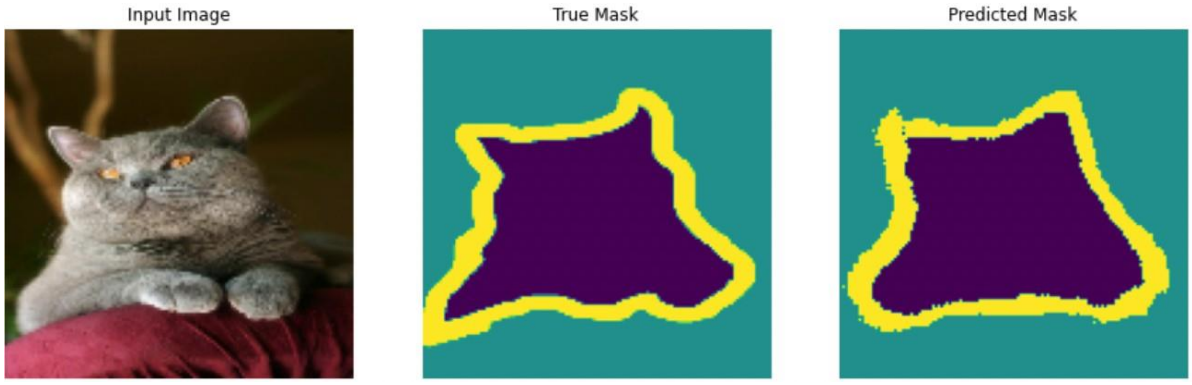

Figure 34:Sample output of Model-8

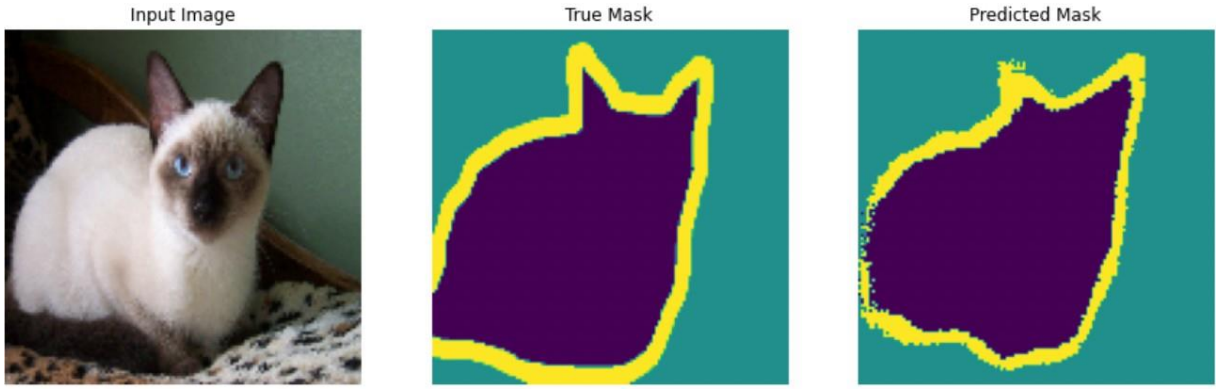

Figure 35:Sample output of Model-8

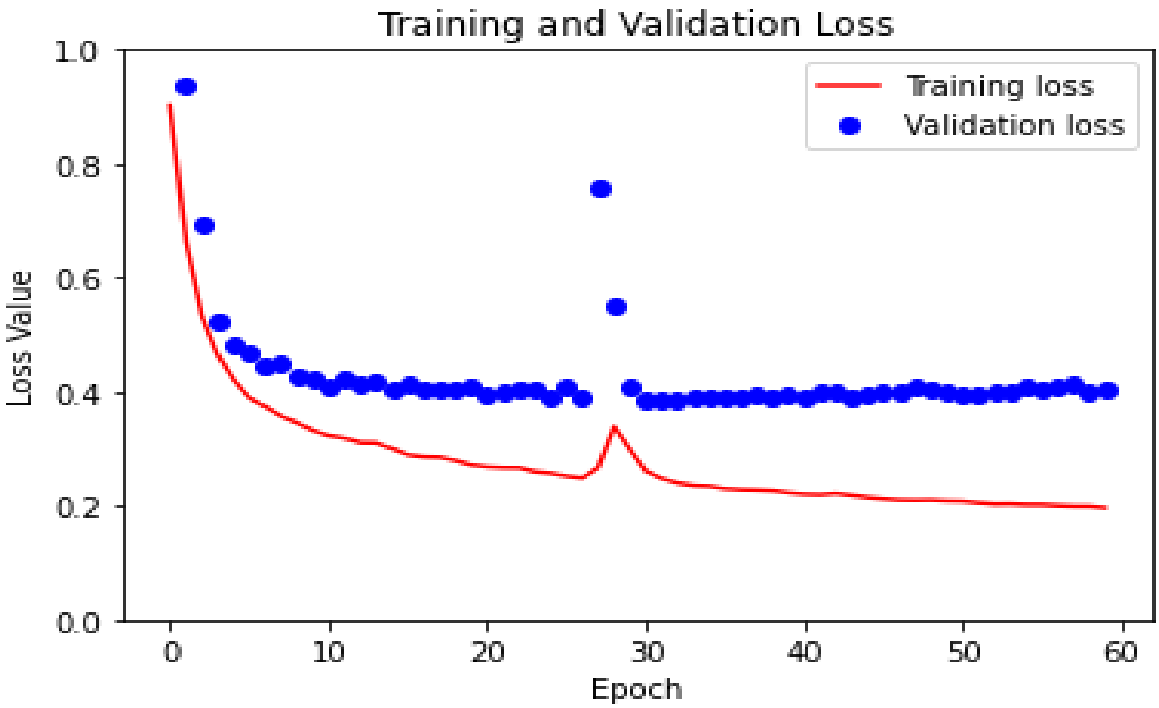

Figure 36:Training and Validation Loss vs Loss value curve for Model-8

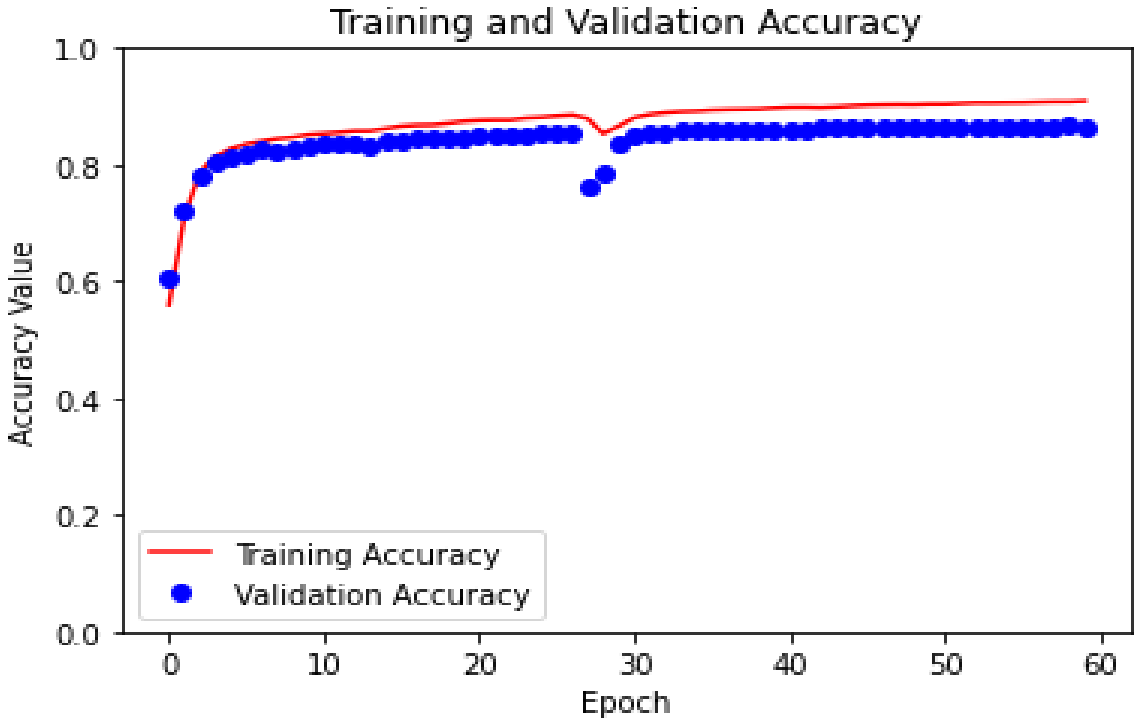

Figure 37:Training and Validation Loss vs Accuracy value curve for Model-8

5.10. Summary

Transfer learning significantly reduces the training time by giving equivalent results to normal neural network training. We tested different models with different combinations of architecture (primarily using encoder-decoder transfer learning) to justify the results and to prove that transfer learning for image segmentation problems on the Oxford IIIT pet data set is very helpful. It saves time and computational capacity.

CHAPTER 6. CONCLUSION AND RECOMMENDATION

6.1. Introduction

This section gives the just of the project along with some additional future work that can be done over this project and throw light on some skills that we have achieved in the tenure of executing the project.

6.2. Discussion and conclusion

In the project, our goal was to implement transfer learning using an encoder-decoder network using the ResNet model. The data set selected was the Oxford IIIT pet dataset which contains images of dogs and cats annotated with segments to identify the pets. We created different models to study the promise concerning the baseline model which is UNet. In the process, the objective became to cre″ate a model using transfer learning which will be as good as UNet and can be trained in a much faster duration without getting overfitted.

The approach was to have ResNet as the transfer learning and coder and create multiple decoders network and fine-tune the decoder. We created 8 different models to study the premises. The initial three models were very basic decoders starting with some dense layers followed by convolutional neural network layers. In the process we do not train our encoder which is ResNet, the idea here is to use pre-trained weights of ResNet trained on the image-net data (very similar to the Oxford IIIT pet dataset) set so that we don't have to reinvent the wheel and we can use the encoded features and decode them to form our segmentation maps.

We also tried one model where we update the weights of our encoder to understand what will be the additional impact if we find you never model according to the given data set. Then we created more complex models by adding more dense layers to increase the complexity and the following dense layers keep getting an increased number of nodes. Then using convolutional neural network layers the encoded vectors from the dense layer get converted into a segmentation map from the desired input. While training model-6 we attained good optimality in the testing data set without overfitting the model, and we found our model.

Now further on we did two more tests to ensure that this model is backed by some scientific justification and not by any random fluke. The first test was to use the same architecture and randomly initialize the first layer weights and freeze the encoder weights to check the impact of pretrained weights of the encoder, as the result we saw a decrease in the testing accuracy of the model where we manipulated the weights which confirmed that the pre-trained weights areimportant to get the result same as UNet.

The second test was to increase the complexity of the model by using resonate 150 instead of ResNet 50 and training the model. This experiment resulted in almost similar accuracy which tells that the model does not require any other complexity and cannot be improved further with similar architecture. By doing Sachin of experiments of creating models and backing our hypothesis with factual data we concluded that transfer learning is very much effective on particular data sets which are similar in nature to which the main models are trained and the encoder-decoder transfer learning process saves a lot of time to train the model. Which results in scalability and generalization of particular types of problems. So if we have a similar type of problem in the future where a particular model has been created we can use that model and fit the model for the data set to train it much faster which intern makes the process scalable.

From the results of transfer learning, we can observe a pattern where the model starts getting overfitted after approximately 20 epochs and the validation loss does not drop while in the case of the baseline model unit it took around 176 epochs to give approximately 90% accuracy whereas in case of model-6 we achieved the same accuracy in less number of epochs. It should also be noted that the average time to run epochs for a UNet was 521 seconds whereas the average time to run 1 epoch of model 6 is 1.64 seconds which is approximately 99% less.

6.3. Contribution to knowledge

The main contribution to the project was to optimize the neural network for the particular problem and try to generalize it. In the process, we settled up multiple tests to verify some assumptions that logically came along the way. The first test was to confirm that the resonate words are adding value to the network as in the first place we are preserving them and the baseof the whole project standing on this concept.

The other test was to check the saturation level for this particular problem and see to what extent it can be solved using the existing ResNet architecture. It was also challenging to code and train UNet from scratch in the form of an encoder-decoder network as UNet is generally used in classification types of projects. This project constantly challenged our data science code implementation and analytical thinking skills, as time progressed we were able to achieve specialty in the domain to solve the problem and think out of the box.

6.4. Future recommendations

the following premises can be extended into various academic and industrial studies. The current premises use only a single data set, a good approach can be to check it on multiple data sets for segmentation problems. Further, this study can be extended in isolation to image segmentation into multiple divisions of image processing for example image layering, background removal, etc

As this study was a progression of using multiple complexities of neural network another study can be made where the complexity of the neural network is handled by another neural network and from the given data set the architecture of the neural network for transfer learning encoded decoder network changes dynamically based on the data set input.

Another study that could be of high importance using the same architecture can be implementing systems for the current social media trends like sketch conversion of an image, cartoon conversion of an image, caricaturing an image, automatic editing of an image, etc.

This study can be taken further to any other area which is already explored in deep learning and we need to make scalable models out of it, basically the model-building process should not be in coherence with reinventing the wheel, it can be more of the plug, finetune and play.

REFERENCES

Blitzer, J., Dredze, M., & Pereira, F. (2007). Biographies, Bollywood, boom-boxes, and blenders: Domain adaptation for sentiment classification. Proceedings of the 45th AnnualMeeting of the Association of Computational Linguistics, 440–447.

Boykov, Y., Veksler, O., & Zabih, R. (2001). Fast approximate energy minimization via graph cuts. IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(11), 1222– 1239.

Brejl, M., & Sonka, M. (2000). Object localization and border detection criteria design in edge- based image segmentation: automated learning from examples. IEEE Transactions on Medical Imaging, 19(10), 973–985.

Chan, A. B., Liang, Z.-S. J., & Vasconcelos, N. (2008). Privacy-preserving crowd monitoring: Counting people without people models or tracking. 2008 IEEE Conference on ComputerVision and Pattern Recognition, 1–7.

Deng, L. (2014). A tutorial survey of architectures, algorithms, and applications for deep learning. APSIPA Transactions on Signal and Information Processing, 3.

Dhanachandra, N., Manglem, K., & Chanu, Y. J. (2015). Image segmentation using K-means clustering algorithm and subtractive clustering algorithm. Procedia Computer Science, 54,764–771.

Farabet, C., Couprie, C., Najman, L., & LeCun, Y. (2012). Learning hierarchical features for scene labeling. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(8), 1915–1929.

Fathi, A., Ren, X., & Rehg, J. M. (2011). Learning to recognize objects in egocentric activities. CVPR 2011, 3281–3288.

Forsyth, D., & Ponce, J. (2002). Prentice hall professional technical reference. Computer Vision: A Modern Approach.

Friedman, N., & Russell, S. (2013). Image Segmentation in Video Sequences: A Probabilistic Approach.

Fung, G. P. C., Yu, J. X., Lu, H., & Yu, P. S. (2005). Text classification without negative examples revisit. IEEE Transactions on Knowledge and Data Engineering, 18(1), 6–20.

Gonzalez, R. C. (2009). Digital image processing. Pearson education india.

Goodfellow, I. J., Erhan, D., Carrier, P. L., Courville, A., Mirza, M., Hamner, B., Cukierski, W., Tang, Y., Thaler, D., Lee, D.-H., & others. (2013). Challenges in representation learning: A report on three machine learning contests. International Conference on NeuralInformation Processing, 117–124.

Jawahar, C. v, Zisserman, A., Vedaldi, A., & Parkhi, O. M. (2012). Cats and dogs. 2012 IEEEConference on Computer Vision and Pattern Recognition, 3498–3505.

Jetley, S., Lord, N. A., Lee, N., & Torr, P. H. S. (2018). Learn to pay attention. ArXiv PreprintArXiv:1804.02391.

Kass, M., Witkin, A., & Terzopoulos, D. (1988). Snakes: Active contour models. InternationalJournal of Computer Vision, 1(4), 321–331.

Kaur, D., & Kaur, Y. (2014). International Journal of Computer Science and Mobile Computing Various Image Segmentation Techniques: A Review. International Journal of Computer Science and Mobile Computing, 3(5), 809–814. www.ijcsmc.com

Kuehne, H., Arslan, A., & Serre, T. (2014). The language of actions: Recovering the syntax and semantics of goal-directed human activities. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 780–787.

Milletari, F., Navab, N., & Ahmadi, S.-A. (2016). V-net: Fully convolutional neural networks for volumetric medical image segmentation. 2016 Fourth International Conference on 3DVision (3DV), 565–571.

Otsu, N. (1979). A threshold selection method from gray-level histograms. IEEE Transactionson Systems, Man, and Cybernetics, 9(1), 62–66.

Pan, S. J., & Yang, Q. (2009). A survey on transfer learning. IEEE Transactions on Knowledgeand Data Engineering, 22(10), 1345–1359.

Pan, S. J., Zheng, V. W., Yang, Q., & Hu, D. H. (2008). Transfer learning for wifi-based indoor localization. Association for the Advancement of Artificial Intelligence (AAAI) Workshop, 6.

Patro, S., & Sahu, K. K. (2015). Normalization: A preprocessing stage. ArXiv Preprint ArXiv:1503.06462.

Plath, N., Toussaint, M., & Nakajima, S. (2009). Multi-class image segmentation using conditional random fields and global classification. Proceedings of the 26th Annual International Conference on Machine Learning, 817–824.

Pramerdorfer, C., & Kampel, M. (2016). Facial expression recognition using convolutional neural networks: state of the art. ArXiv Preprint ArXiv:1612.02903.

Sainath, T. N., Kingsbury, B., Mohamed, A., Dahl, G. E., Saon, G., Soltau, H., Beran, T., Aravkin, A. Y., & Ramabhadran, B. (2013). Improvements to deep convolutional neural networks for LVCSR. 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, 315–320.

Shrestha, K., Shrestha, P. P., Bajracharya, D., & Yfantis, E. A. (2015). Hard-hat detection for construction safety visualization. Journal of Construction Engineering, 2015(1), 1–8.

Starck, J.-L., Elad, M., & Donoho, D. L. (2005). Image decomposition via the combination of sparse representations and a variational approach. IEEE Transactions on Image Processing, 14(10), 1570–1582.

Stein, S., & McKenna, S. J. (2013). Combining embedded accelerometers with computer vision for recognizing food preparation activities. Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, 729–738.

To, M. N. N., Vu, D. Q., Turkbey, B., Choyke, P. L., & Kwak, J. T. (2018). Deep dense multi-path neural network for prostate segmentation in magnetic resonance imaging. International Journal of Computer Assisted Radiology and Surgery, 13(11), 1687–1696.

Tompson, J. J., Jain, A., LeCun, Y., & Bregler, C. (2014). Joint training of a convolutional network and a graphical model for human pose estimation. Advances in Neural Information Processing Systems, 27.

Torrey, L., & Shavlik, J. (2010). Transfer learning. In Handbook of research on machine learning applications and trends: algorithms, methods, and techniques (pp. 242–264). IGIglobal.

Wu, X., Kumar, V., Ross Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G. J., Ng, A., Liu, B., Yu, P. S., & others. (2008). Top 10 algorithms in data mining. Knowledge andInformation Systems, 14(1), 1–37.

Zhang, C., Li, H., Wang, X., & Yang, X. (2015). Cross-scene crowd counting via deep convolutional neural networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,833–84