Deep Learning Models for Survival Analysis on Histology Images

By

Hetang Patel

A thesis submitted in partial fulfillment.
of the requirements for the degree of
Master of Science (MSc) in Computational Sciences

The Faculty of Graduate Studies
Laurentian University
Sudbury, Ontario, Canada

THESIS DEFENCE COMMITTEE/COMITÉ DE SOUTENANCE DE THÈSE
Laurentian Université/Université Laurentienne
Faculty of Graduate Studies/Faculté des études supérieures

Title of Thesis Titre de la thèse	Deep Learning Models for Survival Analysis on Histology Images
Name of Candidate Nom du candidat	Patel, Hetang
Degree Diplôme	Master of Science
Department/Program Département/Programme	Computational Sciences
Date of Defence Date de la soutenance	May 25, 2021

APPROVED/APPROUVÉ

Thesis Examiners/Examinateurs de thèse:

Dr. Kalpdrum passi
(Supervisor/Directeur(trice) de thèse)

Dr. Ratvinder Grewal
(Committee member/Membre du comité)

Dr. Peter Adamic
(Committee member/Membre du comité)

Dr. Mayuri Mehta
(External Examiner/Examinateur externe)

Approved for the Faculty of Graduate Studies
Approuvé pour la Faculté des études supérieures
Tammy Eger, PhD
Vice-President, Research (Office of Graduate Studies)
Vice-rectrice à la recherche (Bureau des études supérieures)
Laurentian University / Université Laurentienne

ACCESSIBILITY CLAUSE AND PERMISSION TO USE

Abstract

In this study, Histology images of the patients were used as input to the Convolutional Neural Network (CNN) models to predict the risk of the patients with brain tumour. Motivation for this study was to highlight the emerging role of deep learning in the field of precision medicine and suggest an expanding utility for computational analysis of histology in the future practice of pathology. The Region of Interest (ROI) of the histology images (1024x1024 pixels) of brain tumour of 769 patients was used to build a machine learning model which can predict the hazard ratio (is frequently interpreted as risk ratio) of the patients and survival time of the patient. Five diverse CNN models have been trained namely, DensNet121, VGG-19, Xception, Inception-V3 and Inception-ResNet-V2. The loss function of Cox-proportional hazard has been used to fit a survival model. For each CNN survival model, a 15-fold cross validation was implemented on the training data. The image data for 769 patients was split into 80% as training with 616 patients and 20% for testing consisting of 153 patients. There are 1239 ROI images for 616 patients and 266 ROI images of 153 patients. For each cross validation, the CNN model was trained for 100 epochs. For the test data (153 patients), 9 High Power Fields (HPFs) (256x256 pixels) were sampled from each ROI, and a risk is predicted for each field. The Median HPF risk is calculated in each ROI and the second highest value among all ROIs was selected as the patient risk. The predicted risk is calculated by taking the average of predicted risk for the last 5 epochs. The median concordance index (CI) and integrated brier score (IBS) is evaluated from the predicted risk and are compared for each of the models. Wilcoxon Ranksum test is used to test the null hypothesis that the CI and IBS of five CNN models was significantly different for 5% significance level. From the five CNN structures, DensNet121 had the best performance, which was followed by VGG19, InceptionResNet-V2, InceptionV3 and Xception, respectively.

Keywords:

Convolutional Neural Network, Survival analysis, Histology, Region of Interest (ROI), High Power Fields (HPFS), DensNet121, VGG-19, Xception, Inception-V3 and Inception-ResNet-V2.

Acknowledgments

First and foremost, I want to thank my professor, Dr. Kalpdrum Passi, for all his time, guidance, and resources that he has generously shared with me. He gave me valuable advice on the course of my research while I was still in the early stages of my master's degree. Throughout my studies, his great patience and consistent guidance have helped me conquer many challenges one by one. Without him, I would not have been able to complete either of these tasks.

Secondly, I want to say special thanks my wife, Dhruvyasha Patel and my parents for supporting me with guidance to solve the problems that I encountered and comforts me with great patience when I was having mental outbreak.

Finally, I want to give my appreciation to my friends. It is with their understanding, support as well as consistent encouragement both mentally and financially, I got the courage and determination to pursue my second master in Canada.

Table of Contents

Thesis Défense Committee ii

Abstract iii

Keywords: iv

Acknowledgments v

Table of Contents vi

Table of Figures ix

Chapter 1 1

Introduction 1

1.1. Research background and motivations 1

1.1.1. Survival Analysis 1

1.1.2. Standard methods in Survival Analysis 2

1.2. Deep Learning in Survival Analysis 3

1.2.1. VGG-19 5

1.2.2. Inception-ResNet-v2 5

1.2.3. Inception-v3 6

1.2.4. Xception 6

1.2.5. DenseNet-121 7

1.3. Model Evaluation Metrics 7

1.3.1. Concordance index 7

1.3.2. Integrated brier score (IBS) 8

1.4. Thesis objectives and Outlines 9

Chapter 2 10

Literature Review 10

Chapter 3 20

Dataset and Material 20

Chapter 4 28

Methods 28

4.1. Methodology 28

4.2. Cox Proportional Hazard Model 30

4.3. Integrated Brier Scores (IBS) 31

4.4. Loss function of Cox-proportional hazard 31

4.5. Tools used for coding 32

4.6. Convolutional Neural Network (CNN) Models used in this study 33

4.6.1. VGG (Visual Geometry Group) Networks 36

4.6.2. Dense Networks (DenseNets) 39

4.6.3. Inception Networks 42

4.6.4. Inception-ResNet Networks 46

4.6.5. Xception Networks 48

Chapter 5 50

Results and Discussion 50

5.1. Survival analysis using VGG-19 structure 50

5.2. Survival analysis using DensNet-121 structure 51

5.3. Survival analysis using Inception-V3 structure 52

5.4. Survival analysis using InceptionResNet-V2 structure 54

5.5. Survival analysis using Xception structure. 55

5.6. Comparing results of CNN structures 56

5.7. Prediction on Testing data using 5 CNN structures 60

Chapter 6 63

Conclusions and Future Work 63

6.1. Conclusion for DenseNet121 structure 63

6.2. Conclusion for VGG-19 structure 64

6.3. Conclusion for Inception-ResNet-V2 structure 65

6.4. Conclusion for Inception-V3 and Xception structures 65

6.5. Suggestions for future study 65

References 67

Table of Figures

Figure 3.1 Proportion of censored and uncensored patients in the dataset. 22

Figure 3. 2 Two ROI images from TCGA-02-0001 with 358-month survival 23

Figure 3.3 upper Left: TCGA-06-0201 with 12-month survival, upper right: TCGA-02-0016 with 2648-month survival. Lower left: TCGA-02-0033 with 86-month survival, lower right: TCGA-02-0014 with 2512-month survival. 24

Figure 3.4 upper left: TCGA-02-0006 before augmentation, upper right: TCGA-02-0006 after augmentation. middle left: TCGA-02-0009 before augmentation, upper right: TCGA-02-0009 after augmentation. lower left: TCGA-02-0033 before augmentation, lower right: TCGA-02-0 26

Figure 4.1 Convolution Neural Network 34

Figure 4.2 CNN one Example [45] 35

Figure 4.3 Classification Stage (Fully Connected Layers) [49] 36

Figure 4.4 VGG19 Architecture [50] 37

Figure 4.5 VGG19 Layers Configuration [36] 38

Figure 4.6 DenseNet Blocks [21] 39

Figure 4.7 DenseNet Architecture [21] 40

Figure 4.8 DenseNet Architecture Layering Details [21] 41

Figure 4.9 Inception Architecture [16] 43

Figure 4.10 Inception V1 Layers Details [16] 44

Figure 4.11 Inception V1 Architecture [16] 44

Figure 4.12 Inception V1 Layering Details [16] 45

Figure 4.13 Inception V3 Layers Details [51] 45

Figure 4.14 Inception V4 vs. Inception-ResNet layering details [19] 47

Figure 4.15 Inception-ResNet V2 Architecture [19] 48

Figure 4.16 Xception Modules [20] 49

Figure 4.17 Xception Architecture Details [20] 49

Figure 5.1 Boxplot presenting concordance index in 15 folds cross validation. 57

Figure 5.2 IBS of 5 CNN structures using 15-fold cross validation. 59

List of Tables

Table 3.1 Survival data for 14 uncensored TCGA patients 20

Table 3.2 Proportion of censored and uncensored patients and ROI slides in train and test data. 25

Table 4.1 Parameters for Adagrad optimizer 32

Table 5.1 Results of VGG19 for 15-fold cross validation 50

Table 5.2 Results of DensNet121 for 15-fold cross validation. 52

Table 5.3 Results of Inception-V3 for 15-fold cross validation. 53

Table 5.4 Results of Inception-ResNet-V2 for 15-fold cross validation. 54

Table 5.5 Results of Xception for 15-fold cross validation. 55

Table 5.6 Wilcoxon ranksum/signed rank test for comparing 5 CNN structures - in term of CI. 58

Table 5.7 Wilcoxon ranksum/signed rank test for comparing 5 CNN structures in term of IBS. 59

Table 5.8 Hazard Ratio of testing data using 5 CNN models. 61

Table 5.9 Concordance index for the whole 153 test patients 62

Chapter 1

Introduction

1.1. Research background and motivations

1.1.1. Survival Analysis

Clinical results particularly prognosis is frequently introduced as the timeframe between the beginning and end of the clinical perception in combination with a binary status data, indicating whether each patient experienced a clinical event of significance, such as a drug reaction or the recurrence of a specific disease. This data is usually incomplete due to a lack of observation of the event of interest. For example, a patient is being studied for recurrence of a certain disease, the disease does not re-appear during that time but occurs after the study ends. In that case it will be ambiguous whether the patient should be ordered into recurrence-positive or recurrence-negative category. Such observations are often treated as censored observations and observation time is integrated into the analysis. Such type of analysis is known as survival analysis.

Survival analysis refers to the statistical methods to evaluate how long it will take for a well-defined event of interest to occur. The time between inception of observation and occurrence of event is called survival time. It has a wide range of applications such as death of a specie, reappearance of a tumor cell, time until the failure of a part of machinery, lifetime of a machinery etc. [1].

The most important issue in survival analysis is censoring which is a missing data problem. It is usually faced when the study data is not available for all subjects under study. When the

time to event information is not available due to the end of the analysis or lack of follow up, the subject is said to be censored [2].

1.1.2. Standard methods in Survival Analysis

In the medical field, survival analysis plays an important role in the prediction of mortality, recurrence of a disease, development of a disease cell etc. The subject of event of interest in these cases are usually studied for years before its occurrence. The data gathered is then analyzed statistically to draw useful conclusions. Traditional methods of survival analysis include KM estimator, log rank test, Cox regression model and penalized cox models [3].

Lately, machine learning techniques have been developed and adapted in many fields which involve statistical analysis of huge amount of data. Machine learning algorithms have been proved to perform tasks very accurately. These algorithms “learn” from the data by extracting and recognizing the trend in data, which, otherwise would be hard for a human being to recognize. Generally, machine learning can be divided into supervised, unsupervised and reinforcement learning [3], [4], [5], [6], [7], [8].

Some of machine learning methods adopted for survival analysis are listed as follows:

- For the analysis of multivariate survival time data, survival trees were constructed. [9].
- Bagging survival trees or bootstrap survival trees [10].
- Cox Boosting, which was extended to statistical problems and found useful in survival analysis [11].
- Artificial Neural Networks (ANNs) were found to be useful in survival analysis and were first used to predict cancer survival data. [12].

1.2. Deep Learning in Survival Analysis

Traditional survival analysis methods involve scalar or categorical features. The advancement of image analysis techniques using deep learning technologies and computational capacities has made it feasible to accomplish exceptional outcomes and implement deep learning architectures on sizeable datasets with various underlying processes and more individual learning inside. Deep convolutional architectures have surpassed the performance of machine learning algorithms in many applications. Some of its applications in medical imaging includes radiology, pathology, detection of abnormalities in magnetic resonance imaging (MRI), computed tomography (CT) and ultrasound etc. Convolutional Neural Networks (CNN) can extract predictive features from the images, thus making the system completely end-to-end. This reduces the expert intervention as the feature extraction is being done by the model itself. CNNs learn the weights of kernels which are used for feature extraction through backpropagation.

Histology is defined as the study of microanatomy of tissues, cells and organs as seen through microscope. It has been a significant device in interpretation of cancer for over a century. It involves the study of structure of cell and relation between its functions and structures. Histopathology and anatomic pathology are branches of medicine that involves the study and diagnosis of abnormalities in tissues or cells. For testing the presence of cancer or tumor cells, sample of tissues from the body is removed and then analyzed under microscope. The interaction of individual with its environment results in phenotypic data. This data present in histology reflects the total impact of sub-atomic adjustments on malignant growth cell conduct which provides conducive observable read out for destructive behavior of disease. These reports assist the specialists in the prediction of stage of cancer, recurrence time of cancer cell and survival time of patients etc. This study is often referred to as survival analysis which has gained importance in pathology since it gives an estimate of current stage

of cell disease and helps the specialist in treatment choice. Human analysis of histology is profoundly abstract and is not repeatable consequently, computational examination of histology imaging has gained huge importance.

Genomic data refers to sequence of an individual's sequenced Deoxyribonucleic Acid (DNA). Genome is one's complete set of genes and genomic data can be derived from it by sequencing the DNA variations. Genomic reports can be used for following purposes [13]:

- It can be used to reveal a person's inheritance and information about someone related to a person. Such information has gained importance especially in crime departments.
- It can be used to predict a certain disease risk that can be caused by genetic error.
- It can be used for gene therapy, which is used to cure a certain disease by correcting a genetic error in advance.
- It can be used for developing applications of genomic technologies for genome editing, to add, cut or change genomic sequences.

Genomic profiles have been serving as a source for discovery of predictive, prognostic, and therapeutic biomarkers since past two decades [13]. The important steps in incorporating the genomic data with survival analysis involves the identification of features which correlates with survival data and assessment of survival association. The repositories of genomic data are expanding which makes it possible to introduce artificial intelligence methods in the study of such types of data. This type of data can be used to implement deep learning algorithms for the detection of features that correlates the survival analysis with genomic data and prediction of survival time [13]. Thus, both genotypic and phenotypic data will be used for survival analysis.

Convolutional Neural Networks (CNN) are SOTA (State of the Art) models for classification of images, segmentation, object detection, object recognition, object localization, as well as

feature extraction. During past two decades these models have been developed and improved to produce outcomes far better than traditional machine learning models. The very first CNN architecture, which became famous and inspired researchers was LeNet-5 proposed in 1998 [14]. It was a shallow convolutional network as compared to modern networks and had only 60,000 learnable parameters. With the development of technology and faster GPUs and CPUs, CNN models have improved. Several architectures namely AlexNet [15], GoogLeNet [16], VGG-19 [17], and ResNet [18] showed improvements in performance for many computer-vision tasks. These models are discussed briefly in the following sections.

1.2.1. VGG-19

VGG or Visual Geometry Group has the simplest architecture, but it can also outperform many complicated architectures. It consists of stack of convolution layers that are of size 3×3 and max pooling filters. The concept of VGG is that each layer should have more filters as compared to the previous layer. As the input propagates through layers, width and height of input decreases whereas depth of the input volume increases. VGG-19 has "19" because of number of layers it contains [17].

1.2.2. Inception-ResNet-v2

Inception-ResNet-v2 incorporates the concept of residual networks in inception networks. Inception networks are complex networks which solved the requirement of "deeper" networks by introducing "wider" networks [19]. Rather than having filters of one size in each layer, it stacks filters of multiple sizes in each layer. The output from each set of filters is concatenated and fed into the next layer. ResNets are residual networks that work by skipping connections. The output for previous layer is added to one or several layers ahead. ResNets solve the problem of vanishing gradient and allow the training of much deeper network effectively. Both are "State of the Art" architectures and have low computational cost.

Inception-ResNet-v2 is a hybrid network that includes a residual block after every inception layer. The parameters of layers are engineered so that the size of output from previous layer remains same as the layer ahead. Training of this type of architecture reduces the computational cost by a great amount.

1.2.3. Inception-v3

This network belongs to the Inception family and comes with several improvements such as 7x7 convolutions, batch normalization and Label smoothing [19]. This model has developed as the most widely used model as it showed improved performance in image recognition tasks. It was proposed by Szegedy, et. al. [19]. The model consists of building blocks which are defined by arranging convolutional layers, average pooling, max-pooling, dropout layers, concatenation layers as well as fully connected layers. The activation of final layer is SoftMax. Inception-v3 consists of both symmetric as well as asymmetric blocks.

In inception networks, there are too many filters in one layer which increases the output feature map of any layer. This results in greater computational cost. To solve this problem 1x1 convolution filter is used. This filter reduces the number of feature maps and hence the computational cost. This technique is known as dimensionality reduction. The filter is also known as projection layer as it projects the most variant features from the stack of feature maps.

1.2.4. Xception

Xception stands for “extreme inception”. It came with modified separable convolution and it outperforms Inception v3. It was developed by researchers of Google. The depth wise separable convolution has two levels of operations, nxn spatial convolution and 1x1 convolution to change the size. In Xception, this arrangement is reversed i.e., 1x1 convolution is performed first [20]. Every separable convolutional layer is followed by batch

normalization. By re-centering and re-scaling, the input layer, batch normalization improves the model's performance.

There are three types of flow as mentioned in [20], the entry flow, the middle flow, and the exit flow. The middle flow is suggested to be executed eight times before final layer. Xception has architecture with two main features: depth wise separable convolution layer and skip connection just like ResNet. The main advantage of this network is its achievements over several state-of-the-art architectures.

1.2.5. DenseNet-121

DenseNet-121 belongs to the Densely connected Convolution Networks. The most important concept in DenseNet is its densely connected block. Each layer is getting additional inputs from all the former layers and then passes on its output in the form of a feature map to the subsequent layers. It utilizes concatenation and each layer receives collective knowledge from preceding layers. The concept is like ResNets, the only difference is that rather than addition, this layer performs concatenation on the results from the previous layers. DenseNets exceptionally eliminates the problem of vanishing gradients as it ensures maximum information flow throughout the network [21]. Densenet-121 with 121 layers is easier to train and requires less memory.

The CNN architectures VGG-19, DenseNet-121, Inception v3, Inception-ResNet-v2 and Xception are used for survival analysis on genomic and histopathologic data.

1.3. Model Evaluation Metrics

The metrics used to test the performance of trained model are given next.

1.3.1. Concordance index

Concordance index is also known as c-index is a metric to evaluate models which are trained for analysis of survival. It also includes censored data while validating the model. It estimates

the extent to which the observation from two different distributions differs. Both the observations are chosen randomly. It can be written mathematically as:

$$C = P(a1 > a2)$$

The intuition behind c-index is that for example we have two patients under observation. The patient with more risk has less survival time or time to disease and the patient with less risk has more survival time. This intuition is extended to several patients and a statistic function is formed which can be applied to survival models.

The best model will have c-index 1 and c-index=0.5 indicates that the model is predicting randomly.

1.3.2. Integrated brier score (IBS)

Brier score measures the accuracy of a probabilistic function. At any given time, t, it calculates the accuracy of predicted survival function. Its value ranges between 0 and 1 where 0 being the best possible value and 1 is the lowest possible value. It is the average squared distance between calculated survival and predicted survival state. A suitable model will have brier score below 0.25. The choice of method of censoring for the data effects the definition of brier function.

The integrated brier function gives the brier score for a model at all given times i.e., t ranges from 0 to Tmax. The integrated brier function can be defined as:

$$IBS = \frac{1}{Tmax} \int_{0}^{Tmax} BS(t)dt$$

where BS is brier function, Tmax is maximum time and t is instantaneous time.

1.4. Thesis objectives and Outlines

Histology has been an important tool in cancer diagnosis for more than a century. Although prognostication relies on genomic biomarkers that measures genetic alterations, gene expression and modifications, histology remains an important tool in predicting the future course of a patient's disease. The phenotypic information present in histology shows the effect of molecular alterations on cancer cell behavior and provides a complete visual lookout on aggressiveness of diseases. The objectives of this study include:

- Understand the histology images of brain tumor and pre-process them
- Find ROI of the images and sample High Power Fields (HPF) and reduce image sizes to enable efficient processing
- Study survival analysis methods such a Cox proportional hazard and apply the technique to predict hazard ratio and survival times
- Apply deep learning models to predict the risk in patients and find the best deep learning model for the histology image dataset of brain tumor

The thesis is organized as follows:

Chapter 1 introduces the concepts of survival analysis and deep learning models.

Chapter 2 reviews the related literature on survival analysis techniques and deep learning models used to process image datasets.

Chapter 3 describes the dataset and pre-processing steps.

Chapter 4 discusses the survival analysis methods for predicting the risk of the patients.

Chapter 5 discusses the results and analyses the deep learning models.

Chapter 6 concludes and discusses the future work.

Chapter 2

Literature Review

Survival analysis was developed in 17th century when first life table was built by J. Graunt in 1662 as discussed in D.A Freedman [22]. The term was related to the inspection of death rate of human beings for a long time but during the last few decades, the applications of survival analysis has been expanding to several areas. The subject has now covered vast areas including finance, economics, breeding etc. Some of its applications involve estimation of time up till employee termination or quit, failure of part of machinery, time until first sale of the salesperson, time until recovery of a patient, time until company fails or bankrupts, etc.

Methods for survival analysis took an innovative turn when Kaplan and Meier proposed KM estimator [3] in 1958. Their work involved estimation of survival probabilities and hazard rates. An alternative non-parametric approach to calculate collective hazard rate was proposed by Wayne Nelson and Odd Aalen and was named Nelson-Aalen estimator. It estimates the mortality rate in case of non-censored data as well as censored data. Another noteworthy contribution was Cox proportional hazards model [5] which was proposed by Cox in 1972. It was a semi-parametric model which estimates a function that involves coefficients of each explanatory variable. It has two factors, one is baseline hazard that reports the variation of risk with time, the other is independent of time i.e., it depends on censoring patterns and evaluate the exponential function of predictors.

Cox model provides the risk ratio for each involved in survival analysis. The risk for each variable provided by the model is referred to as risk ratio (RR). If the risk ratio is below 1 then the model shows decreased risk. If it is more than 1 then it shows increased risk. Several

experiments have been suggested to verify the hazard model's assumptions. These methods were developed at the end on 20th century. One of them, which is easy to apply, is based on Schoenfeld residuals. Similarly, many methods were proposed for the testing of cox model among which the most widely used model is Breslow's method [4].

A few applications and research based on cox-model is discussed in papers [23], [24], [25]. P. Schober and T.R. Vetter [1] illustrated the cox-model with the application of breast cancer. S.H. Moolgavkar et el [26] discusses the relevance of cox-model for Epidemiologic Studies. Thoracic surgical research studies the feasibility of cox model and applies it for thoracic research [27].

With the dawn of machine learning, many possible areas of applications emerged. Although they were first evolved for data mining purposes, it expanded to many different fields. Many researchers applied traditional machine learning algorithms for survival analysis. Some of them can be named as survival trees, emerged from decision trees; support vector machines, which was found to be very useful for censored data; bagging or bootstrapping survival trees, which has the idea of random forests behind it; random survival forests and cox boosting [28]. Furthermore, machine learning algorithms have been used for survival analysis to match nonlinear and complicated interaction effects between predictors and obtain more accurate individual survival probability prediction. Since most physicians and medical researchers can quickly examine statistical programs for evaluating survival data, a review article is currently useful for learning statistical approaches employed in survival analysis. In a variety of practical applications, machine learning (ML) approaches are frequently utilized to simulate nonlinear and complicated gene-to-gene interactions and enhance predictability. ML approaches have been modified for survival analysis to efficiently handle censored data and

properly develop prediction models using high-dimensional data. S. Lee and H. Lim [29] used SVM and cox proportional method for survival analysis on genomic data of breast cancer patients.

One more important method that got significant attention was ensemble method. Ensemble method is based on multiple classifiers. Several classifiers are treated, and the output depends on the maximum voting. For example, several decisions trees would be trained and the class which gets the most votes will be predicted. Then artificial neural networks (ANN) emerged, although they were proposed very early, their importance was realized much later.

During 1990s researchers started exploring the advantages of survival analysis using neural networks. In 1994 M. De Laurentiis and P. Ravdin proposed to use artificial neural networks (ANN) for the first time [6]. The researchers outlined the situations which can be solved by using ANNs. They also proposed "single time point models" in [7]. It was suggested that t year survival can be predicted by fixing a single time point, t and can be used for multiple time predictions repeatedly. Some other research includes work of E. Biganzoli et al. in [8] that uses logistic activation along with feed forward neural network and entropy as error function.

M. Leblanc and J. Crowley [24] implemented ANNs and discussed their performance as compared to traditional statistical Cox model. C.L. Chi et. al [12] applied ANN on breast cancer dataset for survival analysis and study of recurrence time of breast cancer. The tool was called prognosis decision support tool as it aids doctors for decision. However, the model failed to answer a few questions which were highlighted by the authors.

J. Friedman et al. [11] summarizes the results for implementation of ANNs for survival prediction of humans suffering from brain tumors. The dataset consists of gene expressions of cells and classifiers were reported to outperform the grading reported in other studies. D.

Faraggi and R. Simon [30] handcrafted features of biomedical images of lung cancer for survival analysis using cox hazard model.

The development of medical imaging technologies provides variety of complex images. The features from these images were handcrafted by humans for survival analysis using traditional methods. These hand-crafted features have confined capabilities to represent these complex features in a medical image. Convolutional Neural Networks (CNN) can craft those features automatically by learning the weights of kernels. They have shown to be useful in the medical field for disease detection. CNNs were applied for the very first time for end-to-end training and survival analysis in paper [31]. The architecture proposed was called DeepConvSurv. It was trained on lung cancer dataset and its performance was in comparison with state-of-the-art models. It can be concluded from the stated results that the model outperformed those traditional models i.e., Cox with LASSO, Cox with SuperPC, RSF etc.

In [32], J. Yao et. al. used deep learning-based approach to conduct research on biomedical feature extraction for lung cancer survival prediction. Another research stated the advantages of automatically derived features from lung cancer pathology images and suggested that they can be used for the prognosis of patients suffering from lung cancer with far better accuracy as compared to traditional models [33].

A. Krizhevsky et al. [15] researched on application of DeepConvSurv on whole-slide histopathological images. The research was aimed to point out challenges and improve the quality of treatment of an individual. Medical images have millions of pixels, as compared to regular images so a method called WSISA was proposed for survival analysis to implement on high resolution genomic images for lung and brain cancer. The method involved generating patches from images, clustering those patches, selecting clusters, and training them on DeepConvSurv, and aggregating the selected clusters. The performance of proposed

method was compared with traditional methods, and it was concluded that the model outperforms the Cox based models.

Another deep neural network, cox-Nnet was proposed by T. Ching et al. [33] which was based on cox model. It is reformed for survival predictions on high throughput data from gene expression. The architecture proposed consisted of an input layer, a hidden layer which was fully connected layer with 143 nodes and an output layer which comprised of a single node. The output node is named as proportional hazards. To prevent over-fitting, researchers also experimented with regularization techniques e.g., drop-out, ridge etc. It was concluded that it can extract more detailed features from data on pathway level as well as gene level. A method of dimensionality reduction was also proposed in the study to down sample the large number of features in the data.

A model named RNN-SURV was proposed by E. Giunchiglia et. al. [34] which was a recurrent neural network (RNN) model for survival prediction. It was suggested to input each feature along with given time interval so that model can capture effect of each feature along with time. This type of input was created by using embedded layers and the output from these layers were passed to recurrent network and sigmoid non-linearity. The model was optimized using two loss functions: cross entropy and c-index. It was concluded that the c-index of the model shows 28% more accurate results improving the state-of-the-art models [34].

In addition to that, Nnet-survival, which is also a deep learning model for survival analysis was proposed by MF Gensheimer et al. [25]. The model was trained on a very large dataset so minibatch stochastic gradient descent was used. This enables rapid convergence of the model and less loss of information. The performance of model was evaluated on simulated as well as real data and its comparison with DeepSurv and Cox-Nnet was stated. Cox-Nnet and DeepSurv only outputs the prognosis index whereas authors used Breslow method to generate

survival curves for patients. The models were compared using brier score and c-index for three different intervals and it was observed that the results of all models were very close to each other.

K. Matsuo et al. [35] conducted a research to compare deep learning survival analysis and cox hazard model. It was suggested that cox model assumes linear relation whereas medical biomarkers exhibit nonlinear association. Thus, it was proved that methods based on deep learning are more accurate when it comes to medical images.

Models which are specifically designed for survival analysis are still developing and there is much room for improvement. The state-of-the-art deep learning models: VGG-Net, AlexNet, Xception, Inception, and DenseNet which are developed for classification of images can also be applied in medical research. These models leverage the researchers to use pre trained weights if the data on which the model is to be trained on is very small. The models have been trained on much larger data which contains almost 1000 classes and can perform accurately for classification as well as detection-based tasks. The development of advanced computers and GPUs allowed the research to improve these networks. As the networks started getting improved, their applications became wider, and areas of applications increased.

The present study aims to apply different state-of-the-art CNNs for survival analysis on histopathologic as well as genomic data. The models used in the present study are further discussed as related work.

VGG16 was proposed by Simonyan and Zisserman [36] in 2014. It showed significant improvements over LeNet-5 and AlexNet. AlexNet which was proposed in 2012, was the first one to show an accuracy (top 5) of 84% on ImageNet dataset. VGG-16 improved this accuracy and brought it up to 92.7%. It replaced the 11x11 and 5x5 sized filters of AlexNet

by 3x3 filters. The architecture of VGG-16 consists of the input to convolution layer of size (244,244). A stack of convolutional layers is followed by max-pooling layer. The stride is set to 1 and activation after each convolution layer is ReLU. The stride of max pooling was preferred to be 2 pixels with 2x2 window size. Some configurations also have 1x1 convolutional layer which is applied to reduce the depth of input. Each combination of convolution and pooling layer is followed by another combination of same layers. The last few layers of the network consist of three fully connected layers and a SoftMax layer. VGG-16 was trained on multiple GPUs and it took 2 to 3 weeks to train VGG-16 on a system with four NVIDIA GPUs. The drawback of this network at that time was its training time and its size.

J. Yao et al. [32] used pre-trained weights of VGG16 to train RGB images of Thrombotic Microangiopathy (TMA) spot and predict colorectal cancer based on tissue analysis. This problem is related to survival analysis where risk score was used to implement model for prognosis. VGG16 was used for feature extraction to further train on LSTM. The results showed that their model outperformed other machine learning models with the hazard ratio 2.3 and AUC 0.69.

Kaplan and Meier [3] also applied pre trained VGG 16 for feature extraction to conduct research on Prognostic Analysis of Histopathological Images. A strong correlation was found between features and relevant biological outcomes. Cox PH model was used and trained on the features extracted through these networks. It was reported that the C-index achieved was 0.789.

Another architecture was proposed in [36] which was based on VGG-Net. This architecture was used for lung cancer survival analysis with biomarker interpretation. It was used for cell feature learning on the images of cells and was trained on NVIDIA GPU. It detects the cell,

crop out 80x80 image for each detected cell, which is sampled as a training feature. The label for each image is decided according to the corresponding risk of patient. These generated features were then trained on different models and results were obtained.

The inception network was first proposed by Szegedy et al. [16] in 2014. The concept was then adopted by Google and the original architecture was called GoogleNet. In 2015, Szegedy et al. [16] proposed Inception v3 and Inception-ResNet-v2. Inception-v3, which belongs to inception family, uses building blocks including, convolution, pooling, fully connected layers as well as dropout layers. The model was said to be an improved version as it utilizes factored 7x7 convolution, and BN auxiliary. BN auxiliary is the approach in which fully connected layer of auxiliary network is also normalized along with the convolutions. That is why the model is said to be equal to inception-v2 plus BN. The top-5 accuracy of inception-v3 was 93.9%. Inception-ResNet-v2, a variation of Inception-v3, employs residual networks along with the inception network to reduce the computational cost. As compared to a state-of-the-art model, it was more accurate. The top-5 accuracy was reported to be 95.2%. The architecture defined in [19] suggested a residual block after every inception block. Residual networks prevent overfitting by skipping connections. Inception networks go wider and includes a stack of filters of different size in the same layer. Both networks have their own advantages which can be combined using this network.

DenseNet architecture was proposed by G. Huang et.al [21] in 2016. The concept of this network revolves around the fact that the network will produce better results if their connections between layers close to input and output are short. DenseNets are like ResNets in a way that ResNets add outputs of layers whereas DenseNets concatenate them. This allows maximum flow of information across the layers of network; hence the information is less prone to vanish which was a common problem named as vanishing gradient. Different blocks of DenseNet include, dense layers, transition layers, max pooling, and batch normalization.

Dense connections involve stacking output of previous layer over the output of current layer. It has a regularizing effect, that reduces overfitting. Advantages of DenseNet includes eliminating vanishing gradient, improved feature propagation, a smaller number of parameters, and feature reiteration. Top-5 accuracy of DenseNet-121 is reported to be 92.3% [21].

S. Otálora et al. [37] discuss histopathological images as image magnification regression using DenseNet. The dataset used to conduct research was breast cancer and it was evaluated on TCGA prostate patches. Almost 34,000 images were used for the training of CNN with magnifications included.

Xception was proposed by F. Chollet [20] in 2017. This network is based on depth wise separable convolutional layers. Depth wise separable convolutions were first developed by Laurent Sifre in 2014 to slightly increase the accuracy of AlexNet. The concept was then utilized in Xception with some modifications. Xception is easy to implement as compared to inception network. It consists of 14 modules which consists of 36 convolution layers which have residual connections except the first and the last layer. The top-5 accuracy of the network on ImageNet dataset is 94.5%. to check the performance of network, it was trained on Juxta foveal Telangiectasis (JFT) dataset, and it showed better results as compared to ImageNet dataset. JFT dataset is internal Google dataset which consists of 17000 different classes to train for classification. The performance was compared with VGG-16, ResNet-152 and Inception V3 and it showed improvement in accuracy. It has slightly less parameters than Inception V3.

Yang et al. [38] researched on feature extraction of tumor image. The dataset used for this research was tumor CT images. The research showed how CNNs perform better than classic

algorithms in local binary mode. Furthermore, advantages of CNNs in medical field were demonstrated.

R. Joshi and C. Reeves [39] tested and compared the performance of several deep neural architectures for breast cancer prediction. In this research, the data used consists of histology images of cells from breasts. The main steps were preprocessing the images, addressing the problem of small dataset, extracting features using pretrained architectures and training the features on a multilayer perceptron model. The architecture used for feature extraction were VGG-19, InceptionResNetV2, Xception and InceptionV3, DenseNet-121. These networks were used with a few modifications and their results were compared and summarized. It was concluded that DenseNet-121 exhibit the highest average classification accuracy among different architectures.

Many important problems in the clinical management of cancer involves time to event prediction including accurate survival analysis. During this research it was found that deep learning has not been widely applied to these problems. Limitations like incomplete follow up leads to lack of data and research material. The selection of ROIs within each image requires an expert guidance to be accurate. So, in this study different deep learning models were used for automatic selection of ROIs and for incorporating a higher proportion of each image in training and prediction. Different CNN structures were used that can incorporate the additional features to evaluate the value added to histology in these more complex structures. To the best of our knowledge, no recent work was found that has proposed deep learning models for predicting risk of patients on histology images for brain tumor.

Chapter 3

Dataset and Material

The dataset consists of slide images of tissue sections from formalin-fixed, paraffin-embedded specimens and clinical feedback for almost 769 gliomas from the TCGA. TCGA is "The Cancer Genome Atlas" which aims to develop the process to analyze, treat and inhibit malignant growth. It was formed to use genome analysis technology, such as large-scale genome sequencing, to speed up the process of researching and understanding the atomic substructure of cancer. The images are of WHO grade II and III gliomas and WHO grade IV glioblastomas. It contains astrocytoma's and oligodendrogliomas. The dataset also consists of histopathological reports. The images have width 1024 and height 1024 and 4 number of channels. A csv file is also used which contains 210 columns. Two columns with survival time and status are used as target while rest of the genetic features are used for training.

Survival dataset used in this study is TCGA[1] data which was analysed and passed quality control review by (Mobadersany et al, 2018) [40]. This dataset includes Histologic images of 769 unique patients. The images are of brain tumour study. The survival time for dead and alive patients are presented in csv format. Those which were dead are uncensored and those patients which were alive are considered as censored events. The status for uncensored events in this dataset is 0 and the status for censored events is presented as 1. The survival data for a sample of 14 TCGA patients is presented in Table 3.1.

Table 3.1 Survival data for 14 uncensored TCGA patients

Patient ID	Survival months
TCGA-02-0001	358
TCGA-02-0003	144
TCGA-02-0006	558
TCGA-02-0007	705
TCGA-02-0009	322
TCGA-02-0010	1077
TCGA-02-0011	630
TCGA-02-0014	2512
TCGA-02-0015	627
TCGA-02-0016	2648
TCGA-02-0021	2362
TCGA-02-0023	612
TCGA-02-0024	1615
TCGA-02-0025	1300

In this data the term “censored” shows uncensored events when censored = 0 and it shows censored events when censored = 1. From 769 unique patients in this study, 388 of them (50.45%) are uncensored (censored = 0) and 381 of them (49.54%) are censored (censored = 1). As a result, half of the patients in the dataset are censored cases for which we do not know the exact periods when they are at risk. The proportion of uncensored and censored events are shown in Figure 3.1.

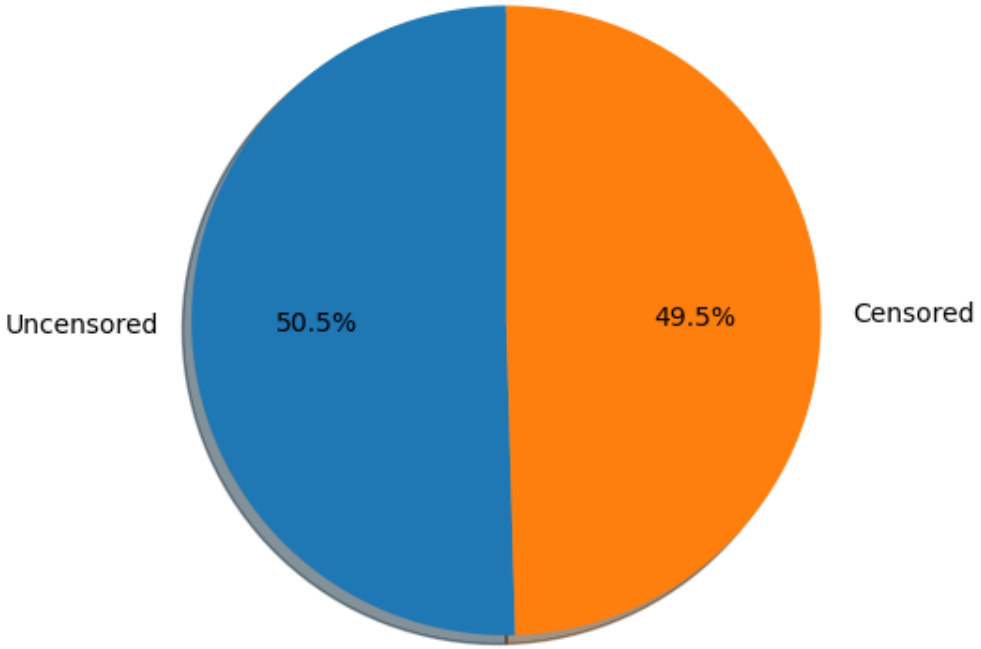


Figure 3.1 Proportion of censored and uncensored patients in the dataset.

The Histologic images of this dataset are presented in two separate folders. One folder includes histologic images for Training data and another one is for Testing data. The Training data folder includes 1239 images. It means that from each patient there is more than one histologic image. Mobadersany et al, 2018 [40] have done manual process to remove images which contain bubbles, those which include pen marks, poor staining, and section folds. They have manually specified the slides which include tumour by doing quality control review. They have used web-based quality control to manually select the Region of Interest (ROI) which include visible tumours for each patient. They have mentioned that a total of 1061 ROIs slide have been analysed in their study from the 769 unique patients. From each patient there could be more than one ROI slide in the dataset. For example, Figure 3.2. Shows two ROI slides for TCGA-02-0001 with survival of 358 months.

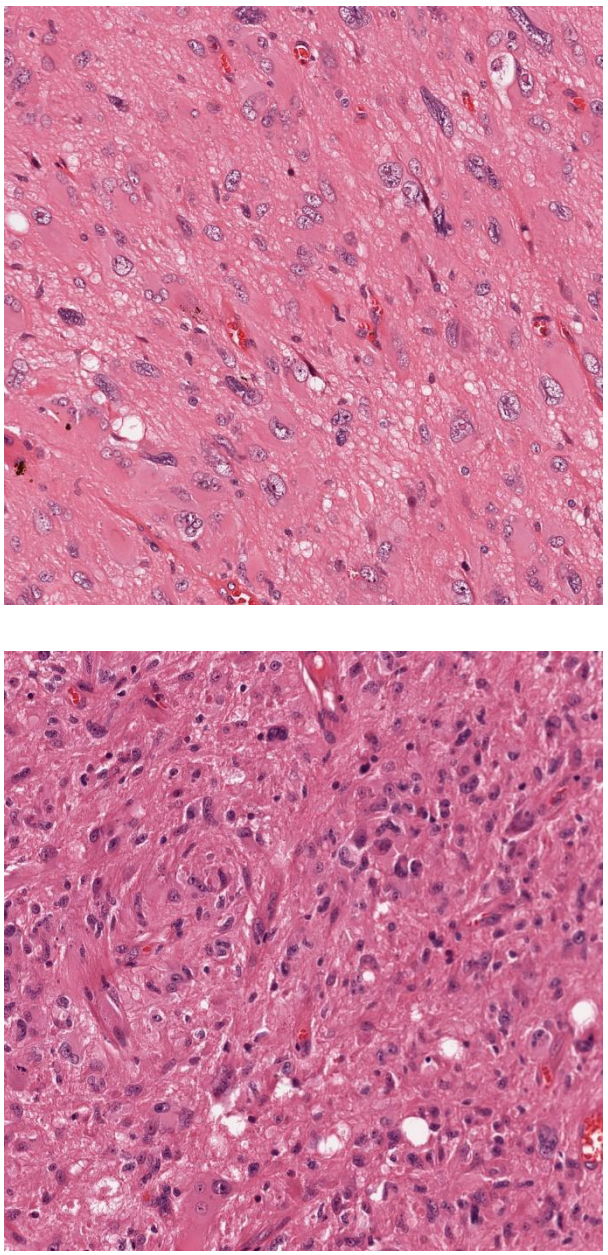


Figure 3. 2 Two ROI images from TCGA-02-0001 with 358-month survival

The ROI slides are RGBA images which include 4 channels. RGBA images are images which include four channels of Red, Green, Blue and Alpha. Alpha channel shows the opaqueness of the pixels in the ROI images. Each of the ROI images have 1024x1024 pixels. Figure 3.3 shows the ROI slides for two other patients with different survival times.

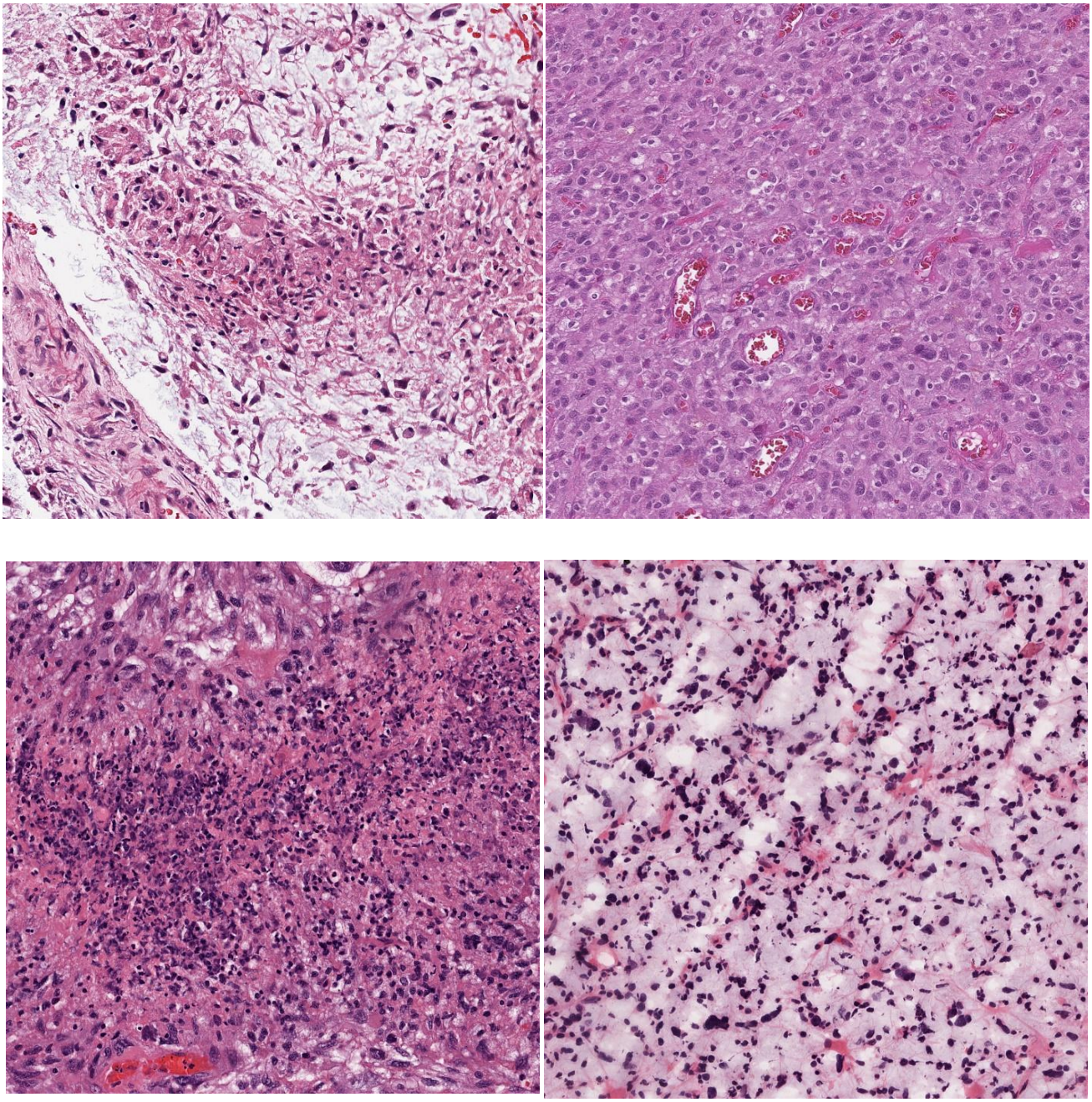


Figure 3.3 upper Left: TCGA-06-0201 with 12-month survival, upper right: TCGA-02-0016 with 2648-month survival. Lower left: TCGA-02-0033 with 86-month survival, lower right: TCGA-02-0014 with 2512-month survival.

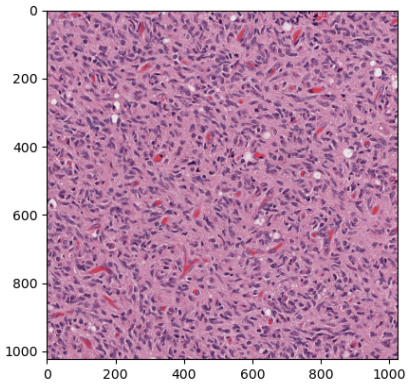
The images on the left show the ROI slides for patients with less survival months and the images on the right show the ROI slides for patients with more survival time. In the training folder we have 1239 ROI images for 616 unique patients. From the 616 unique patients in the training data, 314 of them (50.97%) are censored and 302 of them (49.03%) are uncensored.

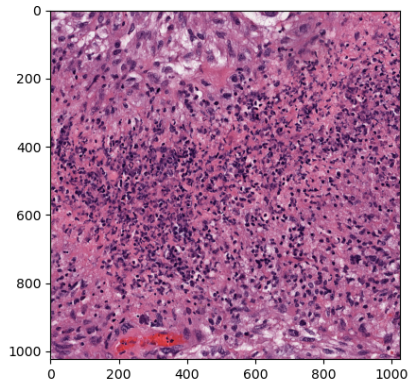
It means a bit more than half of the patients in the training data are censored observations. From 1239 ROI images, 554 of them (44.7%) are uncensored and 685 of them (55.3%) are censored. In the Testing folder, there are images for the remaining 153 unique patients out of a total 769 unique patients, 86 of them (56.2%) are uncensored and 67 (43.8%) are censored. There are 266 ROI images in this folder where 113 (42.4%) of them are censored and 153 (57.5%) are uncensored. Table 3.2 shows the proportion of status events for training and testing data.

Table 3.2 Proportion of censored and uncensored patients and ROI slides in train and test data.

Dataset	Unique patient	censored	uncensored
Training	616	314 (50.97%)	302 (49.03%)
Testing	153	67 (43.8%)	86 (56.2%)
Dataset	ROIs images	censored	uncensored
Training	1239	685 (55.3%)	554 (44.7%)
Testing	266	113 (42.4%)	153 (57.5%)

The size of the ROI images is big (1024x1024 pixels) for the analysis which requires a huge amount of memory. The ROI images were pre-processed and then cropped to 256x256 pixels for analysis. We used “open slide” images with a 20x objective magnification and then cropped the images from 1024x1024 pixels to 256x256 pixels after colour normalisation. Data pre-processing and augmentation was applied to the cropped images. Then on each cropped image a random contrast with interval range of [0.2, 1.8], and a random brightness with interval range of [-0.63, 0.63] have been implemented. Another augmentation that was done in this study is random horizontal and vertical flip. In Figure 3.4. the histologic ROIs are presented for three patients before and after pre-processing.





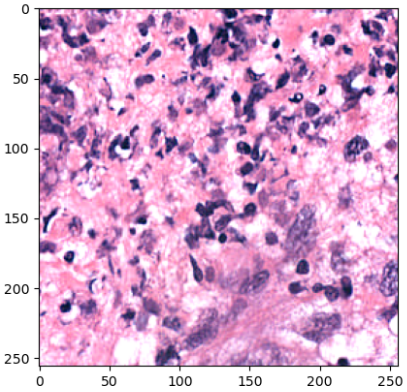


Figure 3.4 upper left: TCGA-02-0006 before augmentation, upper right: TCGA-02-0006 after augmentation. middle left: TCGA-02-0009 before augmentation, upper right: TCGA-02-0009 after augmentation. lower left: TCGA-02-0033 before augmentation, lower right: TCGA-02-0

After augmentation, the images are ready for survival analysis. Convolutional Neural Network (CNN) structures with the cox proportional hazard loss function was used to create a model which can predict the proportional hazard for each patient. For implementation of survival analysis with CNN, python programming language was used. The libraries used from python include scikit-image, scikit-survival, scikit-learn, NumPy and pandas. Also, CNN structure library of TensorFlow keras (TensorFlow version 2.3.0) has been used.

Chapter 4

Methods

4.1. Methodology

For each of the 5 CNN structures that have been used in this study (VGG-19, DenseNet121, Inception-V3, InceptionResNet-V2 and Xception), we have used 1239 histology images of 616 unique patients as training data and 266 histology images from 153 unique patients as testing data. The training data itself has been divided into 80:20 splits for training and validation, respectively. 15-fold cross validation was used. In each cross validation the training data (1239 images) was randomly divided into two parts of 80% for training and 20% for validation. Hence, each training split includes 991 images for training and 248 images for validation. The images are 1024x1024 pixels with 4 channels. The augmentation process that has been employed in this study, crops the original ROIs images to 256x256 pixels. Random contrast in range of [0.2, 1.8], Random brightness in range of [-0.63, 0.63], random right-left flip and random up and down flip. For implementing augmentation, the following functions from TensorFlow have been used: “image.random_crop”, “image.random_contrast”, “image_random_brightness”, “image.random_flip_left_right”, “image.flip_up_down”. The memory for loading all the images together, even after cropping the images to the size of 256x256 was not adequate. Hence, it was not possible to load all the images to the GPUs at the same time. Since the memory of the GPUs is 12 GB, batch size of 32 was used for loading the images into CNN models. Every CNN model was evaluated using 15-fold cross validation. To minimize the loss function, the Adagrad optimizer was used. The parameters used for Adagrad optimizer are depicted in Table 4.1.

Table 4.1 Parameters for Adagrad optimizer

Parameter	Value
initial accumulator	0.1
initial learning rate	0.001
exponential learning rate decay factor	0.1

The models were trained for 100 epochs. Choosing number of epochs is not significant, rather more important is the validation and training error. If it keeps dropping training should continue. If the validation error starts increasing that might be an indication of overfitting. The number of epochs were set as high as possible, and training was terminated based on the error rates. The methodology is given below as Algorithm 1.

Algorithm 1

- Train the model at 100 epochs.
- Calculate the predicted risk for the validation from the average predicted value and define it as a model average risk factor.
- Record the descriptive statistics such as Mean loss, Concordance Index, IBS
- Parameters such as image region from patient m are sampled as (HPFs)
- The Risk means risk for patient at HPFs and region. Then which is the median of 9 sample HPFs in region is calculated.
- The median risks for patient are then sorted in descending order. Then the second highest risk is considered as the predicted risk value for patient , i.e., . This methodology is implemented for each of the 5 CNN structures and the model average risk for validation data and the-second-highest risk for test data were considered as the predicted risk.

4.2. Cox Proportional Hazard Model

The cox regression model has strong applications in medical fields where the survival rates of the patients are observed at the prognosis stage. Over the time, the model has been intervened with the regression analysis to investigate its applications in the machine learning and deep learning fields.

Cox Proportional Hazard Model evaluates the effect of multiple factors on the survival of a variable and helps in assigning the weights while training the model. The prediction variables are termed as covariates in the Cox Proportional Hazard Model and the factors affecting the model are termed as the hazards. The hazard rate is an exponential function and contains the baseline hazard and the regression terms as given in equation (1).

$$\lambda(t) = \lambda_0(t)e^{(\beta_1 x_1 + \beta_2 x_2 + \ldots\ldots\_ \beta_k x_k)} \tag{1}$$

Here β_k are the regression parameters, t represents the survival time, $\lambda_0(t)$ is the hazard function determined by a set of e covariates, the term λ_0 is called baseline hazard and the 't' in $\lambda(t)$ reminds us that the hazard may vary over time. The model is sometimes referred to as a semi parametric because the requirement on $\lambda_0(t)$ does not need to be specified every time. The model requires the concordance index to be calculated for the survival prediction as given in equation (2) [41]. The index value of 1 represents the best prediction, 0.5 represents the random prediction and 0 represents poor prediction model.

$$CI = \frac{\sum_{i \neq j} 1\{\eta_i < \eta_j\} 1\{T_i > T_j\} d_j}{\sum_{i \neq j} 1\{T_i > T_j\} d_j} \tag{2}$$

Where, n_i and n_j represents the risk score of the *ij-th* unit and *T* represents the Time to event. Multiplication by the factor d_j discards pairs of observations that are not comparable because the smaller survival time is censored, that is $d_j = 0$.

4.3. Integrated Brier Scores (IBS)

IBS is an overall measure for the prediction model of risk of patients at all times. Integrated Brier Scores (IBS) is basically derived from the time dependent brier scores. The value of the brier scores ranges between 0 and 1 [42]. To obtain clearer picture of the model performance and the brier scores, we make use of integrated brier scores which predicts how good or bad is the prediction model at any given time. The range of the integrated brier score is from 0 to $\max(t_i)$. The brier scores and integrated brier scores can be computed using the following formulas as in equation (3) and (4).

$$BS(t) = \frac{1}{n}\sum_{i=1}^{n}\left[\frac{S(t \mid X_i)^2 1(t_i \le t \wedge \delta_i = 1)}{G(t_i)} + \frac{(1 - St \mid X_i)^2 1(t_i > t)}{G(t)}\right]$$

(3)

$$IBS = \frac{1}{\max(t_i)}\int_0^{\max(t_i)} BS(t)dt$$

(4)

Equation (3) finds the brier scores in the presence of censored data by adjusting the score weighs. Here $G(.)$ denotes the Kaplan-Meier estimation of the censoring distribution and $S(.\,|x_i)$ stands to estimate survival for the patient t. Note that the $BS(t)$ is dependent on the time t.

4.4. Loss function of Cox-proportional hazard

Cox-proportional hazard loss function has been used in the optimizer. In cox-proportional hazard the maximum partial log-likelihood should be calculated. Since the loss function in the optimizer tries to find minimum value, negative partial log-likelihood of the cox-proportional hazard model is used as the loss function of the CNN model. The negative Partial log-likelihood of Cox-proportional hazard or loss function is as below:

$$l(\beta) = -\sum_{i=1}^{n} \delta_i [\left(\beta^{'} X_i\right) - log\{\sum_{l \in R\left(T_i\right)} e^{\beta^{'} \chi_i}\}] \quad (5)$$

Or,

$$l(\beta) = \sum_{i=1}^{n} \delta_i [log\{\sum_{l \in R\left(T_i\right)} e^{\beta^{'} \chi_i}\} - \left(\beta^{'} \chi_i\right)] \quad (5)$$

where δ_i is the status of i^{th} image (censored (0) or uncensored (1)), $\beta^{'}\chi_i$ is derived from the last layer of the CNN structure and R (T_i) are the set of all patients at risk in time T_i. Patients at risk at time T_i are all the patients which are alive up to time T_i. hence for all the patients that are alive up to time T_i the log value summation for exponential predicted value is considered in the loss function. So, loss function calculates the log of summation of exponential predicted value for all patients at risk in time T_i minus the predicted value for the patient i. Then it sums for all patients i (i = 1 to n) if the patient i is not censored. If the i^{th} patient is censored the loss function for i^{th} patient will be zero. Noting the loss function of cox-proportional hazard model it could be seen that censored observations are only used in the patients which are at risk at each time T_i which include at-least one uncensored failure. The loss function also could be written as below:

$$l(\beta) = \sum_{i=1}^{n} \delta_i [log\{ \sum_{l \in R(T_i)} e^{pred_l} \} - (pred_i)] \quad (6)$$

where $pred_i$ is the predicted value or value returned from last layer of CNN model for patient I and e^{pred_l} is for all the patients l which are at risk at time T_i.

4.5. Tools used for coding

Survival analysis with convolutional neural networks was implemented using Python 3.8. TensorFlow 2.3.0 has been used to code CNN structures. The activation function of the last layer was considered as linear. The loss function to be minimized was considered as negative partial log likelihood of the cox proportional hazard model. CI and IBS have been used for evaluation of the models. Other python libraries which were used in the coding of this study are: “scikit-image”, “scikit-learn”, “scikit-survival”, “NumPy”, “pandas”, “matplotlib” and “re”. Since the computation would take too long on a standard CPU, SHARCNET platform for GPUs was utilized for efficient implementation.

4.6. Convolutional Neural Network (CNN) Models used in this study

The emergence of artificial intelligence has raised the bar for what machines can do and process [39]. Over the decade, Convolutional Neural Networks (CNN) has performed extraordinarily in the analysis of many different machine-based recognitions and classifications. The fields of deep learning and computer vision have intervened to produce amazingly précised and robust results which are far from the human imagination. The deep structures of CNN outperform human eye when it comes to analysing the images [43]. They are usually employed when we must deal with images having repetitive structures and characteristics, such as, textures, shape etc.

Generally, Convolutional Neural Networks consists of the following layers:

- Convolution Layer, the layer responsible for feature extraction.
- Pooling layer is in charge of reducing the number of activation maps
- The fully connected layer, which is generally found at the end of a convolution network, is used to identify the convolution or pooling layer's performance.
- SoftMax function, converts the output of fully connected layer to probability distribution and returns the probabilities as outputs.

In CNNs, the algorithms taken an input image with little pre-processing, assign weights to objects in the image depending on various aspects and differentiate them from the other objects. There is a filter just like in normal convolution called the kernel which is traversed over the whole image. While traversing, the purpose is to find the correlation between the original image section and the kernel components. Whenever the image patch and the kernel match, we get a very high correlation factor [44]. Based on this matching and mismatching, a kernel map is created which tells about the match intensity between the kernel and the image patch. This whole process results in a convolution when we shift the filter over the image. In case there are multiple kernels, multiple feature maps are created with respect to each one. Figure 1 shows the structure of a convolutional neural network. Layer 1 of the convolution (Conv_1) is passed through the filter of ones to read the image in its entirety. Each channel has specified dimensions depending on the image and the application requirements. At the end of each convolution process at each layer, we add a max pooling layer which is required to calculate the maxima after each convolution process over the image.

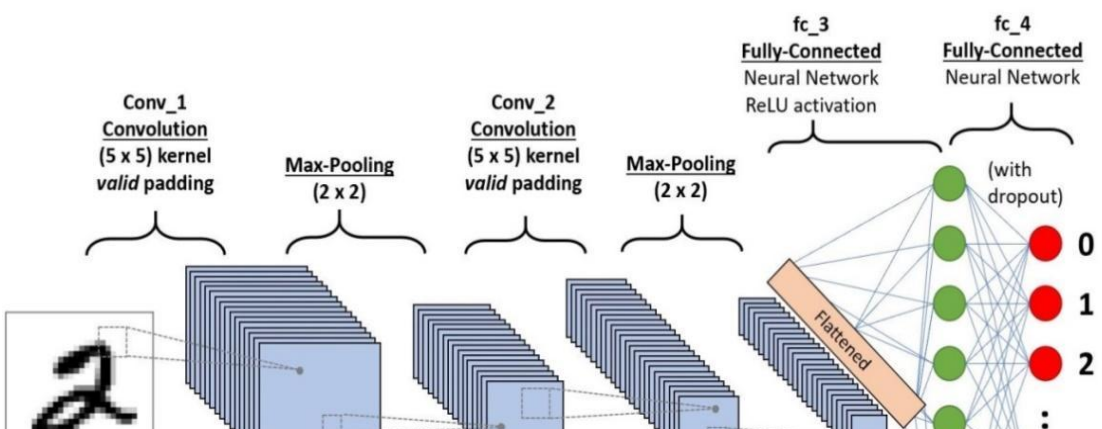


Figure 4.1 Convolution Neural Network

Due to convolution operations, gradually the image size is reduced depending on the kernel size. Figure 4.2 shows what happens while processing 1 layer of a CNN. The classification of the image is performed at the final layer where fully connected neural network layers are present. Before the neural network classification step, one flattening layer is added to robustly identify the parameters in a single dimension. This helps to feed the image to a multi-layer perceptron to process it and assign the relevant weights and biases. The classification through multi-layer perceptron is shown in Figure 4.3.

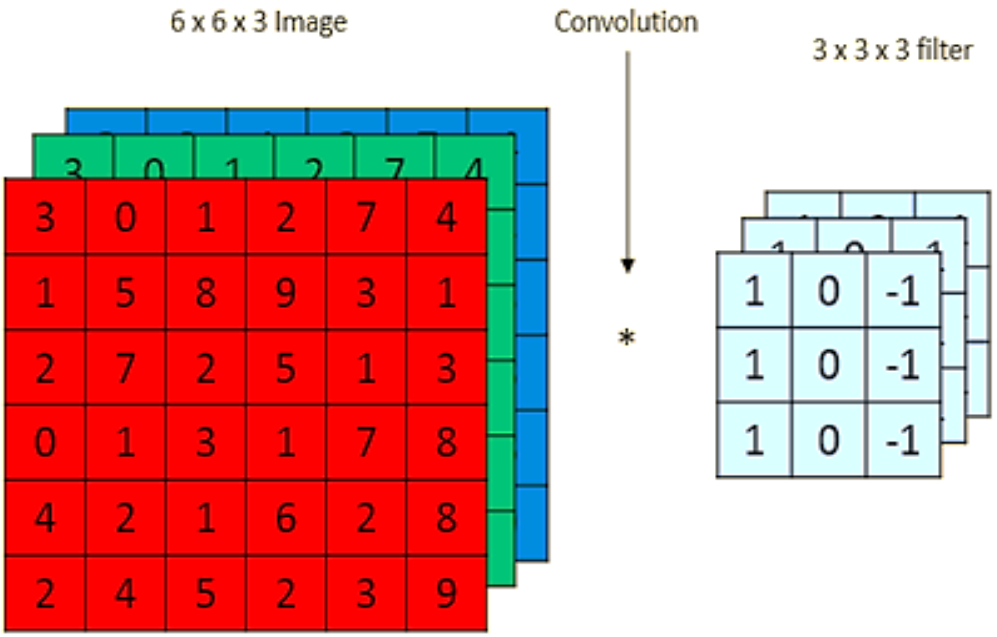


Figure 4.2 CNN one Example [45]

CNNs are more robust due to hierarchal feature maps between layers [45]. The sharing of structures between multiple filters results in effective learning of the parameters. The knowledge of one motif can improve the learning of other motif through shared sub structures. As a result, by using relevant kernels, CNNs can successfully capture the spatial and temporal dependencies in an image.

Through the years, as CNN has improved its learning and prediction capabilities, it is highly being employed for the real imagery data. Real life image data sets such as those available on ImageNet are utilized and CNN models are applied on them to classify more than 1000 classes of images with 1000 images per class. Besides images, CNNs are playing a pivotal role in games like playing GO, Deep mind, digit recognitions, text analysis, sound recognitions and natural language processing. Another major role of CNNs is found in medical imaging applications where there are datasets of radiology, ophthalmology, and dermatology [46], [47], [48].

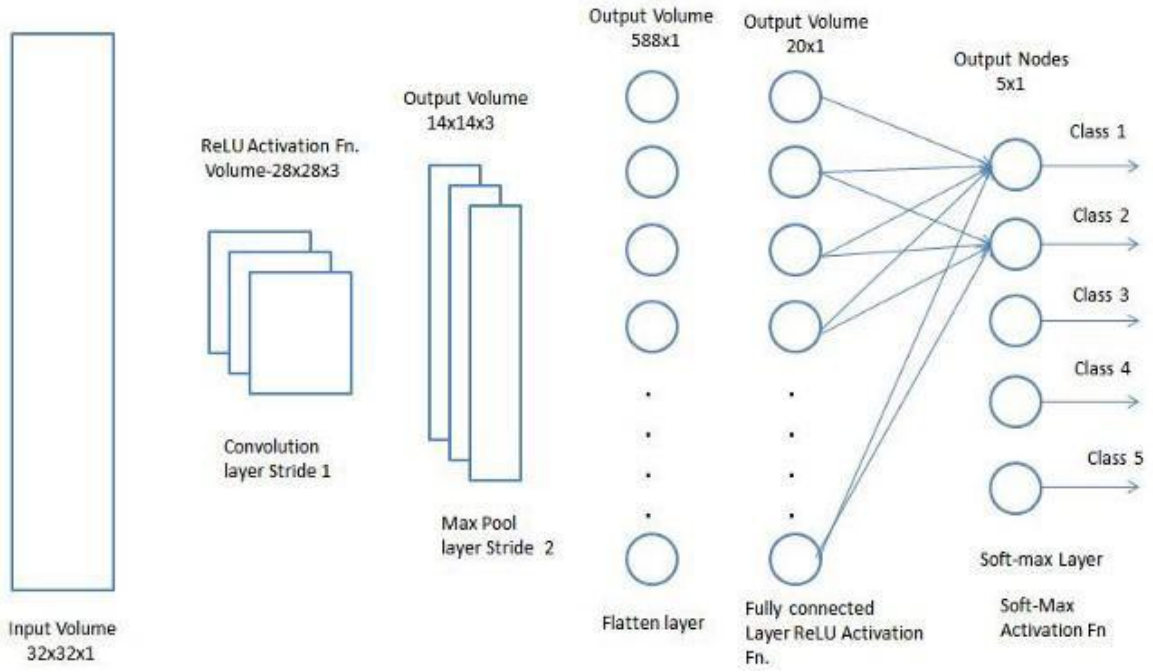


Figure 4.3 Classification Stage (Fully Connected Layers) [49]

Over the time, various techniques, and architectures for the CNNs have been developed and algorithms are built to extract more useful information using the techniques. Out of these algorithms VGG Nets (Visual Geometry Group) [36] are among the highly common architectures which are very helpful when it comes to deal with high dimensional image recognitions and classifications.

4.6.1. VGG (Visual Geometry Group) Networks

VGG came out as a successor of AlexNet [15] architecture and improved over it for classifying the images. It was developed by Visual Geometry Group at Oxford and hence named after it. The architecture of VGG19 along with the layer's information is shown in Figure 4.4. The algorithm takes 256x256 RGB images as input. As a pre-processing, only mean RBG subtraction is done over the whole training set.

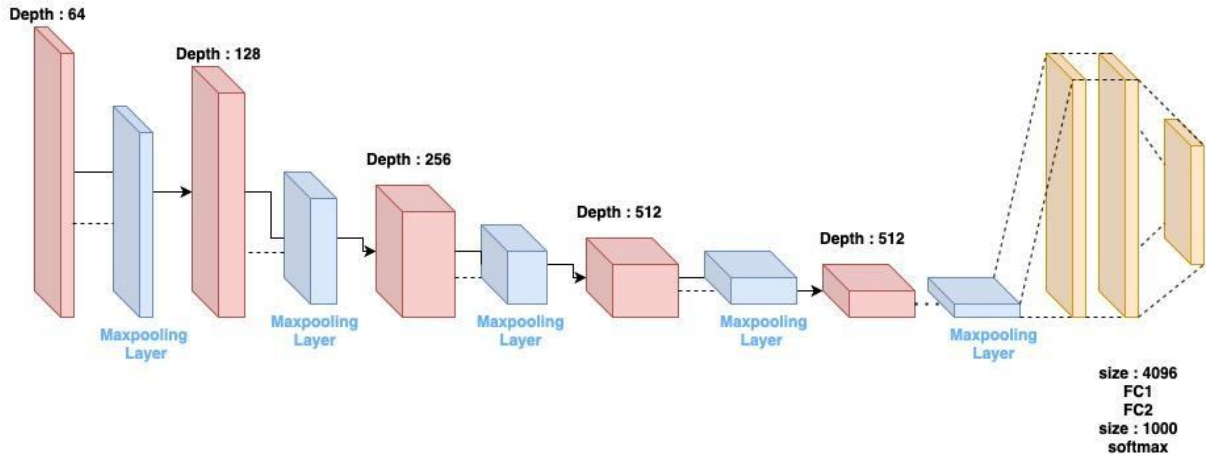


Figure 4.4 VGG19 Architecture [50]

A kernel of size 3x3 with a stride of 1 is used. To preserve the spatial resolution of the image, spatial padding is utilized. At the max pooling layer, a 2x2 layer is designed and convolution is performed with a stride of 2. To make the classifier more robust and improve the computational time of the classifier, ReLU is deployed for non-linearity. At the classification stage, three fully connected layers are added. The first two layers are of size 4096 and the last layer has a size of 1000 channels. The final layer is constructed through the SoftMax

function. The complete details of each layer of VGG19 are shown in Figure 4.5 with all the details of the weights and size of each layer in Column E. Rest of the columns contain information regarding the other VGG Architecture configurations such as VGG11 and VGG16.

The layer of the filter is denoted as conv<size of the kernel>-<total number of kernels in that layer>. Thus, conv3-64 means total of 64 3x3 square kernel filters. It is also worthy to note here that all the convolutional layers in VGG19 makes use of 3x3 kernel filters and that their number increases as a power of two e.g., 64, 128, 256, 512. In all the convolutional layers, a stride length of 1 pixel is used with a padding of 1 pixel on each side. There are 5 sets of convolutional layers present in the architecture. Two out of these 5 sets have 64 kernel filters present in them, next set has 2 convolutional layers with 128 kernel filters, next set has 4 convolutional layers with 256 kernel filters, and next 2 sets have 4 convolutional layers each, with 512 kernel filters. There are max pooling layers in between each set of convolutional layers. Max pooling layers have 2x2 filters with stride of 2 pixels. The output of last pooling layer is flattened in one dimension and fed to a fully connected network of multi-layer perceptron with 4096 neurons present in it. The output from the first fully connected (FC) layer goes to the second fully connected layer containing 4096 neurons, whose output is fed into the third and final fully connected (FC) layer with 1000 neurons. All these layers are ReLU activated for embedded nonlinearity. A SoftMax layer is present at the end which makes use of the cross-entropy loss function to provide the overall loss of the architecture.

Convolution Network Configuration					
A	A-LRN	B	C	D	E
11 Weight Layers	11 Weight Layers	13 Weight Layers	16 Weight Layers	16 Weight Layers	19 Weight Layers
Input (256*256 RGB image)					
Conv3-64	Conv3-64 LRN	Conv3-64 Conv3-64	Conv3-64 Conv3-64	Conv3-64 Conv3-64	Conv3-64 Conv3-64
Max-pooling Layer					
Conv3-128	Conv3-128	Conv3-128 Conv3-128	Conv3-128 Conv3-128	Conv3-128 Conv3-128	Conv3-128 Conv3-128
Max-pooling Layer					
Conv3-256 Conv3-256	Conv3-256 Conv3-256	Conv3-256 Conv3-256	Conv3-256 Conv3-256 Conv1-256	Conv3-256 Conv3-256 Conv3-256	Conv3-256 Conv3-256 Conv3-256 Conv3-256
Max-pooling Layer					
Conv3-512 Conv3-512	Conv3-512 Conv3-512	Conv3-512 Conv3-512	Conv3-512 Conv3-512 Conv1-512	Conv3-512 Conv3-512 Conv3-512	Conv3-512 Conv3-512 Conv3-512 Conv3-512
Max-pooling Layer					
Conv3-512 Conv3-512	Conv3-512 Conv3-512	Conv3-512 Conv3-512	Conv3-512 Conv3-512 Conv1-512	Conv3-512 Conv3-512 Conv3-512	Conv3-512 Conv3-512 Conv3-512 Conv3-512
Max-pooling Layer					
FC1-4096					
FC2-4096					
Last layer size-1000					
SoftMax layer					

Figure 4.5 VGG19 Layers Configuration [36]

The convolution layers and the fully connected layers are the only layers with trainable weights. Maxpool layer is used to reduce the size of the input image where SoftMax is used to make the final decision. The model provides great precision by implementing more layers. It reduces the computation time by making use of 3x3 conv filters rather than 7x7 filters. This helps to reduce the number of parameters and thus computational time is reduced. VGG19 outperforms VGG16 and other classification architectures. The model is also available publicly and people can explore the model and build applications upon it. Besides the insightfulness of the architecture, it also aids the recognition processes such as facial recognition, gender classifications and many other recognition tasks.

4.6.2. Dense Networks (DenseNets)

Moving ahead, a densely connected architecture was presented by Huang called the DenseNet which further digs and exploits the usage of the shortcuts and helps to simplify the architecture to avoid vanishing gradients stage [21]. It goes even deeper and connects all the layers with each other so that no information gets missed in the way, thus making the maximum feature usage instead of just focusing on the increasing layers and depth. The applications include segmentation, object detection and recognition.

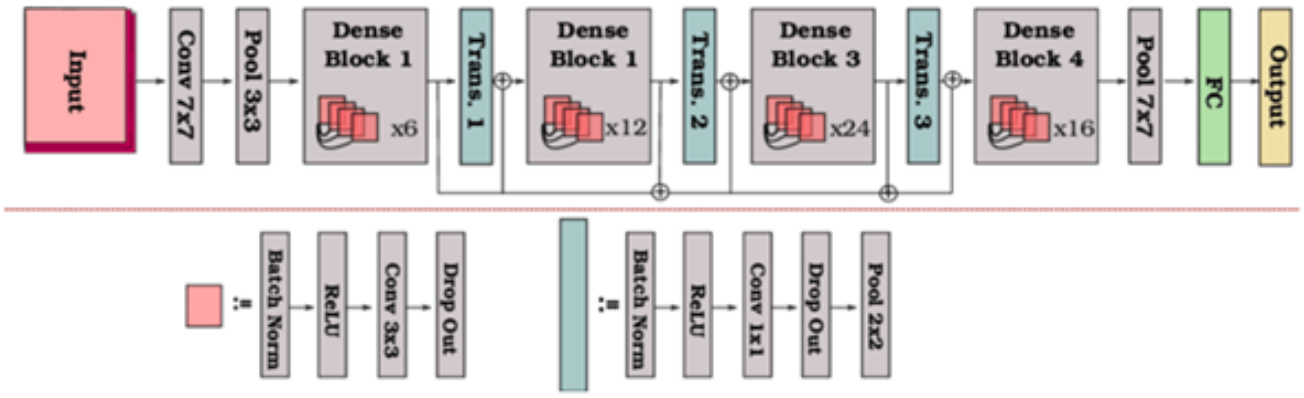


Figure 4.6 DenseNet Blocks [21]

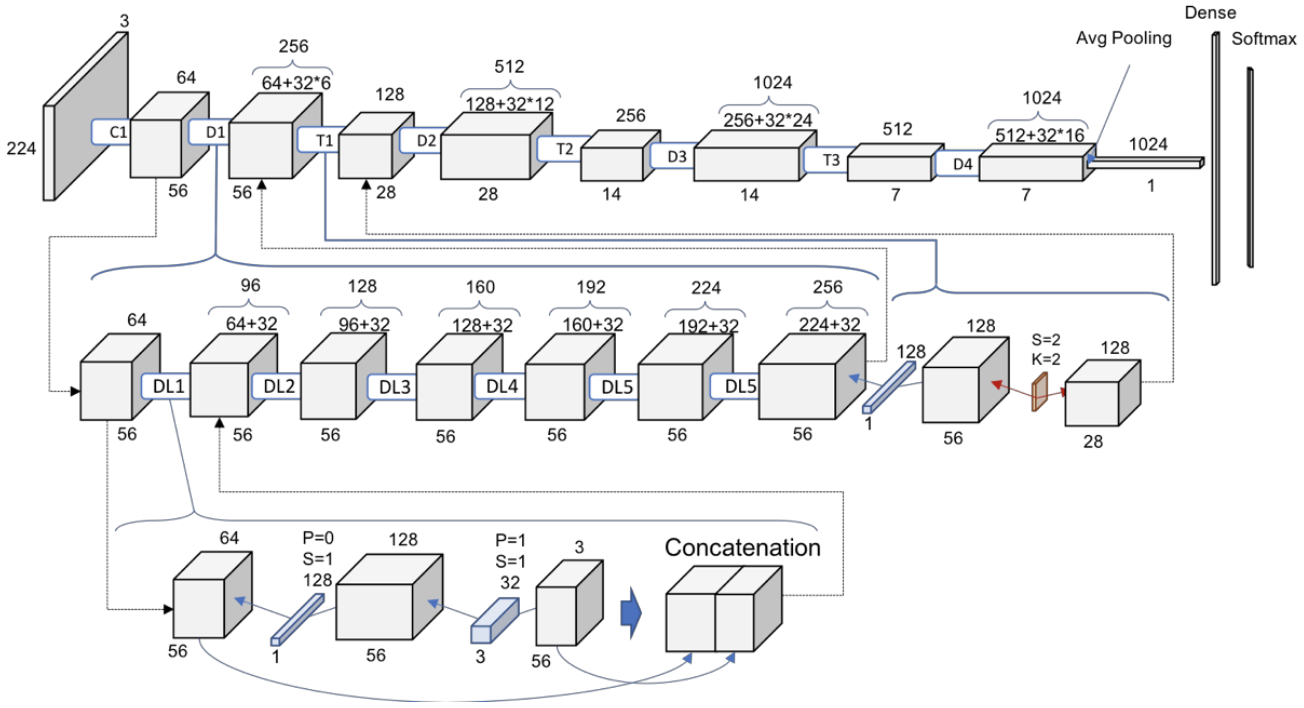


Figure 4.7 DenseNet Architecture [21]

Each layer has the feature map of the previous layers, and it has a role in devising the output of each subsequent layer. As a result, the overall architecture parameters are reduced with narrow layers and a few feature maps with direct access to the gradients from the loss function for each layer. Thus, DenseNet solved the many issues present with the ResNet model. The architecture is shown in Figure 4.6 with an elaborative representation in Figure 4.7.

The DenseNet architecture differs from the ResNet architecture in a way that the input to each layer in DenseNet is concatenated rather than being summed up. This changes the overall behaviour of the two networks and provides improved performance of the DenseNet with more efficient behaviour. The network shows that for N number of layers there are N(N+1)/2 direct connections. Therefore, the network has the capability to strengthen the feature upholding and propagation to the further layer.

To ensure the down sampling along with the concatenation operation in DenseNet, the architecture makes use of multiple densely connected blocks as shown in Figure 4.6 and 4.7. Within each block the size of the feature map remains alike and down samples while proceeding from one block to the other. To make it happen, the convolution and pooling operations are done outside the dense blocks. The detailed description of the DenseNet architecture layers, variants and arrangements is shown in Figure 4.8. This architecture arrangement is used by the ImageNet Datasets to train their learning models effectively. Our model incorporates DenseNet-121 which takes input image of size 256x256 and performs 7x7 convolutions.

Layers	Output size	DenseNet -121
Convolution	256*256	7*7 conv, stride =2
Pooling	56*56	3*3 max-pool, stride =2
Dense Block (1)	56*56	$\begin{pmatrix} 1\ X\ 1\ conv \\ 3\ X\ 3\ conv \end{pmatrix} X\ 6.$
Transition Layer (1)	56*56 28*28	1*1 Convolution 2*2 Average pooling, stride 2
Dense Block (2)	28*28	$\begin{pmatrix} 1\ X\ 1\ conv \\ 3\ X\ 3\ conv \end{pmatrix} X\ 12$
Transition Layer (2)	28*28 14*14	1*1 Convolution 2*2 Average pooling, stride 2
Dense Block (3)	14*14	$\begin{pmatrix} 1\ X\ 1\ conv \\ 3\ X\ 3\ conv \end{pmatrix} X\ 24$
Transition Layer (3)	14*14 7*7	1*1 Convolution 2*2 Average pooling, stride 2
Dense Block (4)	7*7	$\begin{pmatrix} 1\ X\ 1\ conv \\ 3\ X\ 3\ conv \end{pmatrix} X\ 16$
Classification Layer	1*1	7*7 Global average pool 1000D fully-connected, SoftMax

Figure 4.8 DenseNet Architecture Layering Details [21]

When it comes to training the DenseNet models, they are easier to train because of better flow of information and gradients in the network scheme. As each layer has access to the source image and the gradients due to being familiarized with the loss function, the overall training of the network becomes easier.

Besides, the network has less over-fitting problems on smaller datasets due to several factors including the regularization and the normalization. Moreover, the bottleneck layers and compression layers could be adjusted and added respectively as per the requirements and the application perspectives.

4.6.3. Inception Networks

The hallmark group emerged with the paper named “Going deeper with convolutions” where it showcased another versatile architecture for deep learning called the ‘Inception Networks’ [16]. The inception network showcased a grand performance in ImageNet challenge in 2014 and provided excellent results in image recognition and detection.

The author in the paper presented a view that increasing the layers of the deep network does not always solve the problem but most of the times result in over-fitting which degrades the performance. On the other hand, if we increase on the hyper parameters, it results in requiring more computational resources. So instead of working with the fully connected networks, the author mentioned to make usage of the sparse connections within the layers. This way, the architecture is designed which required less computation budget, but the depth of the architecture increased along with the width as shown in Figure 4.9.

Based on this principle, the inception V1 architecture is proposed, which has 27 layers in depth (including the pooling layers). Figure 4.10 shows the details of the layers. The inception layer contained the sparse connections, and each inception module had a similar structure as shown in Figure 4.11. Each inception module has a combination of layer of convolutional filters of size 1x1, 3x3 and 5x5.

The output filters of these layers are concatenated into a single image vector at the output forming the filter bank for the next stage. The 1x1 convolutional layer is added after 3x3 and 5x5 layers to reduce the dimensions of the final output. This helps to pick the relevant filter size while learning the model. The layering details of the inception V1 model are shown in Figure 4.12.

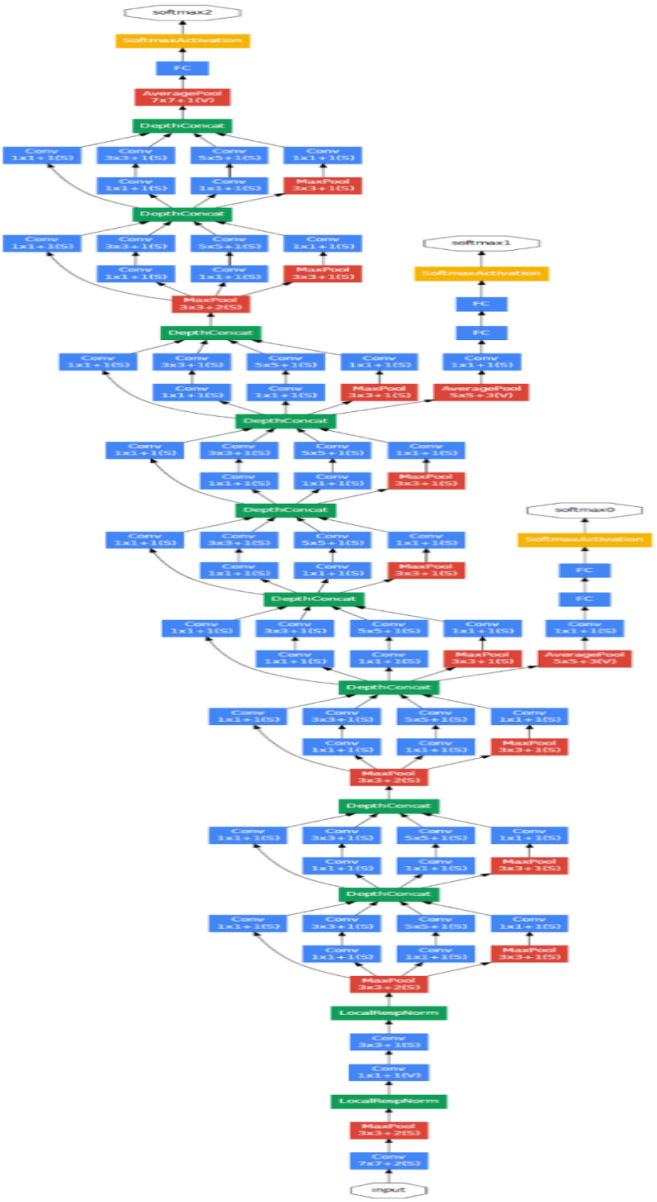


Figure 4.9 Inception Architecture [16]

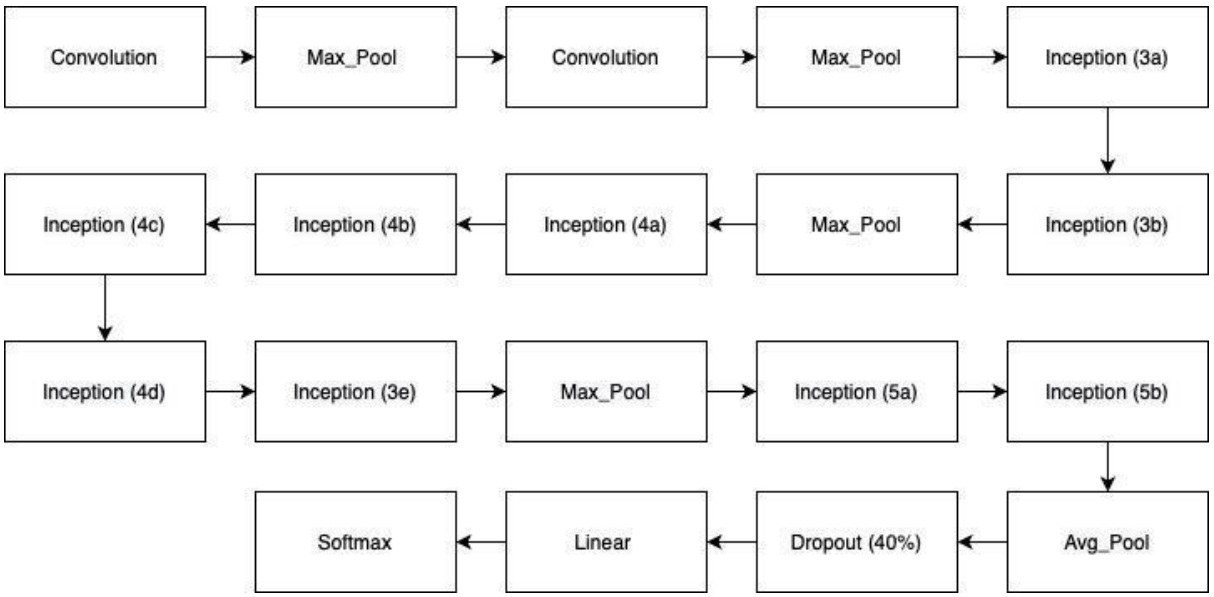


Figure 4.10 Inception V1 Layers Details [16]

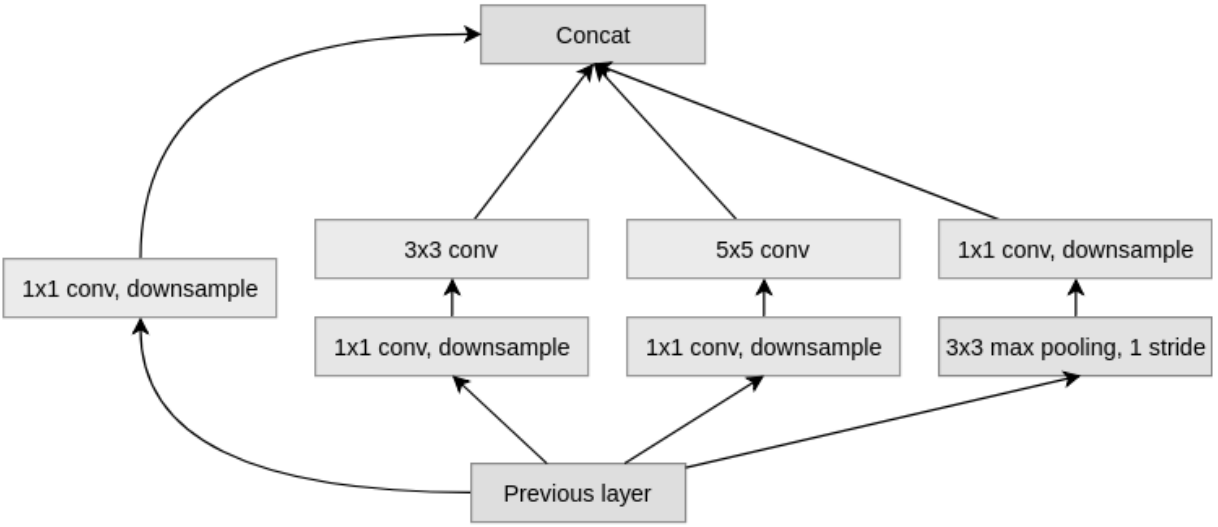


Figure 4.11 Inception V1 Architecture [16]

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

Figure 4.12 Inception V1 Layering Details [16]

To prevent the drying out of the convolutional layers, auxiliary classifier scheme is introduced which applies the SoftMax to the output of the inception module as can be seen in Figure 4.12. The variants of Inception V1 are also present namely InceptionV2 and InceptionV3 by the same group. These were designed by keeping in view the increased accuracy and reduced computational resources and power. In InceptionV2, representational blocks and factorization techniques are targeted, and the high dimensional layers were down sampled which increased the accuracy to about 2.78 times.

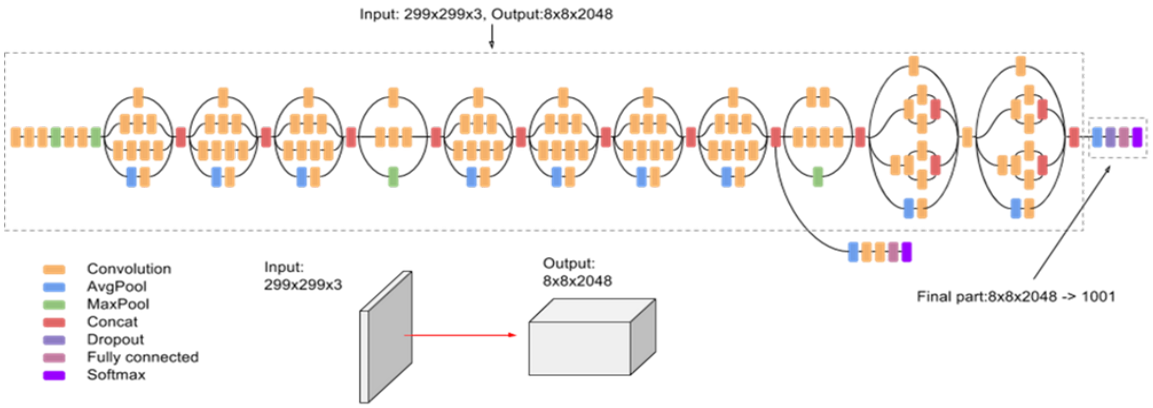


Figure 4.13 Inception V3 Layers Details [51]

While in case of inceptionV3 module, the output side of the network is targeted, and auxiliary classifiers were investigated. Some optimization and factorization were introduced, and the batch normalization is performed before the SoftMax. It provides the benefits of label smoothing by using auxiliary classifiers along with batch normalizations which increased the performance [51]. The algorithm additionally incorporated 7x7 convolutions which helped in more robust training and predictions.

The inception V3 achieves the accuracy of above 78% on ImageNet dataset and is one of the most widely used algorithm for computer vision applications. It contains all the referenced layers of convolutions, average pooling, max pooling, concats, dropouts, and fully connected layers. Batch normalization is performed all around the network and loss is computed via the final SoftMax layer. The architecture is shown in Figure 4.13.

The inception V4 module was developed while taking care of the uniformity of the whole network to achieve even better performance [51]. The module introduced the reduction block to adjust the width and height of the grid.

4.6.4. Inception-ResNet Networks

Afterwards, hybrid Inception modules were designed in conjunction with the ResNet blocks. Two versions of this model were created namely the Inception-ResNet-V1 and Inception-ResNet-V2 with first having the computational cost like InceptionV3 and the second has a computational cost like that of InceptionV4. The stem structures and the hyper parameter details of the two are also different. The difference is shown in Figure 4.14.

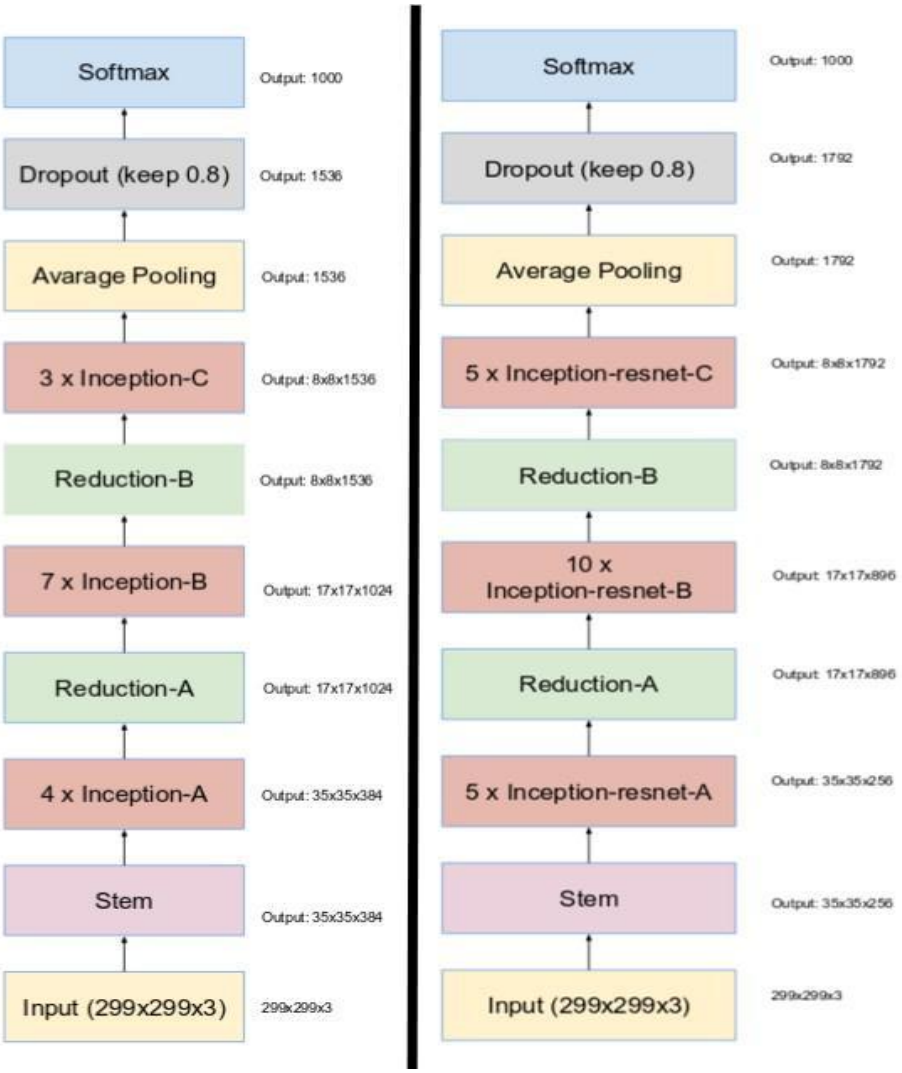


Figure 4.14 Inception V4 vs. Inception-ResNet layering details [19]

The Inception-ResNet-V2 involved the residual connections in place of filter concatenations age. In this model, the batch-normalization is performed on all the traditional layers but not on top of the summation layers as shown in Figure 4.15. The model contains 164 layers and can classify 1000 classes of objects while taking an input image of size 256x256.

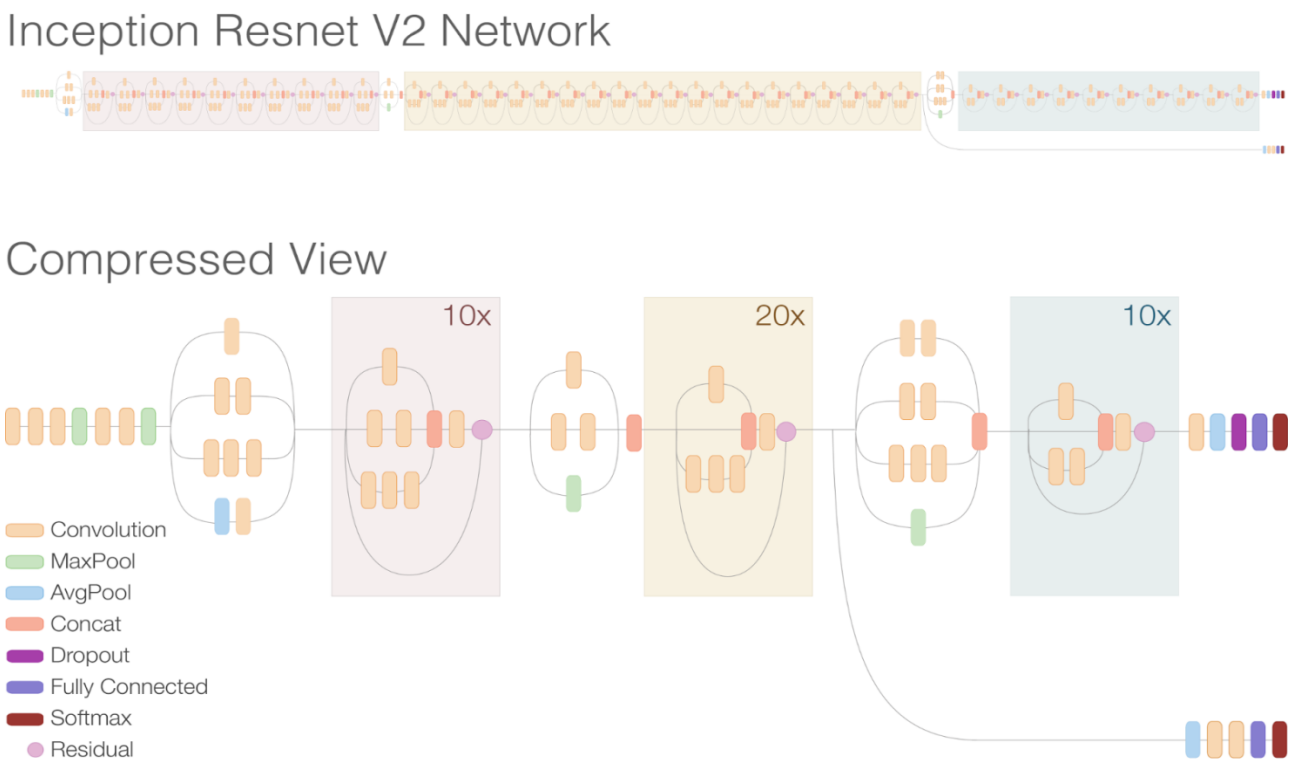


Figure 4.15 Inception-ResNet V2 Architecture [19]

4.6.5. Xception Networks

An extreme version of Inception models was also developed in 2017 and named as Xception which has the capability to perform better than InceptionV3 by incorporating depth wise separable convolutions which means the channel wise nxn spatial convolutions as shown in Figure 4.16 [20]. The point wise convolutions of size 1x1 are done to reduce the dimensions. Overall, the number of performed convolutions is small and the model complexity is less, making it lighter to train. The complete architecture of Xception network model is shown in Figure 4.17.

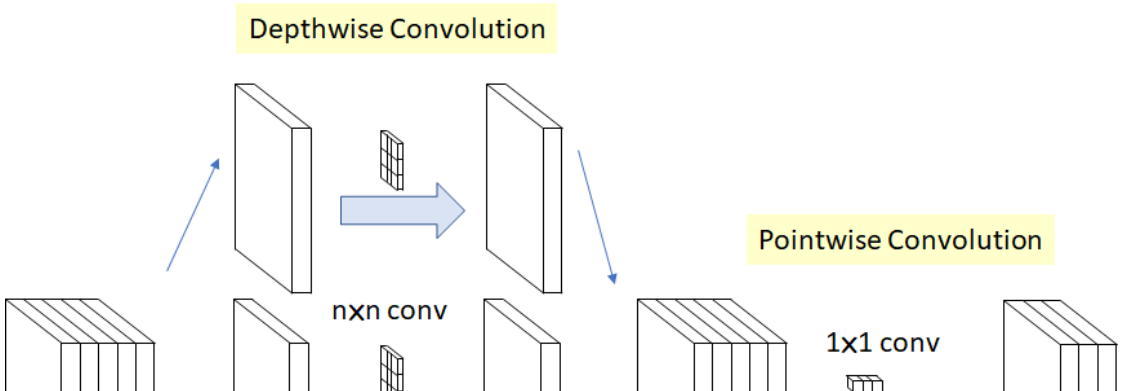


Figure 4.16 Xception Modules [20]

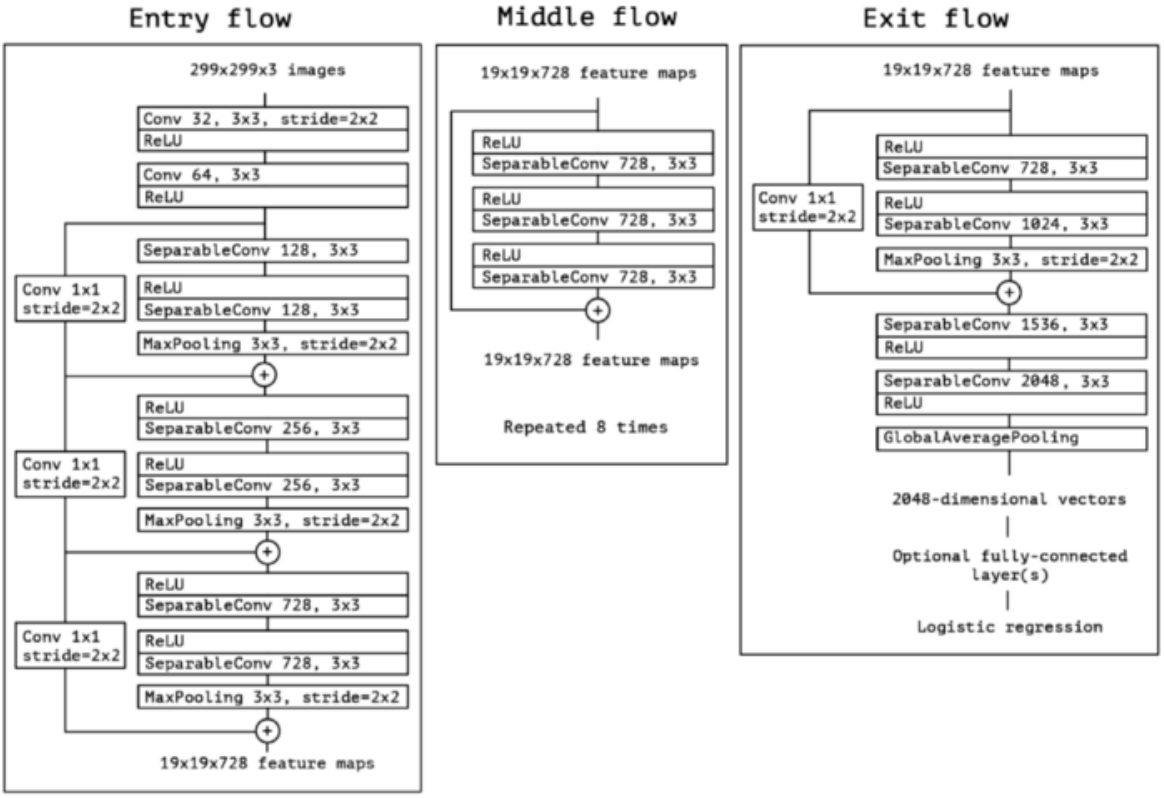


Figure 4.17 Xception Architecture Details [20]

Chapter 5

Results and Discussion

5.1. Survival analysis using VGG-19 structure

The VGG-19 structure details are introduced in Chapter 4. This CNN structure was used with Cox-proportional hazard loss function and Concordance index as evaluation metrics. Adagrad optimizer was used for fitting the model and finding optimal biases, weights, and convolutional kernels and to minimize the risk set. This Survival CNN model was fitted by 15-fold cross validation. In each cross validation the data was split into 80% as training and 20% for validation. The concordance index (CI) and Integrated Brier Score (IBS) for these 15 folds cross validations is presented in Table 5.1.

Table 5.1 Results of VGG19 for 15-fold cross validation

Fold number	Concordance index	Loss	Integrated brier score
1	0.757	1.516	0.116
2	0.758	1.533	0.121
3	0.723	1.588	0.157
4	0.767	1.483	0.109
5	0.799	1.394	0.073
6	0.765	1.452	0.125
7	0.687	1.650	0.229
8	0.701	1.599	0.186
9	0.717	1.527	0.173
10	0.789	1.460	0.065
11	0.724	1.605	0.152
12	0.703	1.648	0.197
13	0.779	1.449	0.096
14	0.753	1.529	0.126
15	0.788	1.489	0.089

The maximum CI in 15 folds cross validation for VGG19 was found to be 0.799. The average CI is 0.747 and the median CI is 0.757. The results are almost equivalent compared with Mobadersany et al, (2018) [40] using histology images only. In the SCNN model reported by Mobadersany et al, (2018) [40]

, the mean C-index was 0.741 and the median C-index was equal 0.745.

Another parameter presented in Table 5.1 for evaluation of the results is integrated brier score. The IBS measure is close to 0.25 for the random model. Smaller values show better fit. The minimum IBS found for VGG-19 in 15-fold cross validation is 0.065. The median and mean for IBS are 0.134 and 0.125, respectively.

The negative partial-log likelihood or Loss value is also presented in Table 5.1. Mean and median for loss are 1.527 and 1.528, respectively.

5.2. Survival analysis using DensNet-121 structure

The DensNet-121 structure detail is introduced in Chapter 4. Like the previous structure, the CNN structure of DenNet-121 was used with Cox-proportional hazard loss function and Concordance index as evaluation metric. Adagrad optimizer was used for fitting the model. This Survival CNN model was fitted by 15-fold cross validation. In each cross validation the data was split into 80% as training and 20% for validation. The concordance index (CI) and Integrated brier score (IBS) for these 15-folds cross validations is presented in Table 5.2.

Table 5.2 Results of DensNet121 for 15-fold cross validation.

Fold number	Concordance index	Loss	Integrated brier score
1	0.815	1.272	0.082
2	0.826	1.407	0.100

3	0.761	1.989	0.176
4	0.825	0.997	0.060
5	0.865	0.182	0.040
6	0.803	0.829	0.087
7	0.696	2.730	0.247
8	0.709	2.206	0.234
9	0.716	1.623	0.193
10	0.873	0.649	0.034
11	0.772	2.092	0.171
12	0.738	2.577	0.249
13	0.841	0.689	0.047
14	0.810	1.403	0.106
15	0.883	0.923	0.030

In DensNet121 the maximum CI is 0.883, The mean and median of the CI are 0.796 and 0.810, respectively.

The IBS of the DensNet10 is also reported in Table 5.2, the minimum IBS is 0.0304, the mean and median of IBS are 0.123 and 0.100, respectively.

The negative partial-log likelihood or loss for DensNet121 are also reported in Table 5.2. The minimum loss is 0.182. The mean and median of loss are 1.438 and 1.403, respectively.

5.3. Survival analysis using Inception-V3 structure

The Inception-V3 structure detail is introduced in Chapter 4. This structure with Cox-proportional hazard loss function and concordance index as metric was used. Adagrad optimizer was used for fitting the model. This Survival CNN model was fitted by 15-fold cross validation. The concordance index (CI) and integrated brier score (IBS) for these 15-folds cross validations is presented in Table 5.3.

Table 5.3 Results of Inception-V3 for 15-fold cross validation.

Fold number	Concordance index	Loss	Integrated brier score
1	0.721	1.569	0.113
2	0.716	1.548	0.098
3	0.727	1.446	0.116
4	0.715	1.614	0.129
5	0.705	1.751	0.151
6	0.696	1.636	0.184
7	0.695	1.311	0.172
8	0.722	1.403	0.163
9	0.709	1.497	0.210
10	0.733	1.681	0.080
11	0.733	1.432	0.091
12	0.711	1.344	0.119
13	0.710	1.666	0.138
14	0.717	1.546	0.114
15	0.710	1.633	0.076

The mean and median of CI for Inception-V3 are 0.715 and 0.715, respectively. The maximum CI is 0.733.

The mean and median IBS for Inception-V3 is 0.130 and 0.119, respectively. The minimum IBS in 15-fold cross validation is 0.076.

The mean and median for loss value for Inception-V3 are 1.538 and 1.548, respectively. The minimum loss is 1.311.

5.4. Survival analysis using InceptionResNet-V2 structure

The InceptionResNet-V3 structure detail is introduced in Chapter 4. This structure with Cox-proportional hazard loss function and concordance index as metric was used. Adagrad optimizer was used for fitting the model. This Survival CNN model was fitted by 15-fold cross validation. The concordance index (CI) and integrated brier score (IBS) for these 15-folds cross validations is presented in Table 5.4.

Table 5.4 Results of Inception-ResNet-V2 for 15-fold cross validation.

Fold number	Concordance index	Loss	Integrated brier score
1	0.737	1.433	0.139
2	0.741	1.480	0.154
3	0.708	1.688	0.198
4	0.741	1.334	0.121
5	0.763	1.043	0.058
6	0.727	1.272	0.123
7	0.665	1.947	0.289
8	0.680	1.764	0.222
9	0.683	1.555	0.184
10	0.772	1.213	0.075
11	0.715	1.725	0.201
12	0.691	1.895	0.261
13	0.750	1.224	0.098
14	0.733	1.479	0.154
15	0.770	1.308	0.116

The maximum CI for Inception-ResNet-V2 model is 0.772. The mean and median CI are 0.725 and 0.733, respectively.

The minimum IBS for Inception-ResNet-V2 model is 0.058. The mean and median IBS for this model is 0.160 and 0.154, respectively.

The minimum loss value is 1.043. The mean and median loss for Inception-ResNet-V2 model is 1.491 and 1.479, respectively.

5.5. Survival analysis using Xception structure.

The Xception structure detail is introduced in Chapter 4. This structure with Cox-proportional hazard loss function and concordance index as metric was used. Adagrad optimizer was used for fitting the model. This Survival CNN model was fitted by 15-fold cross validation. The concordance index (CI) and integrated brier score (IBS) for these 15-folds cross validations is presented in Table 5.5.

Table 5.5 Results of Xception for 15-fold cross validation.

Fold number	Concordance index	Loss	Integrated brier score
1	0.711	1.462	0.112
2	0.718	1.501	0.130
3	0.680	1.674	0.184
4	0.708	1.378	0.091
5	0.721	1.134	0.085
6	0.671	1.321	0.093
7	0.611	1.881	0.294
8	0.635	1.734	0.214
9	0.619	1.557	0.168
10	0.759	1.283	0.034
11	0.697	1.708	0.187
12	0.659	1.843	0.260
13	0.713	1.286	0.063
14	0.705	1.499	0.130

15	0.755	1.358	0.083

The mean and median CI for the Xception model is 0.691 and 0.705, respectively. The maximum CI is 0.759.

The mean and median IBS in 15-fold cross validation is 0.142 and 0.130, respectively. The minimum IBS is 0.034.

The mean and median of loss in 15-fold cross validation for Xception model are 1.508 and 1.4999, respectively. The minimum loss is 1.134.

5.6. Comparing results of CNN structures

In this section, the performance of the 5 CNN structures used in this study are compared together. The Boxplot of the concordance index (CI) for 15 folds cross validation in each of 5 structures is presented in Figure 5.1. In this figure “ICP” means Inception model, “ICP-V3” shows the results for Inception-V3 and “ICP-ResNet-V2” shows the results for Inception-ResNet-V2.

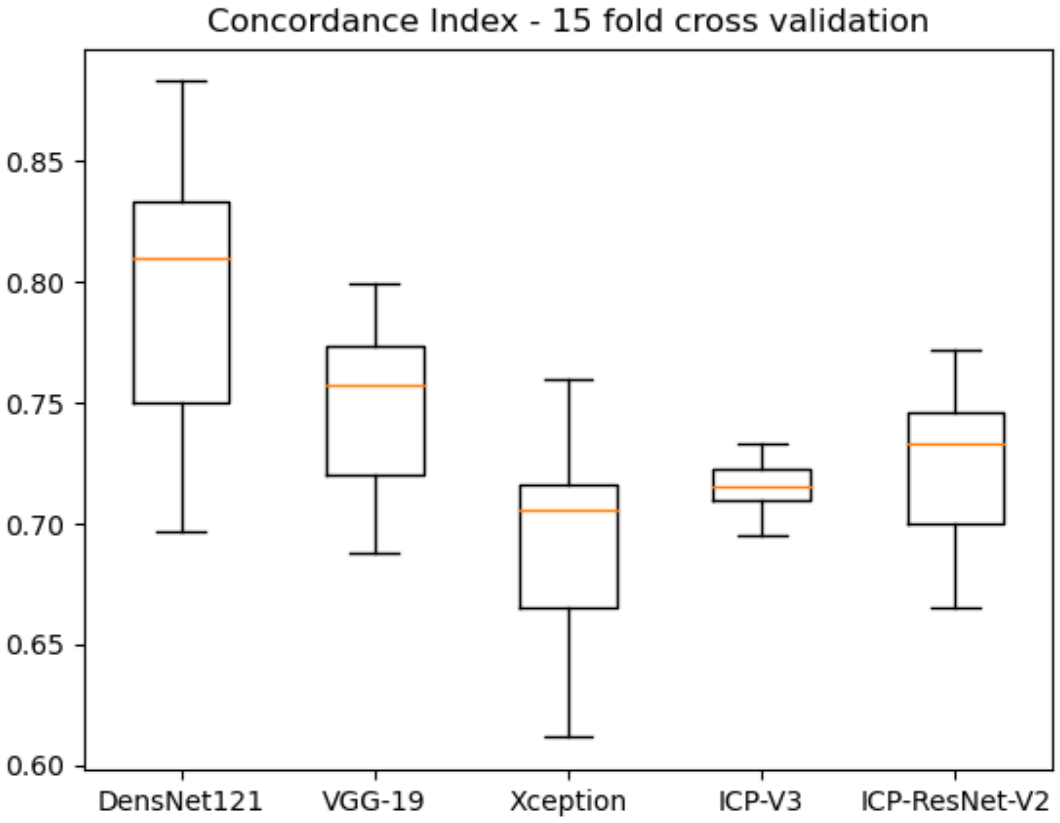


Figure 5.1 Boxplot presenting concordance index in 15 folds cross validation.

It could be seen that DensNet121 model has the highest median Concordance index. The difference between values of CI in these 15-fold cross validation was statistically tested using Wilcoxon Ranksum test. The same random seed for split in cross validation is used, like the analysis was performed in this study. The results could be considered as pairs and hence, Wilcoxon signed rank test which tests the null hypothesis that the difference between two variables is significantly different from zero also is applicable. The statistical test is available in SciPy library of python. The comparison of each of 5 structures are presented in Table 5.6.

Table 5.6 Wilcoxon Ranksum/signed rank test for comparing 5 CNN structures - in term of CI.

Pairwise comparing	Wilcoxon Ranksum	P-value	Wilcoxon Signed Rank	P-value
DensNet121 vs VGG19	2.302	**0.021**	1.0	**0.000***
DensNet121 vs Xception	3.795	**0.000***	0.0	**0.000***
DensNet121 vs Inception-V3	3.380	**0.000***	3.0	**0.000***
DensNet121 vs Inception-ResNet-V2	3.048	**0.002**	0.0	**0.000***
VGG19 vs Xception	3.131	**0.001**	0.0	**0.000***
VGG19 vs Inception-V3	2.467	**0.013**	19.0	**0.018**
VGG19 vs Inception-ResNet-V2	1.596	0.110	0.0	**0.000***
Xception vs Inception-V3	1.721	0.085	27.0	0.063
Xception vs Inception-ResNet-V2	-2.177	**0.029**	0.0	**0.000***
Inception-V3 vs Inception-ResNet-V2	-1.265	0.205	41.0	0.302

* means the p-value is less than 0.001.

The bold values in Table 5.6 shows the p-values which are statistically significant at 5 percent significance level. The CI of DensNet121 is significantly different from other 4 structures. The results of Ranksum test for comparing DensNet121 verse VGG19 shows a p-value = 0.021 < 0.05, which is slightly less than 5% and shows significant difference at 5% significance level.

The-second-high performing model is VGG-19. Pairwise comparison for VGG19 vs other models using Ranksum test show that the CI of VGG-19 is significantly different from Xception, Inception-V3. But the difference between CI of VGG-19 and Inception-ResNet-V2 is not statistically significant. Hence according to the boxplot and the statistical test, we can say that DensNet121 is better than other models followed by VGG-19, then Inception-ResNetV2 and the lowest ranking are Inception-V3 and Xception models.

Another evaluation measure used in this study is integrated brier score, the boxplot for IBS for comparing 5 CNN structures is presented in Figure 5.2. In this figure "ICP-V3" and "ICP-ResNet-V2" indicate Inception-V3 and Inception-ResNet-V2, respectively.

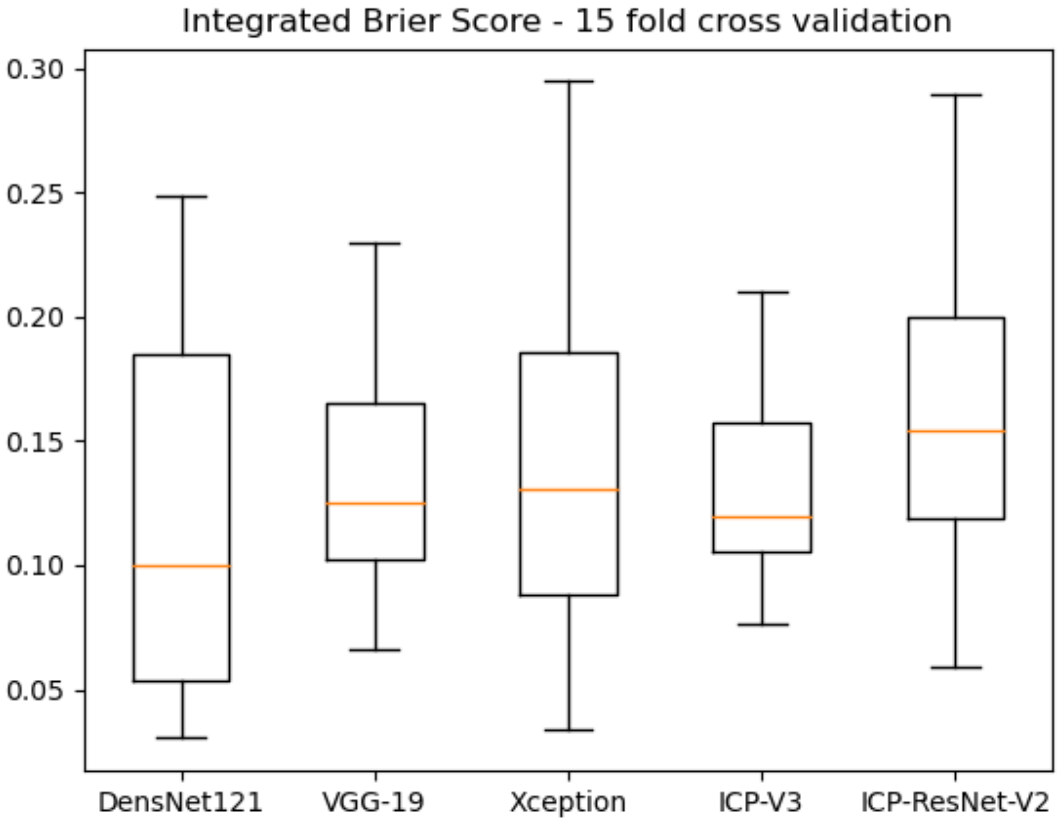


Figure 5.2 IBS of 5 CNN structures using 15-fold cross validation.

In the Figure 5.2, the median of DensNet121 is a bit lower compared with other models. The Wilcoxon Ranksum test were used to compare these 5 CNN structures in term of IBS. The pairwise comparison using Wilcoxon Ranksum test is presented in Table 5.7.

Table 5.7 Wilcoxon Ranksum/signed rank test for comparing 5 CNN structures in term of IBS.

Pairwise comparing	Wilcoxon Ranksum	P-value	Wilcoxon signed rank	P-value
DensNet121 vs VGG19	-1.472	0.140	11.0	**0.0033**
DensNet121 vs Xception	-0.850	0.395	16.0	**0.010**
DensNet121 vs Inception-V3	-0.684	0.493	53.0	0.719

DensNet121 vs Inception-ResNet-V2	-1.430	0.152	3.0	**0.003**
VGG19 vs Xception	0.933	0.350	31.0	0.106
VGG19 vs Inception-V3	1.348	0.177	27.0	0.063
VGG19 vs Inception-ResNet-V2	-0.145	0.884	47.0	0.488
Xception vs Inception-V3	0.103	0.917	51.0	0.638
Xception vs Inception-ResNet-V2	-0.808	0.418	11.0	**0.003**
Inception-V3 vs Inception-ResNet-V2	-1.306	0.191	33.0	0.135

* Means the p-value is less than <0.001.

The Wilcoxon Ranksum test is used to determine if two variables come from the same distribution. For all comparisons, the null hypothesis holds. According to the Wilcoxon Ranksum test, there is no substantial difference between the IBS of the models. However, since the random seed of the splits were the same, the two variables could be considered as paired samples. The Wilcoxon signed rank test shows that the IBS of DensNet121 is significantly less than other models except for Inception-V3. For the other comparisons, only Xception is significantly different from Inception-ResNet-V2. In the Boxplot of Figure 5.2 also it can be seen that the median of IBS of DenseNet121 is a bit lower than other models.

5.7. Prediction on Testing data using 5 CNN structures

After implementing cross validation and training the CNN models, the 5 CNN models were tested using testing data. Test data includes histology images for 153 patients. These 153 patients were not included in the training folder. So, they are considered as new data. The best way to evaluate the machine learning models are by testing them using new dataset which were not included in the training process. The prediction was done as it was explained in summary of method by calculating the median risk for each of 9 High Power Field (HPFs) and then sorting the median prediction in descending order for each region (as we know there

are 266 images for 153 unique patients which means for some of the patients there are more than one region in the test data) and selecting the second highest median prediction as the predicted score for the patient. The Hazard ratio was predicted for each patient. The Table 5.7 shows the predicted Hazard ratio for 8 patients which are included in test folder.

Table 5.8 Hazard Ratio of testing data using 5 CNN models.

Name	Survival	DensNet-121	VGG-19	Xception	ICP-V3	ICP-ResNet-V2
TCGA-02-0006	558	0.865	0.475	0.933	0.238	0.649
TCGA-02-0009	322	0.701	0.549	1.134	0.673	0.613
TCGA-02-0033	86	1.403	0.343	1.537	0.405	0.551
TCGA-02-0048	98	2.071	0.861	0.377	1.069	1.567
TCGA-02-0052	383	0.952	0.525	0.157	0.250	1.019
TCGA-02-0074	310	2.283	1.591	1.480	1.204	3.484
TCGA-02-0085	1561	0.554	0.691	0.327	0.698	0.294
TCGA-02-0258	503	0.452	0.212	1.655	0.181	0.479
Concordant	-	21	16	18	18	19
Discordant	-	7	12	10	10	9
CI	-	0.75	0.57	0.642	0.642	0.678

For comparing reasons 8 patients were included in the table. The predicted hazard ratio for 5 CNN models and the survival of these 8 patients are presented. The Concordance index was calculated manually by using these Hazard ratios and survival times. DensNet121 performs best compared with other models. From 28 comparable pairs, only 7 of them are discordant.

The discordant pairs are {TCGA-02-0006, TCGA-02-0258}, {TCGA-02-0006, TCGA-02-0009}, {TCGA-02-0009, TCGA-02-0052}, {TCGA-02-0033, TCGA-02-0048}, {TCGA-02-0033, TCGA-02-0074}, {TCGA-02-0048, TCGA-02-0074} and {TCGA-02-0085, TCGA-02-0258}. The remaining 21 comparable pairs are concordant. This shows a Concordance index equal to 0.75 for the part of test data that was shown in Table 5.7. After DensNet121, The Inception-ResNet-V2 has the highest CI = 0.678. All 153 patients of testing folder have been tested and compared in 5 CNN structures in Table 5.8.

Table 5.9 Concordance index for the whole 153 test patients

Test Data	DensNet121	VGG-19	Xception	Inception-V3	ICP-ResNet-V2
CI	0.742	0.651	0.546	0.640	0.720

For the testing data, the DensNet121 model includes the highest CI (0.742), the Inception-ResNet-V2 is the second highest model with CI 0.720.

Chapter 6

Conclusions and Future Work

The study was performed using Histology images of the patients to build a survival model which can predict the patients hazard ratio by using the Region of Interest of the Histology images. Five diverse CNN models have been trained, namely VGG-19, DensNet121, Inception-V3, Inception-ResNet-V2 and Xception. The loss function of Cox-proportional hazard has been used to fit a survival model.

6.1. Conclusion for DenseNet121 structure

From five CNN structures, using 15 folds cross validation, it was seen that DensNet121 with the median concordance Index equal 0.81 and Integrated brier score of 0.10 performed best compared with other CNN structures. Wilcoxon Ranksum test has been used to compare the CI and IBS of the DensNet121 with other CNN structures. While CI of DensNet121 were significantly higher than other CNN models, the IBS of DensNet121 using Wilcoxon Ranksum test was not significantly lower than other CNN structures. However, since the random seed for splitting data were same, the data was also considered as paired data and the difference between the IBS of the CNN models in 15 folds cross validation was tested by Wilcoxon signed rank test. The Wilcoxon signed rank test showed that IBS of DensNet121 was significantly lower than other CNN structures, except for Inception-V3. This was seen also in Boxplot that the median of IBS for DensNet121 is less than others, but the IBS of DensNet121 had high variation (interquartile range). Therefore, using Wilcoxon Ranksum test the difference between DensNet121, and other models was not significant at 5% significance level. After 15 folds cross validation the models were tested using testing data.

The results of 15 folds cross validation show a bit higher median CI (0.81) compared with the one found by Mobadersany et al, (2018) [40] using only Histology images (0.754). Testing data included 153 unique patients which were not involved in the training process and were tested using CNN structures. The Hazard ratio was estimated using the 9 High Power Field (HPFs) in the prediction. The median risk for 9 HPs were calculated for each region included in test folder for each patient, since there are 266 images for 153 unique patients, so some of the patients include more than one region in their histology images. The predicted risk for all 153 patients has been calculated. The concordance index of the DensNet121 for testing data was 0.742 which is a bit lower than results of 15 folds cross validation. The CI of DensNet121 remained highest compared with other models in the testing data.

6.2. Conclusion for VGG-19 structure

The VGG-19 structure was another structure which was also used by Mobadersany et al, (2018) [40]. Implementing 15 folds cross validation the second-best performance could be considered for VGG-19. The median concordance index for 15 folds cross validation is 0.757. The median Integrated brier score is 0.134. Using Wilcoxon Ranksum test, the CI of VGG-19 were significantly higher compared with Xception, Inception-V3 but it was not significantly different from Inception-ResNet-V2 at 5% significance level (P-value = 0.11). However, using Wilcoxon signed rank test the CI of VGG-19 were significantly higher compared with Xception, Inception-V3 and Inception-ResNet-V2. In term of IBS both test of Wilcoxon Ranksum and Wilcoxon signed rank test do not show significant difference between IBS of VGG-19 with Xception, Inception-V3 and Inception-ResNet-V2. Prediction was done on 153 unique patients in testing data. The CI of testing data for VGG-19 (0.651) was not higher than Inception-ResNet-V2 (0.72). But it was higher than Xception and a bit higher than Inception-V3.

6.3. Conclusion for Inception-ResNet-V2 structure

The median concordance index for 15-fold cross validation for this structure was 0.733. The median integrate brier score for Inception-ResNet-V2 is 0.154. The Wilcoxon Ranksum test shows that the CI of this model only is significantly higher than Xception model. Using Wilcoxon Ranksum test, the IBS of this model is not significantly different with VGG-19, Xception and Inception-V3. Only using Wilcoxon signed rank test the IBS of Inception-ResNet-V2 were significantly lower than Xception (P-value=0.003). Using testing data of 153 unique patients, this method had the second highest performance compared with other CNN structure with CI = 0.72.

6.4. Conclusion for Inception-V3 and Xception structures

The results of 15 folds cross validation do not show significant difference in Inception-V3 and Xception models for both CI and IBS. Both Ranksum and Signed rank tests do not show significant difference in value of CI and IBS at 5% significance level. These two models were the worst models in these 5 CNN structures. However, using testing data the CI of Inception-V3 (0.64) is 0.1 more than Xception model (0.546).

6.5. Suggestions for future study

The dataset used in this study included a higher fraction of censored data compared with uncensored data. Hence, the consistency of the results of the model could be investigated by using other datasets and preferably with the dataset which include less censored data.

The loss function was computed separately for each batch of size 32 due to memory constraints. Hence, the riskset differs from once batch to another. This makes the convergence of the model to be more time consuming. Using GPU resources with higher volume of RAM seems to be a good idea to get to the convergence to global minimum in loss

function better and faster. Since several regions were included for each patient, so the tied survival events were also added to the higher fraction of the censored data. The robust evaluation needs to be considered with the dataset with less ties and uncensored events.

Another suggestion is to be more restrictive in selecting the Region of Interest. The ROI instead of being selected manually could be selected by implementing unsupervised machine learning techniques. A seemingly good idea is that to use Non-Hierarchical K-means clustering for selecting the ROI regions. The RGBA images included in this study are 1024x1024x4. In augmentation process after objective magnification, the data is randomly cropped to 256x256x4. This augmentation causes losing useful information in some histology images, since it is done quite randomly in the augmentation phase. Hence, the idea of K-means clustering could be replaced in this phase of augmentation to find the centroid of the regions which we are looking for. For example, if the tumour can be seen more in the portion with the highest red colour, the resultant clusters could be checked and the centroid of cluster which has the highest value in Red channel minus Blue channel + Red channel minus Green channel could be selected as the centroid of cropping. Then the augmentation with be done more appropriately compared with random cropping. If large volume of memory in GPU become available, another feasible but time-consuming method is to use the original size of the images for the CNN model instead of randomly cropping them in the augmentation. Since there has been improvement in using the DensNet121 model, it is also suggested to use genomic variables beside histology images and compare the CNN structures using genomic + histology images.

References

1. P. Schober & T. R. Vetter (2018). Survival Analysis and Interpretation of Time-to-Event Data: The Tortoise and the Hare. ***Anesthesia and analgesia***, *127*(3), 792–798. https://doi.org/10.1213/ANE.0000000000003653
2. S. Prinja, N. Gupta, & R. Verma (2010). Censoring in clinical trials: review of survival analysis techniques. ***Indian journal of community medicine: official publication of Indian Association of Preventive & Social Medicine***, *35*(2), 217–221. https://doi.org/10.4103/0970-0218.66859
3. E. Kaplan & P. Meier (1958). Nonparametric Estimation from Incomplete Observations. ***Journal of the American Statistical Association***, *53*(282), 457-481. doi:10.2307/2281868
4. N. Breslow (1975). Analysis of Survival Data under the Proportional Hazards Model. ***International Statistical Review***, *Vol.43*, No. 1, P. 45.
5. D. Cox (1972). Regression Models and Lifetables. ***Journal of the Royal Statistical Society***. *Series B (Methodological), 34*(2), 187-220.
6. M. De Laurentiis & P. Ravdin (1994) A Technique for Using Neural Network Analysis to Perform Survival Analysis of Censored Data. ***Cancer Letters***, vol. 77, no. 2-3, pp. 127-138, 1994. Doi: 10.1016/0304-3835(94)90095-7
7. M. De Laurentiis and P. Ravdin (1994) Survival Analysis of Censored Data: Neural Network Analysis Detection of Complex Interactions Between Variables. ***Breast Cancer Research and Treatment***, vol. 32, no. 1, pp. 113-118, 1994. Doi: 10.1007/bf00666212
8. E. Biganzoli, P. Boracchi, L. Mariani and E. Marubini (1998) Feed Forward Neural Networks for The Analysis of Censored Survival Data: A Partial Logistic Regression Approach, ***Statistics in Medicine***, vol. 17, no. 10, pp. 1169-1186, 1998.

https://doi.org/10.1002/(SICI)1097-0258(19980530)17:10<1169::AID-IM796>3.0.CO;2-D

9. R. Tibshirani (1997). The Lasso Method for Variable Selection in The Cox model**.** ***Statistics in medicine***, *16*(4), 385–395. https://doi.org/10.1002/(sici)1097-0258(19970228)16:4<385::aid-sim380>3.0.co;2-3
10. T. Hothorn, B. Lausen, A. Benner, and M. Radespiel-Tröger (2004). Bagging Survival Trees. ***Statist. Med.***, 23: 77-91. https://doi.org/10.1002/sim.1593
11. J. Friedman, T. Hastie, R. Tibshirani (2000). Additive Logistic Regression: a Statistical View of Boosting (With discussion and a rejoinder by the authors), ***The Annals of Statistics***, Ann. Statist. 28(2), 337-407, (April 2000)
12. C.L. Chi, W. N. Street, & W.H. Wolberg (2007). Application of Artificial Neural Network-Based Survival Analysis on Two Breast Cancer Datasets. AMIA Annual Symposium proceedings. AMIA Symposium, *2007*, 130–134.
13. X. Chen, X. Sun, & Y. Hoshida (2014). Survival Analysis Tools in Genomics Research. ***Human Genomics*** 8**,** 21 (2014). https://doi.org/10.1186/s40246-014-0021-z
14. Y. LeCun, L. Bottou, Y. Bengio, & P. Haffner (1998). Gradient-based Learning Applied to Document Recognition. ***Proceedings of the IEEE***, *86*(11), 2278-2323. https://doi.org/10.1109/5.726791
15. A. Krizhevsky, I. Sutskever, & G.E. Hinton (2012). ImageNet Classification with Deep Convolutional Neural Networks. ***Communications of the ACM***, *60*, 84 - 90. https://doi.org/10.1145/3065386
16. C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, & A. Rabinovich (2015). Going Deeper with Convolutions. *2015* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1-9, June 2015. DOI: 10.1109/CVPR.2015.7298594

17. S. Liu, & W. Deng (2015). Very Deep Convolutional Neural Network-based Image Classification Using Small Training Sample Size. 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), 730-734. DOI: 10.1109/ACPR.2015.7486599
18. K. He, X. Zhang, S. Ren, and J. Sun (2016) Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770-778.https://doi.org/10.1109/CVPR.2016.90
19. C. Szegedy, S. Ioffe, V. Vanhoucke, & A. Alemi (2017). Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. Proceedings of the AAAI Conference on Artificial Intelligence, *31*(1).
20. F. Chollet (2017). Xception: Deep Learning with Depthwise Separable Convolutions. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 1800-1807. DOI: 10.1109/CVPR.2017.195
21. G. Huang, Z. Liu, L. Van Der Maaten, & K.Q. Weinberger (2017) Densely Connected Convolutional Networks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, 21-26 July 2017, 4700-4708. https://doi.org/10.1109/CVPR.2017.243
22. D.A. Freedman (2008) Survival Analysis, ***The American Statistician***, 62:2, 110-119, DOI: 10.1198/000313008X298439
23. A. Hoerl & R. Kennard (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. ***Technometrics****, 12*(1), 55-67. doi:10.2307/1267351
24. M. Leblanc & J. Crowley (1993) Survival Trees by Goodness of Split, ***Journal of the American Statistical Association***, 88:422, 457-467, DOI: 10.1080/01621459.1993.10476296
25. M.F. Gensheimer & B. Narasimhan (2019). A Scalable Discrete-time Survival Model for Neural Networks. *PeerJ*, 7:e6257 https://doi.org/10.7717/peerj.6257

26. S.H. Moolgavkar, E.T. Chang, H.N. Watson, & E.C. Lau (2018). An Assessment of the Cox Proportional Hazards Regression Model for Epidemiologic Studies. ***Risk analysis: an official publication of the Society for Risk Analysis***, *38*(4), 777–794. https://doi.org/10.1111/risa.12865
27. A. Brembilla, A. Olland, M. Puyraveau, G. Massard, F. Mauny, & P.E. Falcoz (2018). Use of the Cox Regression Analysis in Thoracic Surgical Research. ***Journal of thoracic disease***, *10*(6), 3891–3896. https://doi.org/10.21037/jtd.2018.06.15
28. H. Ishwaran, U.B. Kogalur, E.H. Blackstone, M.S. Lauer (2008). Random Survival Forests, ***The Annals of Applied Statistics***, 2(3), 841-860, DOI: 10.1214/08-AOAS169
29. S. Lee & H. Lim (2019). Review of Statistical Methods for Survival Analysis Using Genomic Data. ***Genomics Inform***. 2019;17(4): e41. doi:10.5808/GI.2019.17.4. e41
30. D. Faraggi & R. Simon (1995), A Neural Network Model for Survival Data. ***Statist. Med.***, 14: 73-82. https://doi.org/10.1002/sim.4780140108
31. X. Zhu, J. Yao, & J. Huang (2016). Deep Convolutional Neural Network for Survival Analysis with Pathological Images. 2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), 544-547. Doi: 10.1109/BIBM.2016.7822579
32. J. Yao, S. Wang, X. Zhu, & J. Huang (2016). Imaging Biomarker Discovery for Lung Cancer Survival Prediction. ***Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016***. https://doi.org/10.1007/978-3-319-46723-8_75
33. T. Ching, X. Zhu, L.X. Garmire (2018). Cox-nnet: An Artificial Neural Network Method for Prognosis Prediction of High-throughput Omics Data. ***PLoS Computational Biology*** 14(4): e1006076. https://doi.org/10.1371/journal.pcbi.1006076
34. E. Giunchiglia, A. Nemchenko, & M. Schaar, (2018). RNN-SURV: A Deep Recurrent Model for Survival Analysis. ***ICANN***. https://doi.org/10.1007/978-3-030-01424-7_3

35. K. Matsuo, S. Purushotham, B. Jiang, R.S. Mandelbaum, T. Takiuchi, Y. Liu, & L.D. Roman (2019). Survival Outcome Prediction in Cervical Cancer: Cox Models vs Deep-learning Model. ***American journal of obstetrics and gynecology***, *220*(4), 381.e1–381.e14. https://doi.org/10.1016/j.ajog.2018.12.030

36. K. Simonyan, & A. Zisserman (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556 (2014)

37. S. Otálora, M. Atzori, V. Andrearczyk & H. Müller (2018). Image Magnification Regression Using DenseNet for Exploiting Histopathology Open Access Content. ***Lecture Notes in Computer Science***, vol 11039. Springer, Cham. https://doi.org/10.1007/978-3-030-00949-6_18

38. A. Yang, X. Yang, W. Wu, H. Liu & Y. Zhuansun (2019). Research on Feature Extraction of Tumor Image Based on Convolutional Neural Network, ***IEEE Access***, vol. 7, pp. 24204-24213, 2019, doi: 10.1109/ACCESS.2019.2897131.

39. R. Joshi & C. Reeves (2006). Beyond the Cox Model: Artificial Neural Networks for Survival Analysis Part II.

40. P. Mobadersany, S. Yousefi, M. Amgad, D. Gutman, J. Barnholtz-Sloan, J. Vega, D. Brat, & L. Cooper (2018). Predicting Cancer Outcomes from Histology and Genomics Using Convolutional Networks. ***Proceedings of the National Academy of Sciences***. Doi:10.1073/pnas.1717139115.

41. T.A Gerds, M.W. Kattan, M. Schumacher, & C. Yu (2013). Estimating a Time-Dependent Concordance index for Survival Prediction Models with Covariate Dependent Censoring. ***Statist. Med.***, 32: 2173-2184. https://doi.org/10.1002/sim.5681

42. Y. Liang, H. Chai & X.Y. Liu (2016). Cancer Survival Analysis Using Semi-supervised Learning Method Based on Cox and AFT Models with $L_{1/2}$ Regularization. ***BMC Medical Genomics*** 9**,** 11 (2016). https://doi.org/10.1186/s12920-016-0169-6

43. Y. LeCun, K. Kavukcuoglu, & C. Farabet (2010). Convolutional Networks and Applications in Vision. In *ISCAS 2010 - 2010 IEEE International Symposium on Circuits and Systems: Nano-Bio Circuit Fabrics and Systems* (pp. 253-256). https://doi.org/10.1109/ISCAS.2010.5537907

44. Y. Lecun, & Y. Bengio (1995). Convolutional Networks for Images, Speech, and Time-Series. In M. A. Arbib (Ed.), The Handbook of Brain Theory and Neural Networks, MIT Press.

45. S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a Convolutional Neural Network", 2017 International Conference on Engineering and Technology (ICET), 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186.

46. Q. Li, W. Cai, X. Wang, Y. Zhou, D.D. Feng, & M. Chen (1997). Medical Image Classification with Convolutional Neural Network**.** *In* 2014 13th International Conference on Control Automation Robotics and Vision, ICARCV 2014, pp. 844-848. https://doi.org/10.1109/ICARCV.2014.7064414

47. L. Le, Y. Zheng, G. Carneiro, & L. Yang (2017). Deep Learning and Convolutional Neural Networks for Medical Image Computing. ***Advances in Computer Vision and Pattern Recognition (2017)***. Doi: 10.1007/978-3-319-42999-1

48. B. Lo Shih-Chung, H.P. Chan, J.S. Lin, H. Li, M.T. Freedman& S.K. Mun (1995). Artificial Convolution Neural Network for Medical Image Pattern Recognition. ***Neural networks*** 8.7-8 (1995): 1201-1214. https://doi.org/10.1016/0893-6080(95)00061-5

49. W. Zhang, P. Tang, L. Zhao (2019) Remote Sensing Image Scene Classification Using CNN-CapsNet. ***Remote Sensing***. **2019**, *11*, 494. https://doi.org/10.3390/rs11050494

50. Y. Zheng, C. Yang, & A. Merkulov (2018). Breast Cancer Screening Using Convolutional Neural Network and Follow-up Digital Mammography. ***International Society for Optics and Photonics, 2018.*** Doi: 10.1117/12.2304564

51. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, & Z. Wojna (2016). Rethinking the Inception Architecture for Computer Vision. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2818-2826. Doi: 10.1109/CVPR.2016.308