Prostate Cancer Detection using Deep Learning Neural Network
with
Transfer Learning Approach

by

Ariful Islam Mahmud Badhon
17101314
Md. Sadman Hasan
17101413
Md. Samiul Haque
17301169
Md. Shafayat Hossain Pranto
18301238
Saurav Ghosh
17101355

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
October 2021

Declaration

It is hereby declared that

1. The thesis submitted is our own original work while completing a degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Samiul Haque

Md. Samiul Haque
17301169

Md. Sadman Hasan
17101413

Shafayat

Md. Shafayat Hossain Pranto
18301238

Arif

Ariful Islam Mahmud Badhon
17101314

Sghosh

Saurav Ghosh
17101355

Approval

The thesis titled “Prostate Cancer Detection Using Deep Learning Transformer Neural Network with Transfer Learning Approach” submitted by

1. Ariful Islam Mahmud Badhon (17101314)
2. Md. Sadman Hasan (17101413)
3. Md. Samiul Haque (17301169)
4. Md. Shafayat Hossain Pranto (18301238)
5. Saurav Ghosh (17101355)

Of Summer, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on October 05, 2021.

Examining Committee:

Supervisor:
(Member)

Md. Golam Rabiul Alam, PhD
Associate Professor
Department of Computer Science and Engineering
Brac University

Co-Supervisor:
(Member)

Warida Rashid

Warida Rashid
Lecturer and Assistant Proctor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Ethics Statement

We hereby state that all the work done in preparing this thesis, beginning from the research to implementation of the proposed system and models is our own work. The paper reflects our own research and analysis in a truthful and accurate manner. All the datasets and external sources that we used have been acknowledged. We have refrained from using any unethical means in obtaining any of the resources used. We hereby testify that this paper has not been previously submitted in part or full by anyone or any team to any other universities or institutes to complete a degree.

Abstract

Prostate cancer is a ubiquitous form of cancer detected among men all over the world. It is currently the second leading cause of cancer death worldwide among men. Research shows that about 11% of men worldwide are affected by prostate cancer at some point during their lives. In our thesis, we have used a Transfer Learning approach for the Deep Learning model to compare the precision in results using machine learning classifiers. We have also evaluated performance in terms of classification with different evaluation measures using a Deep Learning pre-trained network (VGG16). Parameters such as Precision, Recall, F1 score and Loss vs Accuracy were assessed thoroughly as different performance measures. After applying the Transfer Learning approach, we have recorded the peak performance using the VGG16 architecture. We used the convolutional block and dense layers of VGG16 architecture to extract features from image datasets. We forwarded those features to Machine Learning classifiers for the final classification result. We have procured outstanding accuracy using the Deep Machine Learning method in our research.

Keywords: Prostate Cancer; Deep Learning; ImageNet; Transfer Learning; VGG16; Image Classification; Machine Learning classifier

Acknowledgement

Firstly, we would like to thank the Almighty for giving us the strength to move forward with our research during this tumultuous time. The Covid-19 pandemic has made life hard for everyone for the past two years; we would not be where we are today without His divine blessings. We would also like to give special thanks to our supervisor, for guiding us throughout this long and arduous journey, correcting our mistakes and keeping us motivated to persevere in the course of our work. Furthermore, we want to thank our co-supervisor for sparing us her precious time every now and then to impart her valuable feedback. Finally, we would like to thank our parents and members of our family for their sustaining love and support.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

$AUROC$	Area Under the Receiver Operating System
BPH	Benign Prostatic Hyperplasia
CNN	Convolutional Neural Network
$Conv2D$	2D Convolution Layer
CRF	Conditional Random Fields
CSV	Comma-separated Values
DRE	Digital Rectal Examination
DT	Decision Tree
$FCNs$	Fully Convolutional Neural Networks
MR	Magnetic Resonance
MRI	Magnetic Resonance Imaging
PCA	Principal Components Analysis
PSA	Prostate Specific Antigen
$PSAD$	Prostate Specific Antigen Density
$ReLU$	Rectifying Linear Unit
RNN	Recurrent Neural Network
$RoIAlign$	Region of Interest Align
$TCIA$	The Cancer Imaging Archive
$TRUS$	Transrectal Ultrasonography
UID	Unique Identification
$VGG16$	Visual Geometry Group 16

Chapter 1

Introduction

1.1 Introduction

Prostate cancer is a form of cancer that is diagnosed frequently among men. The prostate is a gland that sits between the bladder and the urethra of men. It produces fluid that washes semen in addition to keeping sperm healthy for successful fertilization. In the initial stage of prostate cancer, it starts growing within the edges of the prostate. Gradually, it spreads through the lining and around the edges of the prostate. About 1 in 7 men are detected with prostate cancer at some point in their lives all around the globe. Prostate cancer is usually fatal if it is not detected and treated at the early stage of prognosis [18]. The current principal assessment techniques for Prostate Cancer detection are computerized rectal assessment, Prostate-Specific Antigen (PSA) test and Transrectal Ultrasound (TRUS) [17]. Among all the available methods for diagnosis, PSA remains the most commonly known method of screening and detecting cancer. Prostate-Specific Antigen test detects cancer by examining the level of PSA in the patient's blood cells. However, there remain significant impediments when it comes to the reliability of PSA tests. One of those impediments is the high rate of false positives due to prostate hyperplasia or inflammations. Such specific conditions often result in an increased level of PSA, which could render a wrongful prognosis report [9]. Currently, a Transrectal Ultrasound-guided biopsy remains the most reliable method of prostate cancer prognosis. However, a biopsy is regarded as the last resort, as the procedure tends to be expensive and inefficient. Research shows that roughly 30% of benign tumours (Stage I and II) are not detected during the initial phase of prognostication, owing to the unguided approach of TRUS-guided biopsy. We can reduce the mortality rate of prostate cancer by making the diagnosis thorough and precise. Studies show that we can reduce the fatality rate from prostate cancer significantly; if we can diagnose the patients at an earlier stage of their ailments [9].

1.2 Problem Statement

1.2.1 Problem Statement

The Prostate-Specific Antigen test is the most commonly used diagnosis method of prostate cancer. However, PSA has constraints of its own where we can achieve improvements [2]. The accuracy of the PSA diagnosis goes down if the serum values hover between the range of 4 g/L to 10 g/L. Many components can cause fluctuations in serum values which can ultimately render a wrongful diagnosis. In addition, factors such as Benign Prostatic Hyperplasia (BPH), inflammation and chronic prostatitis can also inflate the PSA levels of some patients. To overcome such impediments, we have erected Decision Tree models to assimilate the prostate cancer classification [4]. After using this model, we have recorded that Prostate-Specific Antigen Density has a higher correlation with prostate cancer than PSA. Among the other available tools, Magnetic Resonance (MR) images can play an extensive role in providing detailed analytical information regarding the prostate. However, opaque boundary lines at the peak and the bottom remain a significant obstacle to this procedure. Some networks try to apply a patch-to-pixel method to obtain a prediction; although, it does not extract the expected amount of efficiency or quality in terms of results. Fully Convolutional Neural Networks can also pave the way to training the system picture-on-picture while training a large volume of samples simultaneously. Although, it can be hard to implement FCN in prostate segmentation as the challenges such as mentioned above continue to persist. Likewise, if we want to implement the same method with U-net, we would see the prostate images derived from the segmentation of representative samples have smeared boundaries. We would also notice the distributions of the pixel intensity as inhomogeneous on both sides of the prostate. The segmentation can get conspicuously difficult if the prostate and non-prostate areas show similar contrast and potency distributions [5]. Zhu Q et al. discussed a possible solution to this problem by suggesting a network that can advance the features drawn out from the initial stages to the latter in a bid to avert a potential loss of information. If hidden layers have components manufactured with logical meaning, we can add additional deeply supervised layers at every stage. This recommended network by the writers mentioned above is called Deeply-Supervised CNN with an end-to-end training model. It can also segment the gland on Magnetic Resonance (MR) images thoroughly and expediently. Deeper Network can help us attain a lofty level of precision. Although, there are two obstacles to overcome if we are to implement Deeper Network. Firstly, Deeper Network requires a lot of attributes, which can cause the network to be vulnerable to overfitting problems. Also, another obstacle would be to counteract the increased use of computational sources [7]. Author Zhu Q has proposed using a 1×1 convolutional layer which has two distinct upsides. First of all, it can drastically reduce the proportions and the number of variables, diminishing computational backlogs significantly. It can also enhance the depth of the network significantly and improve the accuracy of image segmentation. Lastly, to summarize, differentiating between PCa and BPH could be difficult as both are related to PSA levels in patients. Besides, some neural networks are inefficient, time-consuming and lacking in precision. However, as discussed above, we can use the DT algorithms for differentiating PCa and BPH and use different Machine Learning techniques such as SVM Kernels and K-Fold Cross-Validation to train certain phases. We can also use feature extraction techniques to implement CNN with transfer learning to overcome all the impediments and procure an optimal result.

1.3 Research Objectives

Our research aims to improve deep neural network's generalization capabilities significantly in prostate cancer detection. Transfer learning for deep machine learning is the process of first training a base network on a benchmark dataset (i.e., ImageNet) and then transferring the best-learned network features (the network's weights and structures) to a second network to train on a target dataset. The objectives of this research are as follows:

1. To deeply understand the working principles of CNN and various machines.
2. To implement the transfer learning approach for CNN to increase efficiency.
3. To develop an effective classification method of prostate cancer images using Deep CNN architecture with Machine Learning classifiers.
4. To evaluate the model. To offer recommendations on improving the model.

Chapter 2

Literature Review and Relevant Work

2.1 Deep Convolutional Neural Network

Deep Learning models have shown significant breakthroughs in medical image segmentation over the last few years [24]. Authors Wang et al (2018) have discussed various CNN models for image segmentation, such as MR images, pancreas in CT images and glands in pathology images. They have also found the proposed 3D Deep dense multipath CNN model for segmentation of prostate MRI dataset. This model could achieve a high level of accurate segmentation. It affirms the stark advantages of using 3D CNN models compared to 2D CNN models in medical image analysis [16]. Inputs in the traditional CNN are screened through convolutional layers, using filters and non-linearity activation functions. ReLU is implemented on the group of feature maps element-wise, generated from the convolutional and max-pooling layers. ReLU, defined as f(x) = max(0, x) is a simpler and quicker activation compared to other available activation functions. If we compare CNN to traditional Neural Networks inputs, it looks to be a 2D image or a 3D volume. Therefore, input properties are not dropped while they are translated to vector forms. CNN has a shared weight among its neurons where Neural Network is not feasible. Tradition Neural Network lacks the pooling layers which CNN generally provides [14].

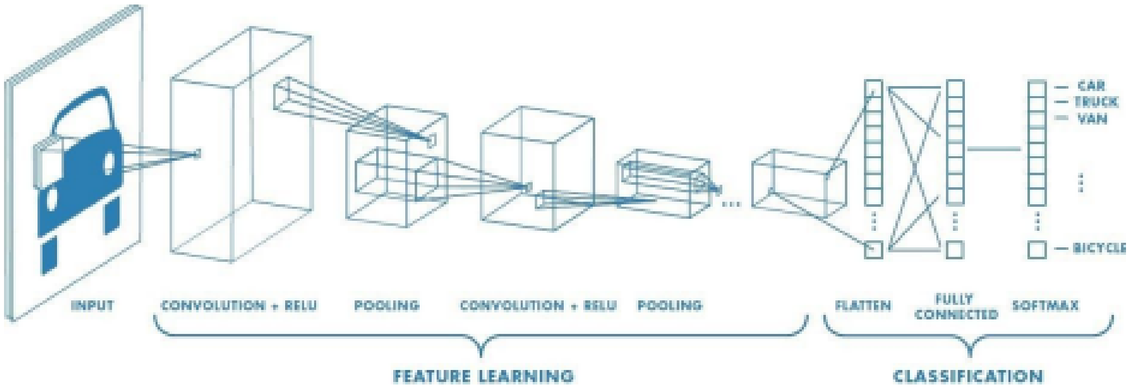

Figure 2.1: Architecture of the Convolutional Neural Network [26]

2.2 Transfer Learning

The Transfer Learning method is a widely used technique in Machine Learning. It is adept at handling big and demanding datasets upon which we train our Deep Learning models [25]. The deep Learning method has proved to be revolutionary in Biomedical Science as the field involves extensive datasets. However, we can overcome the limitations by using Transfer Learning techniques, Data Augmentation or Generative Methods [23]. Transfer learning is efficient in reducing the issue of inapt training datasets. The trained attributes of Transfer Learning in the lower layers of the network are non-specific, which, therefore, can be stored after the pre-training [13]. Transfer Learning is suitable in handling implementations where we have a scarcity of training datasets [10].

2.2.1 ImageNet

ImageNet is an open source database consisting of thousands of pictures with classes. It is a reserve of more than a million images which can be used to supplant the classification layer [18]. Researchers of this era rely heavily on ImageNet as it is a vital cog in training Machine Learning and Deep Learning models. In addition, it can also be used to train a system with a specific task or assignment such as plant leaf discrimination. According to Abbasi et al (2020), ImageNet helps equip the first layers of the system to distinguish the generalizable options from the larger datasets, which in turn enables the latter layers of the network to enlist the precise details of the smaller dataset for the transformed model [19]. According to Wang et al (2018), here is a list of CNN models which were used to extricate the features from the datasets: MobileNet, DenseNet121, VGG16 and VGG19. We can collect the weights from the ImageNet dataset by the means of the transfer learning method rather than training the datasets from the very beginning [20].

2.2.2 VGG16

Visual Geometry Group (VGG16) is a model which happens to be a specific architecture of CNN. The model was invented at the University of Oxford back in 2014. It is one of the best vision models of Convolutional Neural Networks (CNN) [12]. Visual Geometry Group (VGG16) is a model which happens to be a specific architecture of CNN. VGG16 performs better than all the other networks in Area Under the Receiver Operating System (AUROC). The frameworks with CRFs do better than XmasNet in training and testing while comparing AUROC values acquired by XmasNet and CRF-XmasNet [22]. VGG16 architecture has an input layer of (224, 224), with three channels for RGB. The pictures pass through a stack of convolutional layers, with a pixel size of 3x3 and a stride of 1. The purpose of the spatial padding of CONV2d layer input is to keep the spatial resolution preserved, even after the convolution. Following this, the CONV2d layers follow the Max Pooling operation over a 2x2 pixel window and stride of 2 [15]. In total, three fully connected layers, in addition to a stack of convolutional layers, are used in a VGG16 model. The first two of the three layers have 4096 channels each, with the last one having a thousand with activation softmax. The hidden layers of the VGG16 model have activation ReLU equipped inside [3].

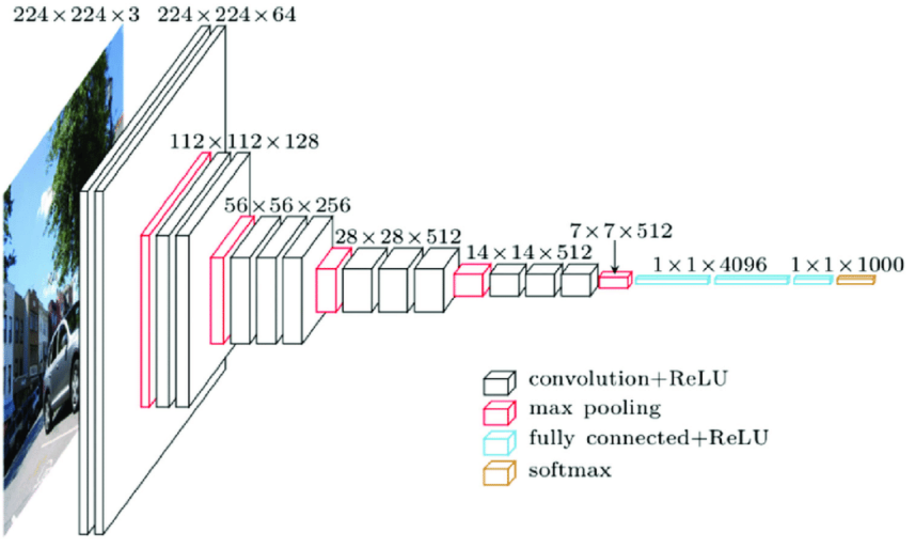

Figure 2.2: Architecture of the VGG16 [27]

2.2.3 MobileNet

Lopez et al (2020) used different methods with their acquired dataset to compare the precision of the system [20]. After using MobileNet with the SVM classifier, they have recorded accuracy very close to our model, with a difference of only 0.9%. Although, if the execution time is calculated to process a WSI, their model proves to be 75% faster than MobileNet with Machine Learning classifier. It is an important factor to consider while working alongside real-time CAD systems [18]. The MobileNet model is dependent on depthwise separable convolutions, which factorizes a standard convolution into a depthwise convolution. The depthwise convolution implements a single filter for all the input channels separately. A pointwise convolution applies a 1×1 convolution to merge the results with the depthwise convolution. A standard convolution filters and combines inputs into a new set of results in a single step. The depthwise separable convolution cuts it into two distinct layers for filtering and combining. This process of factorization can significantly dwindle the computation time and model size. All layers follow a batch normalization and ReLU nonlinearity except the fully connected layer as it has no nonlinearity and takes up the softmax layer for classification purposes [6]. Figure 3 depicts a layer with regular convolutions, batch norm and ReLu nonlinearity to the factorized layer with depthwise convolution. It also shows a 1×1 pointwise convolution, in addition to the batch norm and ReLu after each layer [7].

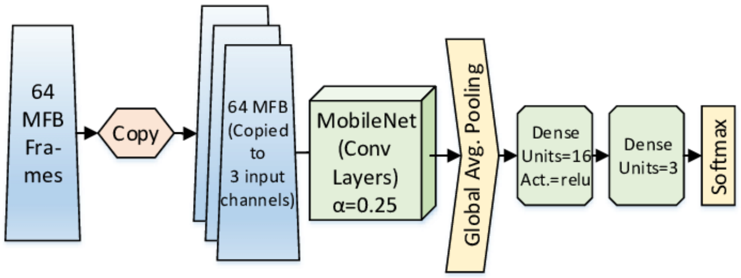

Figure 2.3: Architecture of the MobileNet [30]

2.2.4 ResNet

The ResNet model is a linchpin to drawing out features from input images. Extracted feature maps are loaded into two separate branches. The Region Proposal Network (RPN) generates suggestions on the left side of the branch to determine which regions the Grading Network Head (GNH) should keep in its focal point. The GNH is mainly responsible for assigning Gleason graded to the epithelial cell areas [5]. On the right side of the branch, Epithelial Network Head (ENH) is used to locate any potential epithelial tissue in the image. The final result is dependent on the outcomes of the ENH. The model churns out an image as stroma if there are no epithelial cells found. Conversely, if any epithelial cells are detected, the model renders its results from the GNH [16]. Results suggest the models that focus on base architectures (i.e., CRF-RNN) and deep networks (i.e., VGG16 and AlexNet) render far better outcomes to architectures such as ResNet and XmasNet. ResNet is a complex network with a limited number of training resources available to it. It is one of the main reasons for ResNet's poor accuracy compared to other network models [25]. Most convolutional layers tend to have 3×3 filters. The layers have the same number of filers for the same output feature size. Also, if the map size gets cut in half, the number of filters is doubled to preserve the time complexity per layer. Downsampling can be directly implemented on convolutional layers that have a stride of 2. The network ends with an average pooling layer and a thousand way fully connected layer alongside softmax [7].

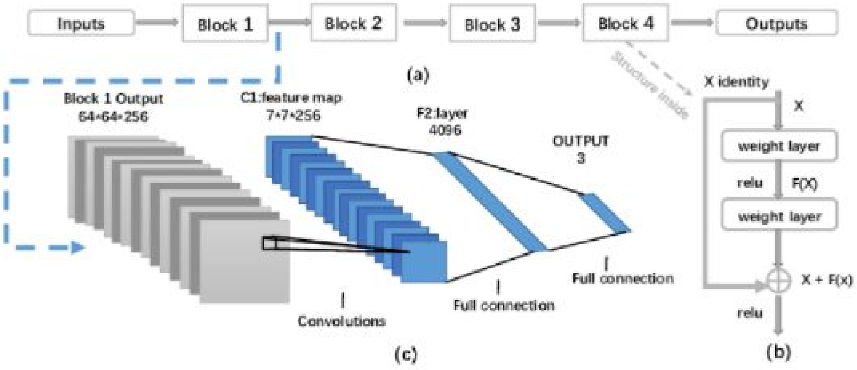

Figure 2.4: Architecture of the ResNet [29]

2.3 Machine Learning Classifiers

Machine Learning classifiers are a system of supervised learning where targets are equipped with input data. Applications of machine learning classifiers are widespread as it is vastly used in medical prognosis, credit approvals and target marketing. Classifiers are divided into two separate categories; lazy learners and eager learners. The training data of ML classifiers are extracted from prior works while awaiting testing datasets [1]. The classification is accomplished on the relevance of the new datasets provided, compared to the training datasets. The training is carried out by erecting a classification model, based on the given training datasets. The prediction is generated based upon the test dataset rendered from the constructed model. The training time is significantly less in Lazy Learning compared to testing time whereas it works in the reverse in the case of Eager Learning [8].

2.3.1 Random Forest Classifier

Random Forest Classifiers (RFC) were developed to improve the performance of Classification and Regression Trees (RFC). RFC entails an ensemble of Classification and Regression Trees, trained on an arbitrary subset of training features and datasets. Bootstrap Aggregation is a classic example of a Random Forest Classifier [11]. The final classification is produced by an amalgamation of all existing trees within the RFC, likewise a majority vote. RFCs are regarded mostly as Ensemble Classifiers (EC) that can take advantage of multiple weak learners to derive a conclusion. Intersection Over Union (IOU) of Random Forest Model for “Low Grade” is computed by combining both low and high grades together. Our proposed path R-CNN has achieved the highest performance in both the single class evaluation and the four-class Mean Intersection Over Union (MIOU) [23].

2.3.2 Gradient Boosting Classifier

Gradient Boosting Classifiers are Machine Learning algorithms that collect weak learning models to create a strong prognostication model. The boosting algorithm is used to reduce the bias error of the model. It is also commonly referred to as the Greedy Algorithm. Gradient Boosting Machines can be used on binary classification problems, multi-class classification problems and even regression problems. The model is used for binary classification of the image dataset with appropriate hyperparameter tuning for the research. It can procure a new way to solve the existing image classification problem. Extreme Gradient Boosting (XGBoost) is a blended learning algorithm, based on Gradient Boosting. It is mainly used to accomplish precise classification outputs through repetitive calculations of weak classifiers. XGBoost is frequently applied in various domains because of its expediency and exactness [11].

2.3.3 SVM Classifier

Among the various machine learning classifiers, such as SVM Gaussian, SVM RBF and SVM Polynomial, Jain et al (2001) achieved the highest performance using different combinations of features. (i.e., texture plus morphological, EFDs plus morphological using SVM Gaussian). Murray et al (2014) have used R-CNN and Fast R-CNN to extract features from their images. They have also introduced a ROIPooling layer to combine the feature scale of all images [12]. Finally, they removed the softmax function with the SVM classifier to merge classification and border regression which helped them in enhancing the accuracy and calculation time [22].

2.4 Challenges of Training Deep Learning Models

2.4.1 Overfitting

When it comes to validation, one of the most frequent problems faced in Machine Learning is that the classifier constitutes the input data vaguely and comprehensively, which does not translate into new samples. This particular issue is known as overfitting. A classifier needs to be confirmed on a set of unrevealed samples to avoid such occurrence. It can be executed by following the "holdout" method, where authors Irvin et al (2017) differentiated a subgroup of samples that are used in the ending classifier [26]. Also, among other ways to implement such models are K-Fold Cross-Validation (KFCV) and Leave-One Out Cross-Validation (LOOCV). In KFCV, the training sets are distinguished in K subgroups. These subgroups of data validate a classifier trained on different subsets. LOOCV is a special case of KFCV, where K equates the number of independent samples in a given training set. If there is an inadequate number of images available, it will produce a model ailing from overfitting. The data augmentation technique was used to expand the training dataset. This technique enhanced sturdiness and reduced the overfitting problem [13].

2.4.2 Training Time

As the algorithm offered in this paper has a high degree of complexity in its computation of the ROIAlign layer, the total time consumed is higher than the FCN algorithm. The time required to train datasets hinges upon many different factors such as hyperparameters and the choice of optimizers. Therefore, the training and prediction times are not the ultimate benchmarks on the networks [16]. Using the batch normalization layer reduces the time needed for training purposes. It permits the Convolutional Neural Network to utilize high learning rates to regulate the weightage issue [21].

2.4.3 Gradient Vanishing

Recurrent Neural Networks (RNNs) have a struct that should make them adept at handling long-term dependencies. However, Vanilla Recurrent Neural Network significantly struggles from Gradient Vanishing problems. Therefore, it has difficulties in learning the dependencies in the long sequences [21]. Hinton et al (2006) proposed in their paper a greedy technique, which is similar to the Restricted Boltzmann machine. In this technique, they trained one layer at a time. This method averted the Gradient Vanishing problem and created a gateway for deeper networks [19]. Deeply supervised layers can help to improve the learning ability of hidden layers. According to Zhu et al, adding eight additional layers in the network vastly improved the efficacy of the hidden layers. These supervised layers can help to oversee the process of training. Also, since the network entails a huge depth, additional supervised layers can help in propagating the gradients through preservation from an early stage [14] .

Chapter 3

Dataset Handling, Implementation and Research Methodology

3.1 Dataset

The data used in this model is collected from 'The Cancer Imaging Archive (TCIA)', the dataset is publicly available at the wiki cancer imaging archive. From this dataset, we have used MRI images of 678 patients [28].

3.2 Data Pre-processing

The "Target Data_2019-12-05.xlsx" was analyzed to split the images into "Malignant" and "Benign", based on the UCLA scores provided. Any score greater than 2 is considered a risk of Malignant and the images are extracted in a malignant folder to series UID. A similar thing is done for Benign for a UCLA score of less than 2. We have used Python OS Library to iterate over the directories to find a path to MRI images in dicom format. Subsequently, we enlisted the MRI images against their series instance UID. A general CSV file was generated for all the datasets. The file named "df_patient_MRI_details.csv" entails columns of data labelled as 'Patient ID', 'Series Instance UID' and 'DicmPath'. DicmPath contains the path to each dicom image against their PatientID and SeriesInstanceUIC which was generated using an OS library. Initially assigned SeriesInstanceUID for Malignant and Benign was originally extracted from the "df_patient_MRI_details.csv" file as two additional CSV files were generated for Malignant and Benign patients having similar columns. Two newly obtained datasets and CSV files were loaded into two separate data frames. We took two variables to load the 'DicmPath' column which contained the location of each dicom image. Subsequently, it was converted to a list. We imported CV2 and NumPy in addition to importing images from the PIL Library. Two separate folders were created to extract the images in PNG format from the dicom files and the pydicom library was used to execute this operation. We worked with 5158 MRI images in total where we had an equal number of images for the two classes, benign and malignant. The train and validation split was done on 4126 images with the ratio of 0.2 for validation. We kept a total of 1032 image data for testing which was kept separate for prediction using our model.

3.2.1 Reshaping

For VGG16, we reshaped the images to keep the data intact to find a better accuracy. After reshaping the image, we found our dimensions to be (224, 244). Also, we added Channel 3 to the input shape as (224, 224, 3) it is the standard input shape for VGG16 model architecture. We restructured the image to 16*16 sizes and added Gabor filters to each image in Machine Learning classifiers. It aided in texture analysis of images. 1D convolve, Fourier-ellipsoid, Fourier-gaussian and 1D Gaussian filters were recorded using Gabor filters. Parameters for Gabor filters were used appropriately for our dataset.

3.2.2 Splitting

We imported the ImageDataGenerator to load our dataset using the image_dataset_from_directory, which provides the pathway to training and testing data to infer Benign and Malignant labels. After passing the training labels, we set the validation labels to the (to_categorical) function that converts the class to a binary class matrix. After all the necessary preprocessing, Class "1" indicates Maniglant and Class "0" indicates Benign in data visualization.

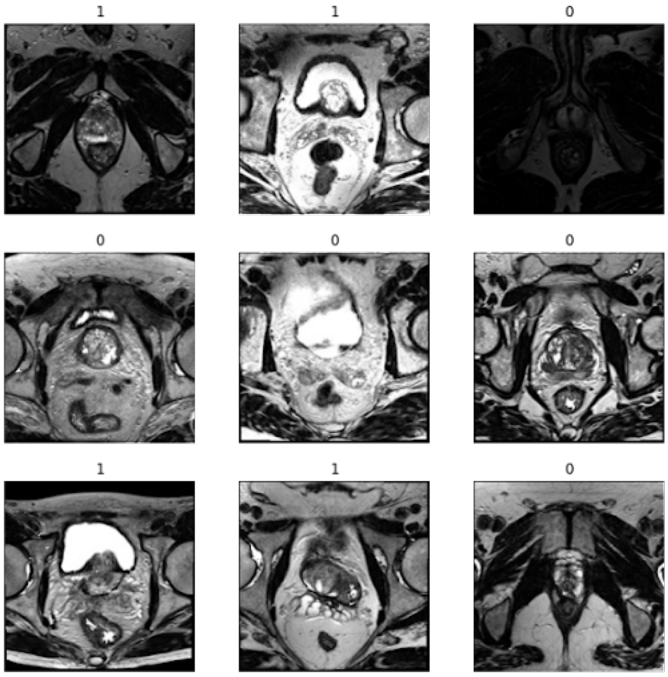

Figure 3.1: Labelled MRI dataset after Preprocessing.

We saved our datasets in TFRecords format for optimal performance and expediency. Each record contained the information of 323 images. We also set the experimental deterministic to false to allow order-altering optimizations. In addition, we used 13 records for training where 65 batches were used to train each epoch, 15 batches were used for every validation run and 3 records were used for testing.

3.3 Implementation of ML Classifier

We have used Random Forest, Gradient Boosting and Support Vector Machine Algorithms for MRI image classification. We also added Gabor Filters for texture analysis of each image after preprocessing. After performing the necessary hyperparameter tuning of each ML classifier, we have performed training on the MRI image datasets. We obtained the prediction accuracy and classification report after concluding our training.

3.4 Hyperparameter Optimization

3.4.1 N_estimators

N_estimators define the number of trees we wish to assign to our model. Having a higher number of trees tends to render better accuracy. However, adding a large number of trees can slow down the overall computation time while training. It is imperative to build the complete tree before taking the maximum voting or averages of predictions. We conducted the best fit parametric search on the random forest to increase the efficiency of our model. We set the Gradient Boosting classifiers with n_estimator range (1 to 200) and fit best fit our dataset. In addition, we also recorded the best number of n_estimator used in our model for classification. We found the best n_estimator to be 100 for both the classifiers in our dataset.

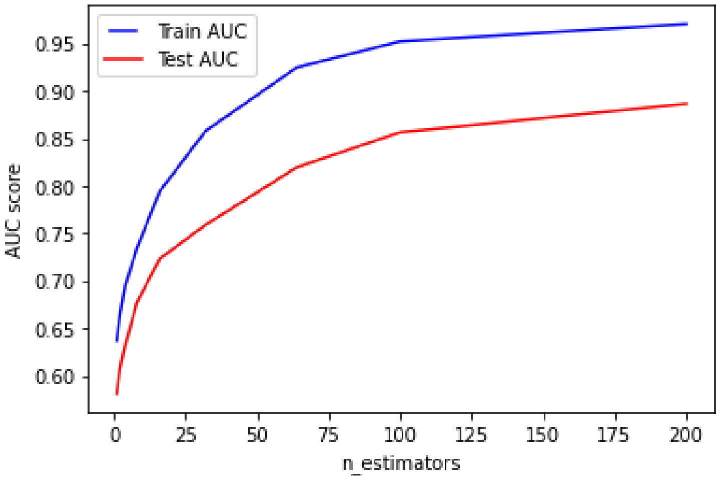

Figure 3.2: Hyperparameter tuning of n_estimator for Gradient Boosting

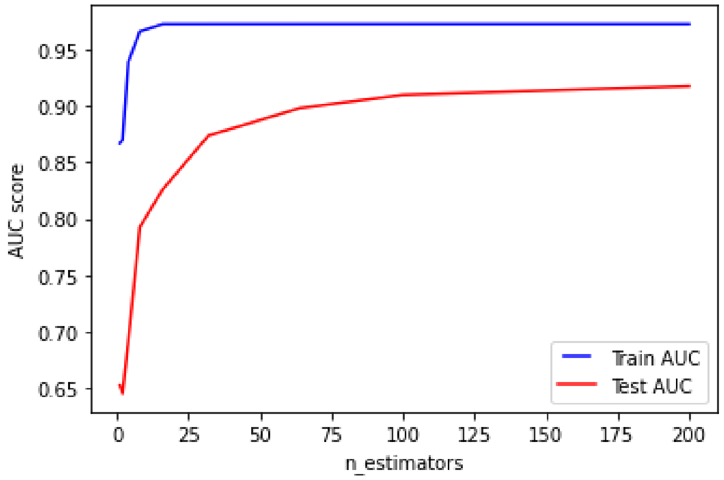

Figure 3.3: Hyperparameter tuning of n_estimator for Random Forest

3.4.2 Learning Rate

Gradient Boosting often exhibits a common problem while being at use. It tends to learn the training data quickly and ends up overfitting the dataset. Learning Rate is used as a speed bump to control the speed of learning in Gradient Boosting models. Gradient Boosting creates trees to the existing model in a sequential manner. It helps eradicate prediction and residual errors by adding new trees. We have used a Parametric search on the Gradient Boosting classifier to make our model more convenient for the dataset. We set the range to 0.5 to 0.01 to best fit our dataset and recorded the best value of 0.01 to use in our model for classification.

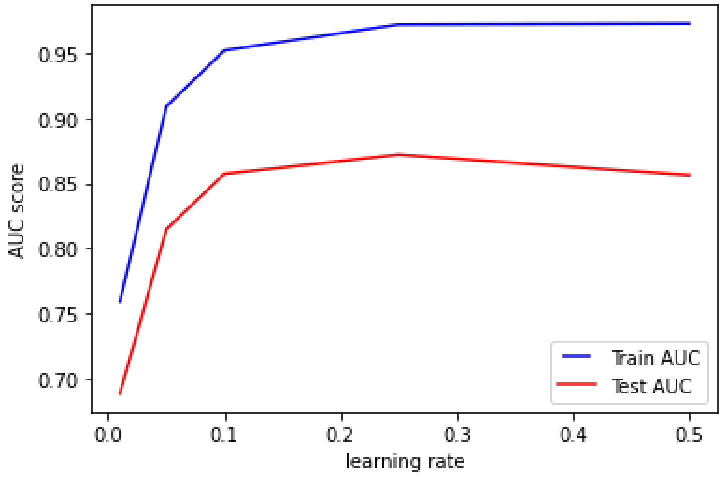

Figure 3.4: Hyperparameter tuning of learning rate for Gradient Boosting

3.4.3 Tree Depth

Depth in Random Forest indicates how the model will perform with the data; a deeper mean tree can split and capture more information than trees with low depth. We set the tree depth by the same process of parametric search to determine the required depth of each tree in our Random Forest model. We have found our model to give consistent training and validation accuracy at max_depth of 30 from the AUC score.

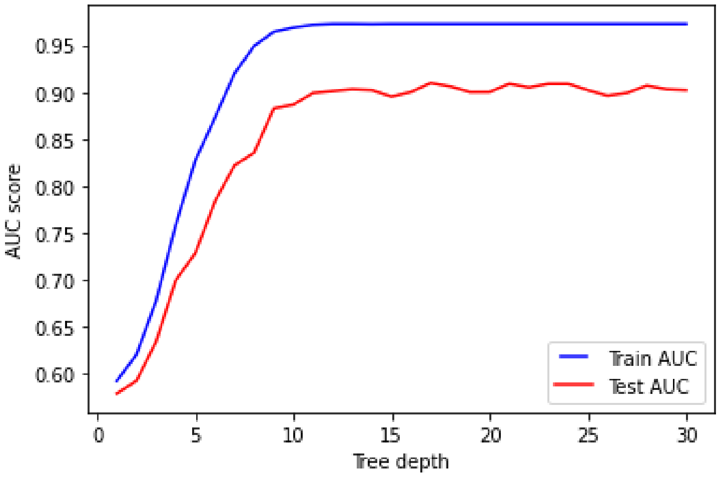

Figure 3.5: Hyperparameter tuning of max_depth for Random Forest

3.5 Implementation of VGG16

We have used ImageNet based CNN architecture, such as MobileNet, ResNet and VGG models to train our image datasets. We maintained an equal weight to that of ImageNet for our training. Afterwards, we recorded the Loss Vs Accuracy for both training and validation run and chose the best model following the output. We have found VGG16 to be the optimum model for implementation. In addition, we added the hidden layer and optimizer tuning to use in our fusion method.

3.5.1 Adjustment to the Model and using transfer learning

We followed VGG16 architecture to implement the model based on our dataset and made the following adjustments.

1. We loaded the VGG16 model from Keras API, initialized the weights to ImageNet and set the input shape to standard (224,224,3). We also removed the top layer and set the trainable to false.

2. We will be using the pertained model of VGG16, with its weights equal to ImageNet and extract features from our images.

    - Vgg16 functional Layer type input
    - Flatten:
    - Dense: unis: 2048, activation: relu
    - Dense: unis: 1024, activation: relu
    - Dense: unis: 512, activation: relu
    - Dense: unis: 256, activation: relu
    - Dropout: (0.05)
    - Dense: unis: 128, activation: relu
    - Dropout: (0.05)
    - Dense: units: 64, activation: relu
    - Dropout: (0.05)
    - Dense: units: 32, activation: relu
    - Dropout: (0.05)
    - Dense: 2, activation: softmax

We have set the activation function for each convolutional and hidden layer to Rectified Linear Unit (ReLU), except for the output layer. Additionally, we used Softmax activation for our last output layer. We compiled the model with Adam optimizer to compile the model and kept the loss function as ”binary_crossentropy” and metrics as “accuracy”. We implemented a dropout ratio of 0.05 to overcome the overfitting problem.

3.5.2 Activation function, Optimization Algorithm and Loss Function

We have used activation functions such as ReLU and Softmax for image classification tasks while building the model. ReLU performed the best for solving the gradient vanishing problem in addition to performing faster computation. For our final layer, we have used the Softmax activation function, as VGG16 was trained for multi-classification output and ImageNet dataset. However, it worked well even for our binary classification. As the probability of positive and negative results summed up to 1 for image classification, we used Softmax due to the compatibility in results. The best fit optimizer used in our model was the Adam optimization algorithm, which is an extension of stochastic gradient descent. It has good computational efficiency and less memory consumption. Adam optimizer was well structured in dealing with a high number of parameters and the complexity in datasets. We also set the learning rate to 1e-3 and matrics to accuracy. We used the Binary Cross-Entropy for a better fit with our optimizer and the loss was calculated as:

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i . \log(p(y_i)) + (1 - y_i) . \log(1 - p(y_i))\Pi \tag{3.1}$$

3.6 Methodology Used for our VGG and ML Fusion

After appropriate data preprocessing and train and test split, we used the VGG16 pre-trained model to extract features from our test dataset. After feature extraction and reshaping the features for input to ML classifiers, we have trained the classifiers on the extracted features. We used the hyper tuned ML classifiers, Random Forest, Gradient Boosting and SVM classifiers to train the dataset. Afterwards, we used the VGG16 model to extract the features on the test set. After reshaping the test features, we predicted test features and recorded the accuracy. A comprehensive workflow of the methodology is given below:

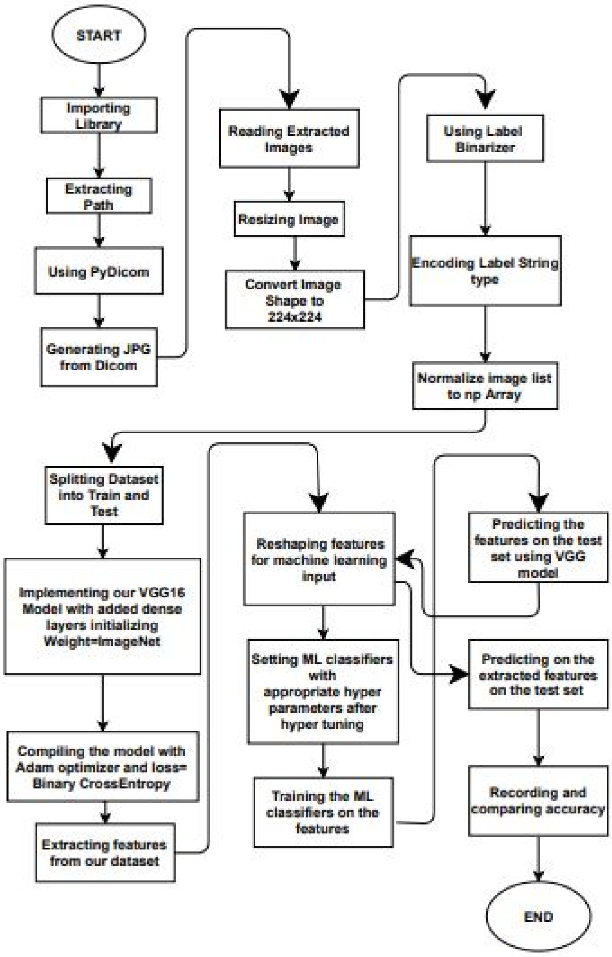
START
Importing Library
Extracting Path
Using PyDicom
Generating JPG from Dicom
Reading Extracted Images
Resizing Image
Convert Image Shape to 224x224
Using Label Binarizer
Encoding Label String type
Normalize image list to np Array
Splitting Dataset into Train and Test
Implementing our VGG16 Model with added dense layers initializing Weight=ImageNet
Compiling the model with Adam optimizer and loss= Binary CrossEntropy
Extracting features from our dataset
Reshaping features for machine learning input
Setting ML classifiers with appropriate hyper parameters after hyper tuning
Training the ML classifiers on the features
Predicting the features on the test set using VGG model
Predicting on the extracted features on the test set
Recording and comparing accuracy
END

Figure 3.6: Feature extraction using VGG16 and classification done by ML classifiers

Chapter 4

Result Analysis

4.1 Machine Learning Classification

After implementation of our ML classification model we got the following accuracy for each of the ML classifiers and their fit time and prediction time was also recorded. Given in the table below:

Classifier	Accuracy	Fit time (s)	Prediction time (s)
Gradient Boosting Classifier	0.8730	1479.6313	0.3087
Rendom Forest Classifier	0.8672	37.7754	1.0045
SVM Classifier	0.7722	242.7738	131.3345

Table 4.1: Accuracy for our selected ML classifier without hyperparameter optimization

Machine Learning classifiers with Precision, Recall and F1 scores are shown in the following table without hypertuning:

	precision	recall	f1-score	support
Benign	0.86	0.89	0.87	516
Malignant	0.88	0.86	0.87	516
Accuracy			0.87	1032
Macro avg	0.87	0.87	0.87	1032
Weighted avg	0.87	0.87	0.87	1032

Table 4.2: Classification report of Gradient Boosting without hyperparameter optimization

	precision	recall	f1-score	support
benign	0.87	0.87	0.87	516
malignant	0.87	0.86	0.87	516
accuracy			0.87	1032
macro avg	0.87	0.87	0.87	1032
weighted avg	0.87	0.87	0.87	1032

Table 4.3: Classification report of Random Forest without hyperparameter optimization

	precision	recall	f1-score	support
benign	0.76	0.80	0.78	516
malignant	0.79	0.74	0.76	516
accuracy			0.77	1032
macro avg	0.77	0.77	0.77	1032
weighted avg	0.77	0.77	0.77	1032

Table 4.4: Classification report of SVM without hyperparameter optimization

4.2 Transfer Learning approach with VGG16, MobileNet, ResNet

We have conducted the training and prediction on our image data using the transfer learning approach with equal weights to ImageNet. We found the following Accuracy Vs Loss graph of training and validation.

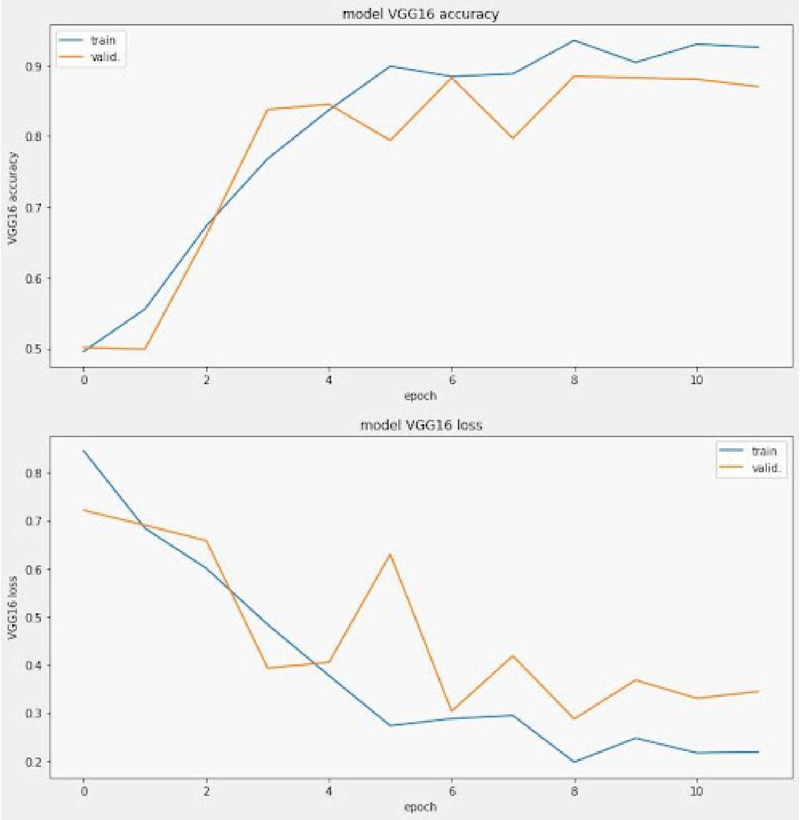

Figure 4.1: Training and validation loss and accuracy of VGG16

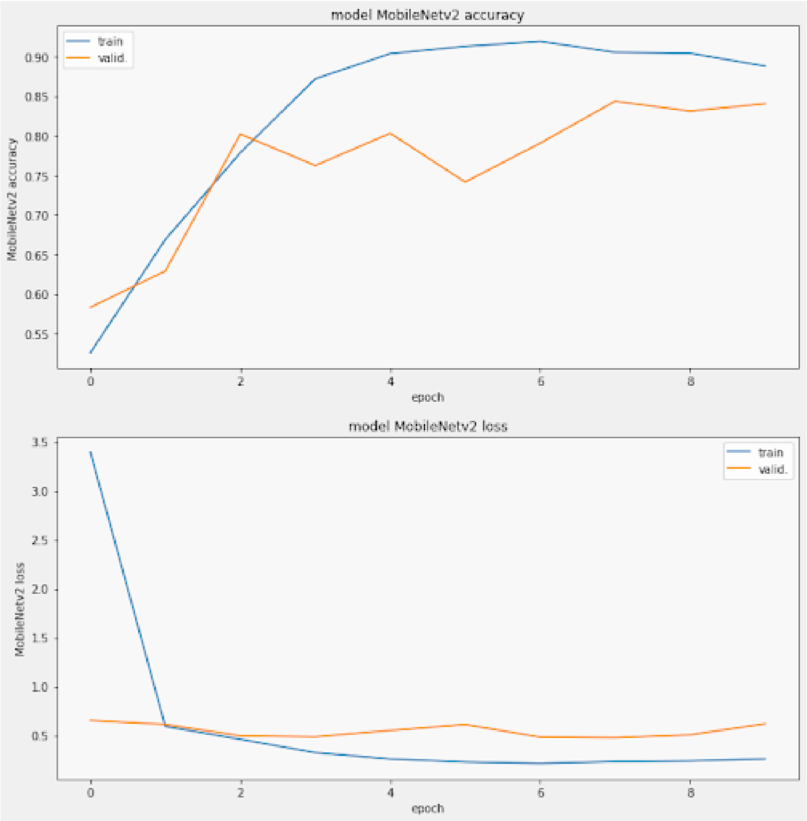

Figure 4.2: Training and validation loss and accuracy of MobileNet

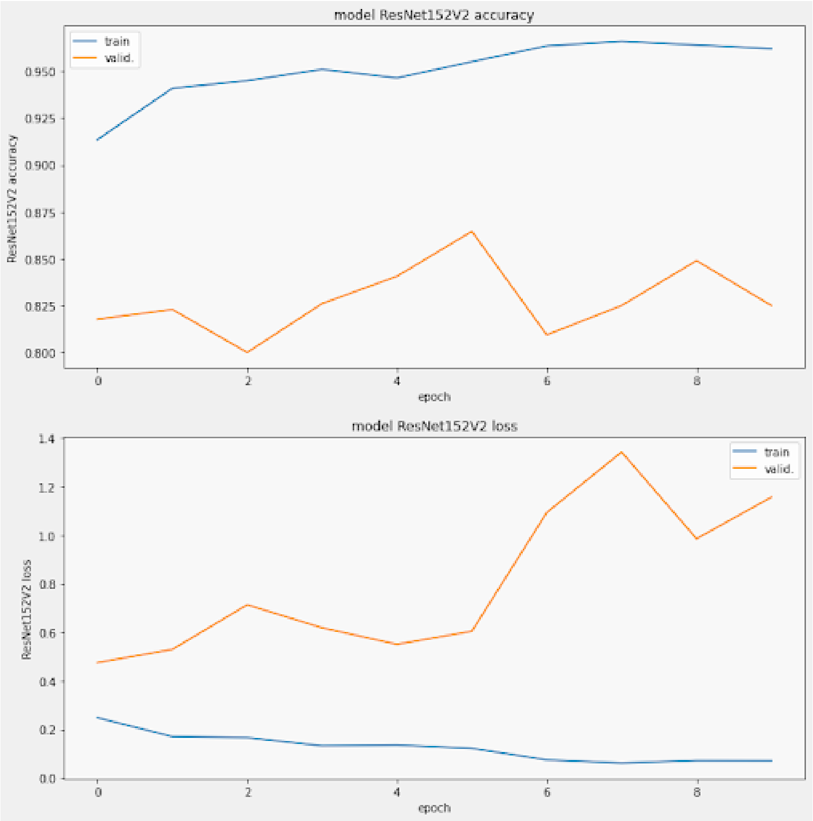

Figure 4.3: Training and validation loss and accuracy of ResNet

We have found VGG16 to be the most accurate and precise model for our image classification problem. Our obtained accuracy got better with every epoch for training and validation. The only drawback we have witnessed from our model was the higher training and validation time compared to the other two architectures. Although MobileNet was fast in training and validation, it rendered overfitting problems which resulted in a low validation accuracy. Likewise, the ResNet model had significantly high training accuracy but it faltered in validation score due to overfitting as well. Therefore, we chose the VGG16 as the ideal candidate for extracting features from images. The mean validation accuracy for three separate models is shown below in a Bar Chart.

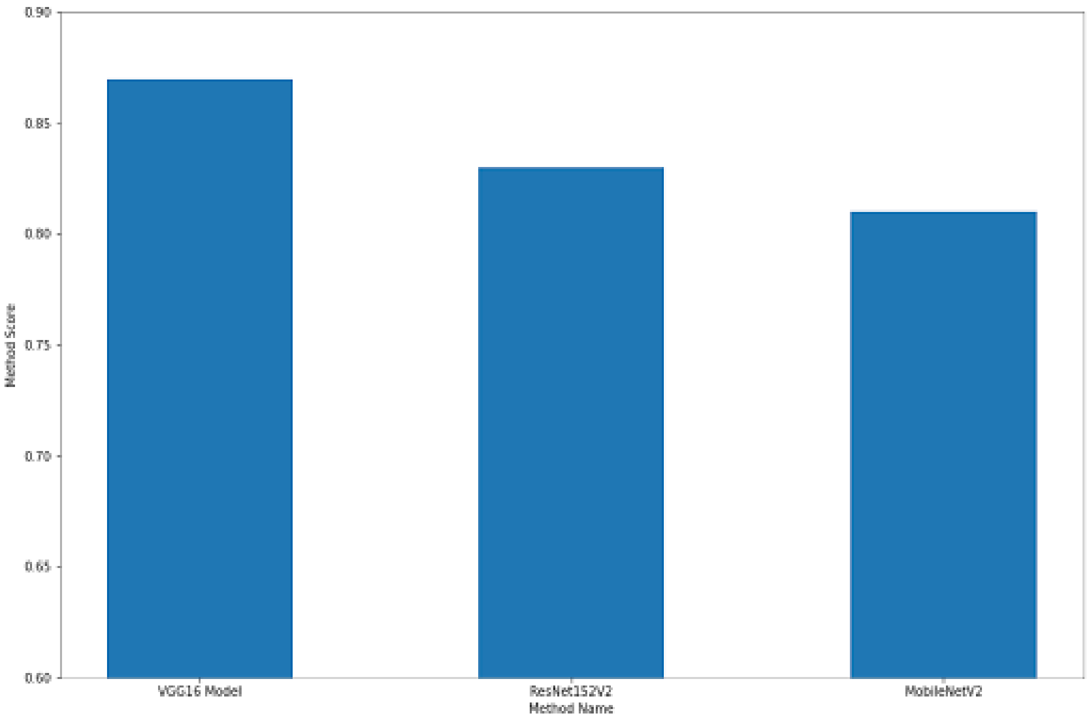

Figure 4.4: Mean validation accuracy of VGG16 compared to other pretrained models

4.3 Result Analysis of our Deep ML Fusion Model

After using VGG16 as our feature extraction layer, the obtained outputs were forwarded to our existing hyper tuned ML classifiers. We found notable improvements in accuracy and predictions. We saw a significant rise in Precision, Recall and F1 Score.

Classifier	Accuracy	Fit Time (s)	Prediction Time (s)
Gradiant Boosting Classifier	0.9166	5402.2925	1080.4585
Random Forest Classifier	0.9079	25.2677	0.1554
SVM CLassifier	0.8507	385.6882	95.7803
Decision Tree CLassifier	0.73643	47.28203	0.03991

Table 4.5: Improved accuracy of our Deep ML model

	precision	recall	f1-score	support
0	0.91	0.92	0.92	520
1	0.92	0.91	0.92	512
accuracy			0.92	1032
macro avg	0.92	0.92	0.92	1032
weighted avg	0.92	032	0.92	1032

Table 4.6: Classification report of Gradient boosting as classification layer in Deep ML network

	precision	recall	f1-score	support
0	0.91	0.91	0.91	520
1	0.91	0.91	0.91	512
accuracy			0.91	1032
macro avg	0.91	0.91	0.91	1032
weighted avg	0.91	0.91	0.91	1031

Table 4.7: Classification report of Random Forest as classification layer in Deep ML network

	precision	recall	f1-score	support
0	0.82	0.90	0.86	520
1	0.88	0.80	0.84	512
accuracy			0.85	1032
macro avg	0.85	0.85	0.85	1032
weighted avg	0.85	0.85	0.85	1032

Table 4.8: Classification report of SVM as classification layer in Deep ML network

Chapter 5

Conclusion

5.1 Conclusion

Prostate cancer is responsible for claiming thousands of lives each year all around the globe. Early and efficient prognosis can be an efficient way of reducing the mortality rate of prostate cancer significantly. Our research has aimed to develop an effective classification method of prostate cancer images by using Deep Convolutional Networks with Machine Learning Classifiers. We put a special emphasis on thoroughness as we tried to obtain the highest level of accuracy and validation score possible from our model. We earnestly hope that our work would bring along a few noteworthy novelties in this particular research field and inspire others to add their contributions to the vast explorations of prostate cancer.

5.2 Challenges

Working with a large dataset was a challenge we had to meet during our research. Also, due to the Covid-19 pandemic, it was hard to collect the dataset we wished to use for our thesis. Although we initially struggled to find the large amount of data required to run the models; however, in the end, after going through a lot of resources, we managed to gather the dataset best suited for our thesis work. Lastly, it was hard to get good accuracy using the weights of ImageNet, even with hyper tuning and adjustments to the dense layers due to the dissimilarity in datasets used to train the pre-trained model. Nonetheless, we were able to get acceptable accuracy using the Deep ML method we used.

5.3 Future Work Plan

We have largely focused on the architecture of a CNN model for transfer learning, such as VGG16, ResNet, MobileNet in our paper along with Machine Learning classifiers. However, our future approach should be extended to large MRI images datasets with Federated Learning. The Federated Learning technique might be more efficient in detecting prostate cancer. We aim to see how well it works with Federated Learning and bring additional improvements to our work.

Bibliography

[1] K. Clark, B. Vendt, K. Smith, J. Freymann, J. Kirby, P. Koppel, S. Moore, S. Phillips, D. Maffitt, M. Pringle, *et al.*, “The cancer imaging archive (tcia): Maintaining and operating a public information repository,” *Journal of digital imaging*, vol. 26, no. 6, pp. 1045–1057, 2013.

[2] G. A. Sonn, S. Natarajan, D. J. Margolis, M. MacAiran, P. Lieu, J. Huang, F. J. Dorey, and L. S. Marks, “Targeted biopsy in the detection of prostate cancer using an office based magnetic resonance ultrasound fusion device,” *The Journal of urology*, vol. 189, no. 1, pp. 86–92, 2013.

[3] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.

[4] S. Nouranian, M. Ramezani, I. Spadinger, W. J. Morris, S. E. Salcudean, and P. Abolmaesumi, “Learning-based multi-label segmentation of transrectal ultrasound images for prostate brachytherapy,” *IEEE transactions on medical imaging*, vol. 35, no. 3, pp. 921–932, 2015.

[5] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[6] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.

[7] J. T. Kwak and S. M. Hewitt, “Nuclear architecture analysis of prostate cancer via convolutional neural networks,” *IEEE Access*, vol. 5, pp. 18 526–18 533, 2017.

[8] X. Ren, H. Guo, S. Li, S. Wang, and J. Li, “A novel image classification method with cnn-xgboost model,” in *International Workshop on Digital Watermarking*, Springer, 2017, pp. 378–390.

[9] Y. Tsehay, N. Lay, X. Wang, J. T. Kwak, B. Turkbey, P. Choyke, P. Pinto, B. Wood, and R. M. Summers, “Biopsy-guided learning with deep convolutional neural networks for prostate cancer detection on multiparametric mri,” in *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, IEEE, 2017, pp. 642–645.

[10] X. Yang, C. Liu, Z. Wang, J. Yang, H. Le Min, L. Wang, and K.-T. T. Cheng, “Co-trained convolutional neural networks for automated detection of prostate cancer in multi-parametric mri,” *Medical image analysis*, vol. 42, pp. 212–227, 2017.

[11] Q. Zhu, B. Du, B. Turkbey, P. L. Choyke, and P. Yan, “Deeply-supervised cnn for prostate segmentation,” in *2017 international joint conference on neural networks (IJCNN)*, IEEE, 2017, pp. 178–184.

[12] T. Fang, “A novel computer-aided lung cancer detection method based on transfer learning from googlenet and median intensity projections,” in *2018 IEEE international conference on computer and communication engineering technology (CCET)*, IEEE, 2018, pp. 286–290.

[13] W. Li, J. Li, K. V. Sarma, K. C. Ho, S. Shen, B. S. Knudsen, A. Gertych, and C. W. Arnold, “Path r-cnn for prostate cancer diagnosis and gleason grading of histological images,” *IEEE transactions on medical imaging*, vol. 38, no. 4, pp. 945–954, 2018.

[14] I. Reda, B. O. Ayinde, M. Elmogy, A. Shalaby, M. El-Melegy, M. Abou El-Ghar, A. Abou El-fetouh, M. Ghazal, and A. El-Baz, “A new cnn-based system for early diagnosis of prostate cancer,” in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, IEEE, 2018, pp. 207–210.

[15] M. N. N. To, D. Q. Vu, B. Turkbey, P. L. Choyke, and J. T. Kwak, “Deep dense multi-path neural network for prostate segmentation in magnetic resonance imaging,” *International journal of computer assisted radiology and surgery*, vol. 13, no. 11, pp. 1687–1696, 2018.

[16] Y. Wang, B. Zheng, D. Gao, and J. Wang, “Fully convolutional neural networks for prostate cancer detection using multi-parametric magnetic resonance images: An initial investigation,” in *2018 24th International Conference on Pattern Recognition (ICPR)*, IEEE, 2018, pp. 3814–3819.

[17] Y.-Y. Zhang, Q. Li, Y. Xin, and W.-Q. Lv, “Differentiating prostate cancer from benign prostatic hyperplasia using psad based on machine learning: Single-center retrospective study in china,” *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 16, no. 3, pp. 936–941, 2018.

[18] M.-T. Nguyen, D. H. Le, T. Nakajima, M. Yoshimi, and N. Thoai, “Attention-based neural network: A novel approach for predicting the popularity of online content,” in *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, IEEE, 2019, pp. 329–336.

[19] A. A. Abbasi, L. Hussain, I. A. Awan, I. Abbasi, A. Majid, M. S. A. Nadeem, and Q.-A. Chaudhary, “Detecting prostate cancer using deep learning convolution neural network with transfer learning approach,” *Cognitive Neurodynamics*, vol. 14, no. 4, pp. 523–533, 2020.

[20] L. Duran-Lopez, J. P. Dominguez-Morales, A. F. Conde-Martin, S. Vicente-Diaz, and A. Linares-Barranco, “Prometeo: A cnn-based computer-aided diagnosis system for wsi prostate cancer detection,” *IEEE Access*, vol. 8, pp. 128 613–128 628, 2020.

[21] N. N. C. Institute, “Cancer facts figures 2020.,” *CA Cancer J. Clin.*, pp. 1–76, 2020.

[22] P. Lapa, M. Castelli, I. Gonçalves, E. Sala, and L. Rundo, "A hybrid end-to-end approach integrating conditional random fields into cnns for prostate cancer detection on mri," *Applied Sciences*, vol. 10, no. 1, p. 338, 2020.

[23] R. R. Wildeboer, R. J. van Sloun, H. Wijkstra, and M. Mischi, "Artificial intelligence in multiparametric prostate cancer imaging with focus on deep-learning methods," *Computer methods and programs in biomedicine*, vol. 189, p. 105 316, 2020.

[24] Z. Liu, C. Yang, J. Huang, S. Liu, Y. Zhuo, and X. Lu, "Deep learning framework based on integration of s-mask r-cnn and inception-v3 for ultrasound image-aided diagnosis of prostate cancer," *Future Generation Computer Systems*, vol. 114, pp. 358–367, 2021.

[25] B. Almas, K. Sathesh, and S. Rajasekaran, "A deep analysis of google net and alexnet for lung cancer detection,"

[26] *Https://medium.com/@raghavprabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148*, https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148, Accessed: 2021-09-25.

[27] *Https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c: :text=vgg16%20is%20a%20convolution%20neural,vision%20model%20architecture%20till%20date*, https://towardsdatascience.com/step-by-step-vgg16-implementation-in-keras-for-beginners-a833c686ae6c#:~:text=VGG16%20is%20a%20convolution%20neural,vision%20model%20architecture%20till%20date, Accessed: 2021-09-25.

[28] *Https://wiki.cancerimagingarchive.net/pages/viewpage.action?pageid=68550661*, https://wiki.cancerimagingarchive.net/pages/viewpage.action?pageId=68550661, Accessed: 2021-09-25.

[29] *Https://www.researchgate.net/figure/a-the-structure-of-resnet-101-b-the-basic-structure-of-each-block-in-resnet-101-c*$_{f}ig1_{3}$31110894, https://www.researchgate.net/figure/a-The-structure-of-ResNet-101-b-The-basic-structure-of-each-block-in-ResNet-101-c_fig1_331110894, Accessed: 2021-09-25.

[30] *Https://www.researchgate.net/figure/network-architecture-of-mobilenet*$_{f}ig1_{3}$28654938, https://www.researchgate.net/figure/Network-Architecture-of-MobileNet_fig1_328654938, Accessed: 2021-09-25.