Tampere University

Umair Akhtar Hasan Khan

Prostate Cancer Detection Using Deep Learning

Faculty of Information Technology
and Communication Sciences
M.Sc. Thesis
March 2019

ABSTRACT

Umair Akhtar Hasan Khan : Prostate Cancer Detection using Deep Learning
M.Sc. Thesis
Tampere University
Degree Programme in Computer Science
April 2019
Supervisors: Martti Juhola and Kati Iltanen

Cancer detection is one of the principal topics of research in medical science. May it be breast, lung, brain or prostate cancer, advances are being made to improve detection precision and time. Research is being carried out on broad range of procedures at different stages of cancer to understand it better. Prostate cancer, in particular, has seen some novel approaches of detection using both magnetic resonance imaging (MRI) and histopathology data. The approaches include detection using deep neural networks, deep convolutional neural networks in particular because of their human level precision in image recognition task.

In this thesis, we analysed a dataset containing multiparametric magnetic resonance imaging (mpMRI) prostate scans. The objective of the research was Gleason grade group classification, through mpMRI scans, which has not been attempted before on a small dataset. We first trained several conventional machine learning algorithms on handcrafted features from the dataset to predict the Gleason grade group of the cases. After that the dataset was augmented using two different augmentation techniques for further experimentation with deep convolutional neural networks. Convolutional neural network of varying depth were used to understand the effects of network depth on classification accuracy. Furthermore, we made an attempt to shed light on the pitfalls of using small dataset for solving problems of such nature.

Keywords: Deep Neural Networks, multiparametric Magnetic Resonance Imaging (mpMRI), Convolutional Neural Networks

The originality of this thesis has been checked using the Turnitin Originality Check service.

ACKNOWLEDGEMENT

I would like to take the opportunity to thank Timo Heikkinen (CEO, Top Data Science) for hiring me for this research project and providing me the opportunity to work on cutting edge technologies. I'd like to thank my supervisors Martti Juhola (Professor, Tampere University) and Kati Iltanen (Dean, Faculty of Natural Sciences, Tampere University) for their support and guidance. I feel very fortunate to have them as my supervisors, their positive feedback and constructive criticism always pushed me to refine my work to the highest level.

Last but not the least, I am grateful to Oguzhan Gencoglu (Head of Data Science, Top Data Science) for all the guidance and knowledge he shared with me, not only did he guide me throughout the course of this research project but also inspired me to deliver quality work. I am impressed with his attention to detail and dedication to the project. Working with him helped me instill the same traits in myself.

Once again, I'd like to express my gratitude to all the people mentioned above, without them I would not have been able to execute this project.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ADC	Apparent Diffusion Coefficient
AUC	Area Under the Curve
BVAL	B-Value
CAD	Computer Aided Diagnosis
CNN	Convolutional Neural Network
DCE	Dynamic Contrast Enhanced
DL	Deep Learning
DWI	Diffusion-Weighted Imaging
GGG	Gleason Grade Group
HOG	Histogram of Oriented Gradients
ILSVRC	Imagenet Large Scale Visual Recognition challenge
KS	Kappa score
LBE	Lesion-based Evaluation
LBP	Local Binary Pattern
LDA	Linear Discriminant Analysis
LR	Logistic Regression
mp-MRI	Multiparametric Magnetic Resonance Image
PCA	Principal Component Analysis
PI-RADS	Prostate Imaging Reporting and Data System
PssAgg	Passive Aggressive Classifier
QDA	Quadratic Discriminant Analysis
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
R-CNN	Regional Convolutional Network
ROC	Receiver Operating Characteristic
ROI	Region of Interest
SBE	Slice-based Evaluation
SVC	Support Vector Classifier
SVM	Support Vector Machine
TF	Tensorflow
TSE_SAG	T2-Weighted Sagittal
TSE_TRA	T2-Weighted Transaxial
TSNE	t-Distributed Stochastic Neighbor Embedding
WSI	Whole-Slide Image
VGG	Visual Geometry Group

1. INTRODUCTION

Prostate cancer is the most common cancer in men after skin cancer in America. One man in nine will develop clinically significant prostate cancer and one man in forty one will die of it. A combination of invasive and noninvasive techniques, such as Transrectal ultrasound guided (TRUS) biopsy and Multiparametric Magnetic Resonance Imaging (mpMRI) respectively, have enabled us to accurately diagnose the aggressiveness of prostate cancer. The type and course of treatment is then decided, based on the stage of the cancer, it could be any of the following surgery, radiation therapy, cryotherapy, hormone therapy or chemotherapy. ["Key Statistics for Prostate Cancer", 2018].

Having said that, during the last two decades, a separate vertical, although not directly related to medical science, has been analysing medical data and making strides, especially in diagnosis. Computer scientists, with the help of image processing and machine learning algorithms, have been fine tuning their systems to diagnose different types of cancerous nodules, cysts and tumors. They call it Computer-aided Diagnosis (CaD). Some of the recent endeavours have surpassed human accuracy. These CaD techniques are driven by Deep Neural Networks which are discussed in the following chapters. At this stage the idea is to not completely eliminate human involvement from the diagnosis process, rather to introduce a hybrid system consisting of CaD as well as human diagnosis where both systems are complementary to each other, resulting in significant improvement in cancer detection.

1.1 Prostate Cancer

The prostate is a gland responsible for making most of the sperm carrying semen in the male reproductive system. It sits between the bladder and the upper part of urethra which carries the urine from the bladder. ["The Basics of Prostate Cancer", 2018].

Prostate cancer is quite common in Finland with more that 4700 cases of prostate cancer are recorded each year in Finland ["Docrates is the first oncology center in Finland to offer new radiation therapy for metastatic prostate cancer", 2018]. Cancer at an early stage is curable, a vast majority patients undergo treatments which completely cure them. Therefore, early diagnosis of cancer is pivotal in curing cancer. However, prostate cancer at an advanced stage can not be completely cured and requires perpetual treatment. In case of American men about 85% of cases diagnosed with prostate cancer are in early stage of the disease, where it has not spread much ["The Basics of Prostate Cancer", 2018].

If the cancer spreads beyond the prostate for example to bones, lungs or lymph nodes, then it is no more curable, but advancement in medical science has enabled us to keep it under

control and the life span of the patient can be extended to several years. In some cases it has been observed that patients with advanced prostate cancer lived for many years and their cause of death was entirely different from prostate cancer.

Prostate cancer is generally found in older men. Over 80% of men diagnosed with prostate cancer are over 65 years of age and less than 1% are under 50. Susceptibility to prostate cancer increases if a person has family history of it. There is not yet an established research on the exact cause of cancer; however, prostate cancer is linked with certain types of diet which include overconsumption of fats and red meat. Substance produced as a byproduct of meat cooked on high temperature is dangerous for a prostate. The rate of prostate cancer varies in countries, based on their food consumption. It is common in countries with high consumption of dairy products and meat as compared to countries with vegetable and rice based diet.

Hormones are another factor. Testosterone increases with higher consumption of fats which works as a catalyst in the growth of prostate cancer. Lack of exercise also makes prostate cancer more likely. A few occupational hazards have also been found to be contributing factors to prostate cancer. For example, some jobs in rubber and battery manufacturing require workers to be exposed to metal cadmium, which make them prone to getting prostate cancer. Drugs such as aspirin, finasteride (Proscar) and dutasteride (Avodart) have proven to reduce the risk of developing prostate cancer. Similarly, regular consumption of some vegetables such as broccoli, cauliflower, and cabbage have also been a deterrent to prostate cancer [“The Basics of Prostate Cancer”, 2018].

1.2 Machine Learning

The overlap of the fields of computer science and statistical methods gave birth to the field of machine learning. Machine learning enables computer programs to progressively learn which does not necessitate any explicit programming assistance. Arthur Samuel came up with the term machine learning. [Samuel, 1988]. The field of machine learning evolved as the amalgamation of pattern recognition and computational learning theory in artificial intelligence [“Machine learning | artificial intelligence”, 2018]. The aim of machine learning is the construction of such algorithms that can learn from data and make predictions on unseen data, the outcome of such algorithms are not contingent on any strict programmatic logic. Primarily there are two types of learning, supervised and unsupervised. [Mohri et al., 2012; Trucker, 2004]

Supervised learning: In supervised learning learning the data is fed to the program along with the corresponding labels, the theme is to learn a function that maps the data to its label. However, there are special cases of

- *Semi-Supervised learning:* In semi-supervised learning the training data has incomplete labels, in most cases the input data has missing labels.

- *Reinforcement learning:* In reinforcement learning the data is fed to the algorithm from a dynamic environment in form of rewards and punishments. It could be playing a first person shooting game against multiple opponents or flying a plane in a simulated environment.

Unsupervised learning: In unsupervised learning, no labels are provided to the learning algorithm, the algorithm relies on hidden patterns and structure in the data for learning. An auxiliary purpose of unsupervised learning could be to find out the intrinsic patterns or learn features of the data.

Machine learning can also be categorized on the basis of the type of desired output of a machine learning system.

Classification: In classification the input data is divided into two or more classes, the learning algorithm learns a mapping function during training and assigns one or more of these classes to an unseen data point either during testing or in an actual use case. This is typically done is a supervised way. [Alpaydin, 2010]. Figure 1 visually illustrates support vector machine (SVM) [Cortes & Vapnik, 1995], a classifier that separates its input space into two regions using a linear boundary.

Regression: Unlike classification the output variable of regression is continuous. Regression too is usually done in a supervised way [Freedman, 2009].

Clustering: Clustering is an unsupervised task in which the input is divided into k number of groups using different clustering algorithms [Bailey, 1994].

Density Estimation: Density estimation finds the distribution of inputs in some space [“2.8. Density Estimation”, 2018].

Dimensionality Reduction: It is a process of transforming high dimensional data to lower dimension [Roweis et al., 2000]. One of the applications of dimensionality reduction is visualisation of data in either two of three dimensional space.

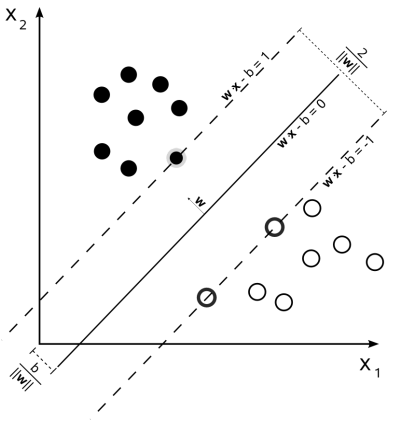

Figure 1: Classification using support vector machine (SVM).

In the present era it is difficult to find a field which is not disrupted by machine learning or to say the least affected by it. May it be stock predictions, virtual personal assistance, video

surveillance, social media, customer support, spam filtering, weather prediction or online fraud detection; machine learning has affected each one of them. Among these, the field which has seen very serious efforts and interest is medical science. Machine learning is now solving as complicated problems as cancer/tumor detection which is a non-trivial problem by any definition. Computer aided diagnosis solutions for cancer detection are at the crossroads of computer vision and machine learning. Over time these solutions have gone through an evolutionary process which tremendously improved their efficiency and effectiveness.

1.3 Artificial Neural Networks

The idea of artificial neural networks is roughly based on the working of a human brain, it is inspired by the biological neural network. An artificial neural network (ANN) based system learns progressively without task specific programming [“Artificial Neural Networks as Models of Neural Information Processing”, 2018]. For instance, an ANN capable of image recognition learns to identify objects that are manually labelled, e.g. it can tell apart images that contain a car from images that do not contain a car. These networks can learn without any prior information about the features of a car.

An artificial neural network consists of connected layers of nodes called neurons, much similar to neurons in an animal brain. Connections between neurons simulate the function of synapse by transmitting information from one neuron to another. After receiving a signal, an artificial neuron can process it based on a certain function and can pass it to other artificial neurons. ANNs are implemented in a way that a real number passes as a signal through connections between artificial neurons. The output of each artificial neuron is calculated by a nonlinear function of the sum of its inputs. Part of the nonlinear function is a weight which modulates as the network learns during the training process. Weight is pivotal in determining the strength of the signal over a connection. Artificial neural network generally consists of layers stacked with artificial neurons, layers may differ from each other with respect to how they transform the inputs. Usually, signals iterate from first layer to the last, several times optimally adjusting the weights of artificial neurons in each iteration. [“Artificial Neural Networks as Models of Neural Information Processing”, 2018].

Originally the idea of ANN was inspired by human brain; however, the focus deviated from biological path over time in order to solve specific problems. Presently, ANNs are being used in a wide range of fields such as natural language processing, computer vision, speech recognition, video games, and medical science. Figure 2 is a visual representation of a basic artificial neural network.

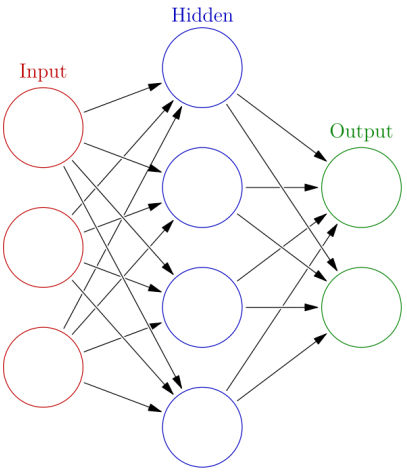

Figure 2: Visualisation of a basic neural network.

1.4 Deep Learning and Deep Neural Networks

Deep learning is a sub-discipline of machine learning that is philosophically based on in-depth learning from data, comprising of several layers of representations. The learning takes place in a hierarchical fashion, where learning at higher layers is extracted from learnings of lower layers. This process of concept learning in the domain of deep learning is called feature learning or representation learning. Input (for instance image data) can be represented in different ways (like a matrix containing pixel information), but some representations are more efficient than other in terms of learning features from data, and rapid advancements are being made in the area to make representation more efficient. [Deng & Yu, 2014]

Generally, artificial neural networks are the key component of deep learning models. However, certain types can comprise of syntactic formula or latent (unobserved) variables structured layer-wise in deep generative models, for instance the nodes in Deep Boltzmann Machines.

Learning in deep networks varies from layer to layer, in terms of level of abstraction and transformation of input data. In case of image processing, the data could be a 3 dimensional array of pixels: the first representational layer captures the understanding of edges, the second representational layer may capture the sequence or arrangement of edges, the third representational layer may capture features of a car such as formation of tyres, windscreen, headlights, wipers and the fourth layer may discern the presence of a car in the image. Deep learning is capable of appropriately allocating the feature learning to specific layers and that too is learned. This, however, does not eliminate the need of manual configuration of a network, for instance, the number and size of layers have big impact on the performance and output of a deep learning model. [Bengio et al., 2012; LeCun et al., 2015]

Deep learning can be employed for both supervised and unsupervised learning. In supervised learning it eliminates the process of feature engineering by learning terse transitional representations similar to principal components. Unsupervised learning is a good use case for deep learning because of the abundance of unlabeled data, it can be done using deep belief networks [Hinton, 2009] and neural history compressors [Schmidhuber, 2015]. Figure 3 is a visual representation of a fully connected deep neural network.

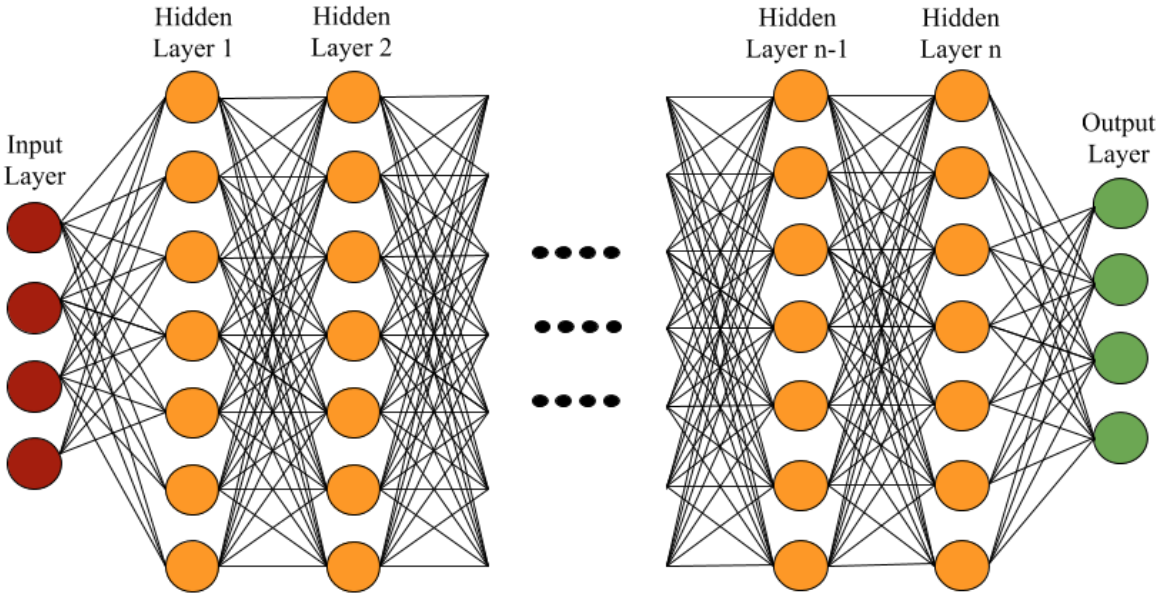

Figure 3: Visualisation of a fully connected deep neural network with n hidden layers.

1.5 Thesis Problem Statement

In this thesis research, the analysis of the mp-MRI scans and metadata lead to the formulation of the problem statement. The idea was to use the mp-MRI scans and train a machine learning model which can later be employed to detect the prostate cancer aggressiveness based on Gleason grade group of unseen cases. The 5 different Gleason grade groups make it a multi-class classification problem. Multi-class classification is a more complex problem than binary classification of clinical significance of cancer given the class imbalance and small size of the dataset. Moreover, an auxiliary challenge was the small size of the dataset, which was dealt with by engineering suitable techniques to augment the data.

1.6 Aim and Objective

The aim of this research was to use a small dataset and test how effectively deep learning can be applied on it to get meaningful results. Furthermore, a smaller dataset compelled us to experiment with different augmentation techniques as well.

2. LITERATURE REVIEW

A considerable amount of work has been done, both, in the domain of prostate cancer and the technology (computer vision and artificial intelligence). Research on cancer detection, in general, has gained a lot of traction in recent past. Research is being conducted using different types of artifacts such as mpMRI scans, CT scans and histopathological slides to understand and detect cancer better. There has been an increase in online competitions for cancer detection and its clinical significance. These competitions include breast, prostate, lung and brain cancer detection. But medical science is a domain, riddled with scarcity of data and it is one of the biggest challenges to overcome before we even begin to think about ANN model building and classification. This section elucidates the cutting edge research being done in the CAD area for cancer detection, prostate cancer detection in particular. It also explains the inner workings of the most advanced image recognition artificial neural networks being used in ImageNet Large Scale Visual Recognition Competition [Russakovsky et al., 2014].

2.1 Computer-aided Diagnosis (CAD) for Prostate Cancer

Computer-aided diagnosis (CAD) are systems that help doctors and physicians interpret and understand medical images better. Images captured through X-rays, MRI and microscopes contain a tremendous amount of minute details which are supposed to be analysed and interpreted by radiologist, histopathologists and other medical professionals in a relatively short amount of time. Typically the idea behind CAD systems is to accentuate anomalous information in the scans in order to make it discernable for medical professionals to aid their decision.

Moore's law (roughly) states that the processing power doubles every 18 months [Moore, 2006], the paper was written in 1965 and the law still holds. With ever so fast processors and advancement in the hardware field, another field has catched up and progressed quite a lot in recent years, that is machine learning. In today’s age, every field is employing machine learning driven solutions to bring about performance improvements, which were never witnessed before. CAD systems also happen to be one of the areas where machine learning has achieved human level results and there are instances where machine learning driven solutions proved to have done better than humans.

2.1.1 Conventional Machine Learning based CAD

As mentioned earlier that one of the biggest challenges in the field of medical science, when it comes to machine learning driven systems, is the dearth of data. Since a typical machine

learning system requires a large and comprehensive dataset in order to have good generalising power; therefore, researchers came up with a workaround to tackle this challenge. They started using synthetic data to train machine learning models.

A solution proposed by [Fehr et al., 2015] solved the data size problem using Synthetic Minority Oversampling Technique SMOTE [Chawla et al., 2002] and Gibbs sampling [Geman & Geman, 1984]. The motivation behind the solution was to improve the efficacy of the non-invasive MRI scan in detecting initial stage benign cancer without necessarily subjecting patients to invasive procedures such as biopsy. The fundamental idea of the research was to perform classification based on Gleason scores, distinguishing Gleason scores 6 (3 + 3) from scores above or equal to 7 and Gleason score 7 (3 + 4) from 7 (4 + 3). The research used two different image modalities T2-weighted MR images and apparent diffusion coefficient ADC to calculate image texture features.

T2-weighted and ADC MR images were preprocessed and two sets of texture based features were engineered from these images. The first set of features included moments of the intensity volume histogram (mean, SD, skewness, and kurtosis) computed from the region of interest. The second set of features consisted of Haralick features [Haralick et al., 1973] was computed using the gray level co-occurrence matrix (GLCM) with 128 bins and consisted of energy, entropy, correlation, homogeneity, and contrast. Both sets of features were then used to train SVM [Cortes & Vapnik, 1995], SVM with (t-test and RFE) and Adaboost models. It was observed that even with a high imbalance in the classes, data augmentation helped obtaining high specificity and sensitivity. Finally, SVM combined with recursive feature elimination yielded best classification accuracy, distinguishing Gleason score 6 (3 + 3) from scores above or equal to 7 with 93% accuracy and Gleason score 7 (3 + 4) from 7 (4 + 3) with 92% accuracy.

Another solution took a similar approach of using conventional machine learning algorithms such as SVM and decision trees with image features based on texture, morphological scale invariant feature transform (SIFT) [Cortes & Vapnik, 1995; Lowe, 2004], and elliptic Fourier descriptors (EFDs) [Soldea et al., 2010] to detect clinically significant prostate cancer [Hussain et al., 2018]. The dataset used in this study is publicly available by Harvard University (National Center for Image Guided Therapy Department of Radiology, Brigham and Women's Hospital, Harvard Medical School). The database is publicly available for research purposes. The study used a total of 682 MRI scans from 20 patients consisting of 482 images from prostate subjects and 200 images from brachytherapy subjects for the purpose of feature extraction and to use those features to detect cancer.

Models were trained using single as well as combination of feature sets for the task of classification. Jackknife [Efron, 1979] and k-fold were employed as cross validation techniques to ensure performance evaluation validity. Receiver Operating Characteristic (ROC), specificity, sensitivity, positive predictive value (PPV), negative predictive value

(NPV), false positive rate (FPR) were used to evaluate the strength of model. For single feature sets, SVM with Gaussian kernel gave the highest accuracy of 93.34% with an AUC score of 0.999. Whereas, for a combination of different feature sets SVM Gaussian kernel with texture morphological features gave the highest accuracy of 99.71% and AUC of 1.00.

A study conducted in 2015 took a rather unconventional approach of combining a CAD system and prostate imaging reporting and data system (PI-RADS) score ["Global PI-RADS Standardization of Prostate MRI", 2018] to detect prostate cancer [Litjens et al., 2015]. The approach used in the study is a three step approach. In the first step, the system extracted quantitative voxel features from mpMRI scans, following PI-RADS guidelines in order to capture characteristics described by the PI-RADS guidelines. These voxel features were used as input to train a random forest classifier to determine a continuous likelihood score for each voxel for cancer identification, resulting in a likelihood image. The second step predicted cancer likelihood per lesion using a random forest classifier, trained by using a combination of the symmetry, local contrast and statistical features. Probability based segmentation of the prostate zones enabled the system to consider a lesion's zonal location. The third and final step drew a contrast between CAD, PI-RADS and their combination using logistic regression and evaluated them based on ROC AUC score. Figure 4 describes the workflow of a CAD system.

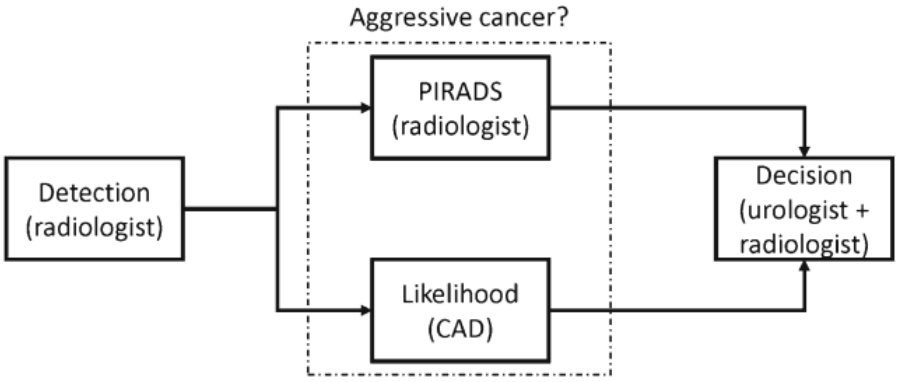

Figure 4: Suggested workflow for the proposed CAD system [Litjens et al., 2015].

The final analysis of the study included a total of 107 patients with 141 lesions. The ROC AUC score of the combination (CAD and PI-RADS) was significantly higher than the PI-RADS only score of the radiologist. In case of benign vs cancer, the combination score was 0.88, whereas radiologist score was 0.81 and the p-value was 0.013. In case of indolent vs. aggressive, the scores were 0.88 and 0.78 with p-value < 0.01 for combination and radiologist respectively. The actual cancer grade and combination score had a higher correlation (0.69, p-value=0.0014) as compared to the individual CAD system or radiologist alone (0.54 and 0.58 respectively). It was concluded that the solution based on the combination of CAD and PI-RADS had the potential to improve cancer detection accuracy.

2.1.2 Deep Learning based CAD

Ever since deep learning started making strides in image recognition and outclassed other conventional techniques, it has become synonymous with a panacea. Researchers with highly efficient GPU powered compute resources and advanced deep learning libraries such as TensorFlow at their disposal, are using these latest resources to come up with CAD systems to solve primitive problems. Deep neural networks are being used not only to detect prostate cancer but also segmentation of prostate in three-dimensional space. A special kind of neural network called Generative Adversarial Network (GAN) [Goodfellow et al., 2014] is found to be very effective in solving insufficient data problem. It generates very similar synthetic data points based on features learned from the actual dataset.

A study in 2017 drew a comparison between traditional machine learning CAD and deep learning based CAD [Tsehay et al., 2017]. The approach used in this study was based on an edge detector network, which processes input images to generate their respective probability maps. The performance of this deep learning based CAD was evaluated using receiver operating characteristic (ROC) curve and free-response ROC (FROC). Then the performance of deep learning based CAD was compared with an already existing CAD which used SVM with handwrought features, such as local binary pattern (LBP) [He, et al., 1990] extracted from the same dataset. The study included data of 52 patients in three different modalities, namely, T2W, ADC, and B2000 MR images. Since these modalities captured information about the same instance of the organ stored in different ways; therefore, it provided room to employ a novel approach of superimposing these modalities to create a three-channel RGB image as shown in Figure 5. The enriched three-channel input allowed the network to exploit the contrast of the image to learn meaningful features. Furthermore, the contrast of the RGB image was enhanced using histogram equalisation.

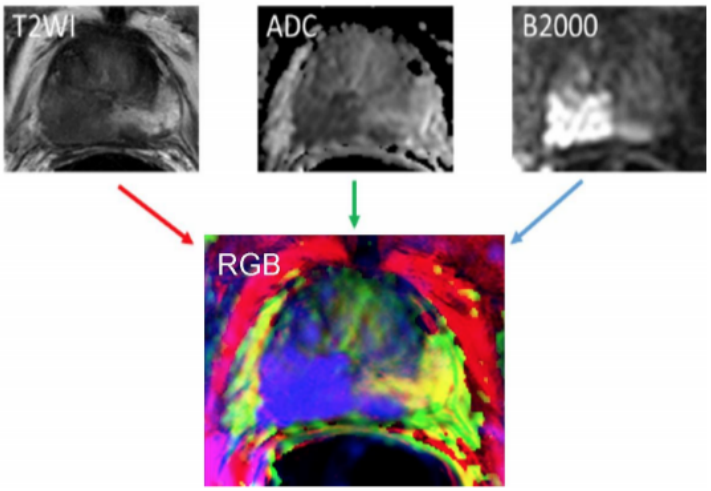

Figure 5: Generating a three-channel RGB image from three mp-MRIs [Tsehay et al., 2017].

The performance of deep learning based CAD was compared with the SVM based CAD that used the handwrought features. DL based CAD had an 86% true positive rate and 20% false positive rate. Whereas, the SVM based CAD had 80% true positive rate and 20% false positive rate, which when projected on the FROC came out to be 94% and 85% detection rate at 10 false- positives per patient. The results were conclusive enough to establish that DL based CAD had potential and further exploration could lead to even better results.

A study which emerged as a result of an online competition PROSTATEx challenge ["PROSTATEx Grand Challenge", 2018] used a novel deep learning architecture named XmasNet to detect clinical significance of prostate cancer [Liu et al., 2017]. The deep learning architecture (XmasNet) comprised of several convolutional layers. The dataset comprised of 341 cases and there were 4 different multiparametric MRI modalities, namely, diffusion weighted images (DWI), apparent diffusion coefficient (ADC), Ktrans and T2-Weighted images. A unique way of data augmentation based on 3D rotation and slicing was employed to incorporate the information of the lesion as shown in Figure 6 part c. The study used a similar approach of superimposing modalities used in the aforementioned study [Tsehay et al., 2017] to create a three-channel RGB image as shown in Figure 6 part a. The training set comprised of 274 lesions and validation set had 43 lesions. Based on the superimposition technique, four different types of inputs were generated using DWI (D), ADC (A), Ktrans (K), and transverse T2WI (T) as the RGB channels: DAK, DAT, AKT, DKT (shown in the figure before). Separate deep learning models were trained for each input type. 207144 training samples were generated after augmentation using in-plane rotation, random shearing and translation for every slice. Each sample was 32x32 region of interest (ROI), surrounding the lesion.

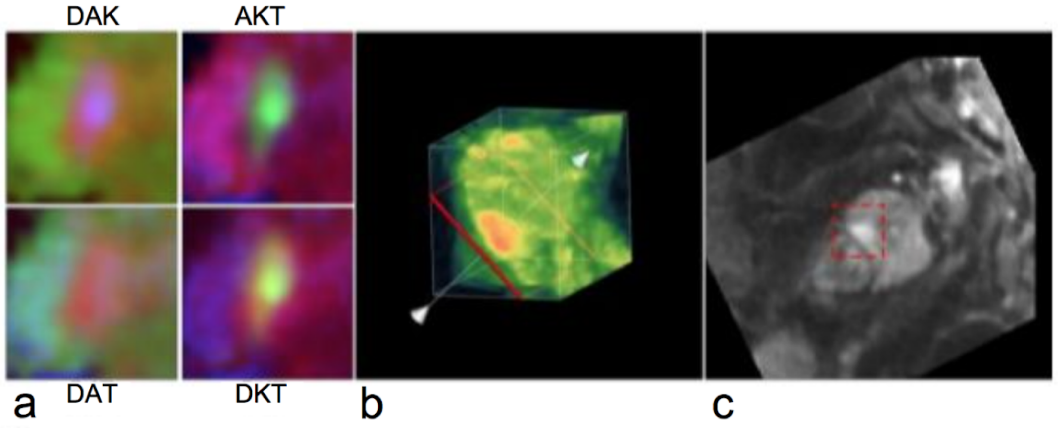

Figure 6: a. Examples of the four types of input images for XmasNet. b. Illustration of data augmentation through 3D slicing. c. Illustration of data augmentation through in-plane rotation. The dashed box is a 32×32 region of interest (ROI) centered at the lesion [Liu et al., 2017].

87 handcrafted features were used to train 140 decision trees using a gradient boosting algorithm called Xgboost [Chen & Guestrin, 2016]. The features included mean, standard deviation and intensities of each lesion along with texture features such as energy and

contrast. Experiments on the validation set proved that XmasNet with a ROC AUC score of 0.95 with sensitivity of 0.89 and specificity of 0.89, outperformed the best XGBoost model with ROC AUC score of 0.89 with sensitivity of 0.87 and specificity of 0.77. On the training set XmasNet achieved a score of 0.83 to become the second best score in the competition.

As mentioned earlier, one of the biggest challenges in using medical science data, especially visual data in deep learning, is its insufficient size. Enormity of data is one of the essentials for most deep learning models. Researchers have been working on exploring new ways to synthetically augment the data. Data augmentation can profoundly impact the learning of deep neural networks. One such approach is called Generative Adversarial Networks (GAN) [Goodfellow et al., 2014]. A study on the same ProstateX dataset uses Generative Adversarial Networks to generate synthetic data samples [Kitchen & Seah, 2017]. Generative models can be used to generate new data that looks very similar to the real data. A generative model's ability to encompass the data distribution itself, as opposed to conditional probability of a certain label given the data, makes it suitable for the task of data augmentation, unlike a discriminative models:

$$P(X|Y) \quad \textbf{Discriminative Model}$$

$$P(X) \quad \textbf{Generative Model}$$

where P(X) is the data distribution and P(Y|X) is the conditional distribution of target variable given the data distribution.

A generative adversarial network can be generalised by using an analogy of a game played between two players with distinct competing objectives. The two players are the generator G and a discriminator D. G is a content generator that tries to create realistic looking content. D is a judge that tries to classify the authenticity of a content as fake or original. The principal equilibrium strategy in this game is for G to draw from P(X) in which case D performs no better than random guessing, i.e. the best way for G to fool D is to create content that are indistinguishable from real content (according to D). Generative model do not really have a global loss function for performance gain, instead these models are trained to an equilibrium point where neither player can improve their performance given a small unilateral change to their strategy; where their strategy is represented by continuous neural network weights. A leap-frog gradient descent algorithm is used for training, where a gradient descent step is taken for G with D held constant, then D with G held constant. With some luck and under conditions, that are in general not well understood, this algorithm can move both players into a suitable equilibrium strategy. This method is particularly powerful if the discriminative models are large Deep Convolutional Neural Networks. If there are any recognizable statistical aberrations in the data generated by G, then D can catch out the generator by recognizing these aberrations. Unrealistic structures are thus suppressed when training has reached equilibrium — G produces highly realistic samples. [Kitchen & Seah, 2017]. The

study uses the same logic to develop generator and discriminator architectures and train them to produce 200 synthetic samples. Samples of real and synthetic lesions are shown in Figure 7.

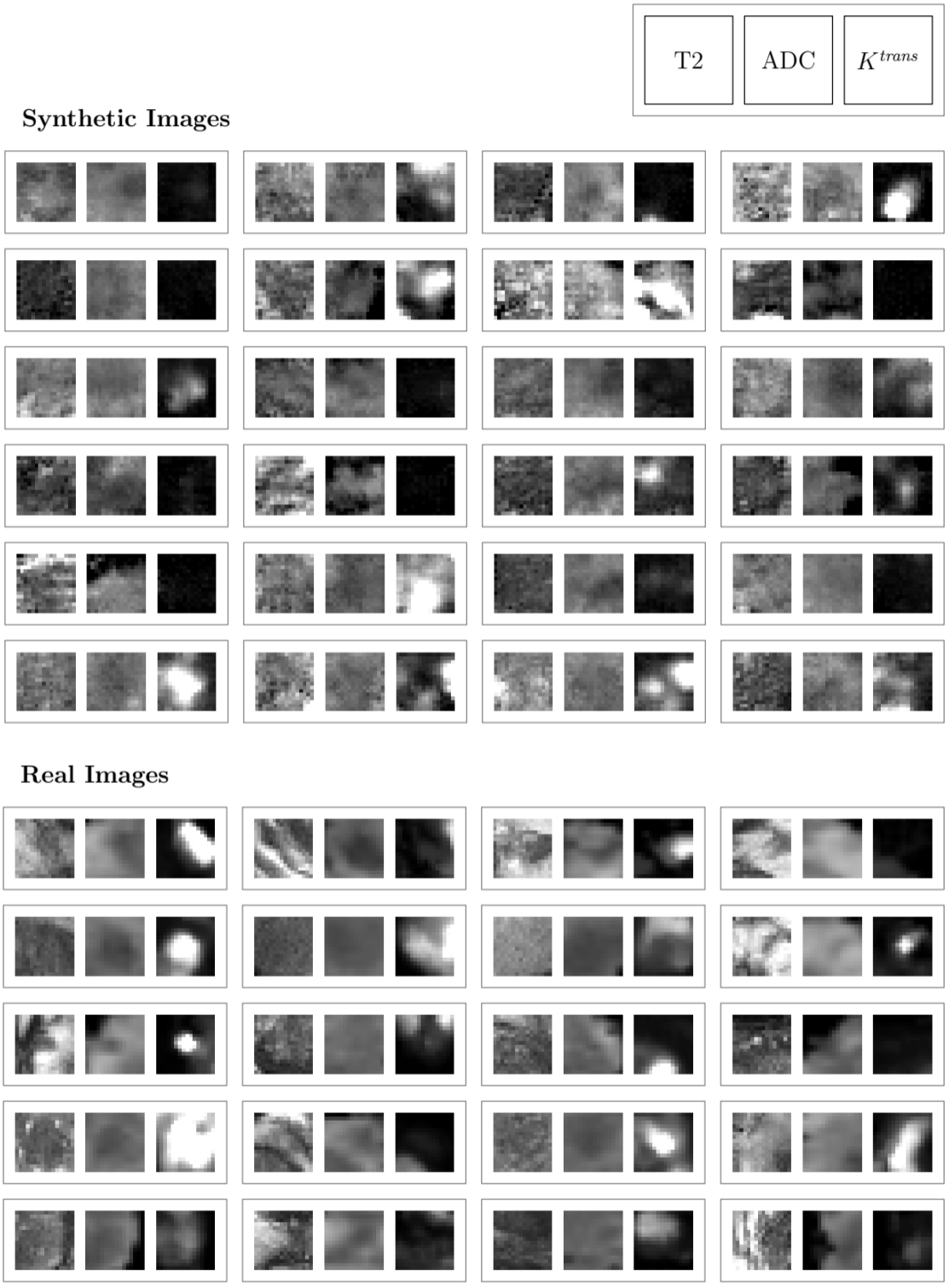

Figure 7: Comparison of real and synthetic ROI patches [Kitchen & Seah, 2017].

Another study which emerged out of the ProstateX challenge, used 3D convolutional neural networks to detect the clinical significance of the Prostate cancer [Mehrtash et al., 2017]. The

network architecture had three different modalities as input: ADC maps, maximum b-value from DWI, Ktrans from dynamic contrast enhanced DCE MRI along with that the zone information was explicitly fed to the network. The network had 9 convolutional layers followed by a dense layer which was connected to the input of zone information. The network achieved an ROC AUC score of 0.80 on the test set.

Prostate segmentation could be an important step in separating the prostate region in an MRI scan for the purpose of clarity. The varying shape and indistinguishable boundaries of the prostate in 3D MRI scans poses a big challenge in automating prostate segmentation. A study proposed 3D segmentation of prostate using a novel approach of volumetric convolutional neural network with mixed residual connections [Yu et al., 2017]. Since the approach uses 3D convolutional networks; therefore, it fully exploits the three dimensional spatial information. Secondly, incorporating the residual connections proved to be very effective in enhancing training efficiency and information propagation which eventually improved the overall performance of the network. The dataset used to train the network was from MICCAI PROMISE12 challenge. The solution stood first in the competition. The study used a complex set of metrics to calculate the final score. The set included, the Dice coefficient [Frakes & Baeza-Yates, 1992], the percentage of the absolute difference between the volumes, the 95% Hausdorff distance [Rote, 1991] and the average over the shortest distance between the boundary points of the volumes.

A study in 2017 used high-level feature representation, extracted through deep neural network, to perform hierarchical classification [Zhu et al., 2017]. In essence, the study tried to draw a comparison between a set of handcrafted features such as LBP and Haar-like features and high-level features for detecting prostate cancer regions. The study is based on data from 21 real patient subjects. The study used stacked autoencoders (SAE) to learn latent high-level feature representation from MRI scan. In SAE model, several single auto encoders (NN) were stacked in a hierarchical way [Hinton & Salakhutdinov, 2006]. As opposed to processing the whole image, the study used the segmentation approach, in which three dimensional image data is disintegrated into smaller parts called voxels. The approach is referred to as voxel-wise training in the study, in which the deep networks learn separately for each voxel as opposed to relying on the whole image. High-level features learned from different modalities were concatenated in the final step. Random forest was chosen for the task of hierarchical classification. Since the study cohort was small, leave-one-subject-out was chosen for cross validation and section-based evaluation (SBE) [Farsad et al., 2005], sensitivity and specificity were used to evaluate the performance of the proposed solutions. The high-level features achieved an averaged section-based evaluation (SBE) of 89.90%, an averaged true positive rate of 91.51%, and an averaged true negative rate of 88.47%, whereas the combination of both high and low-level features yielded an averaged SBE of 91.40%, an averaged true positive rate of 92.32% and an averaged true negative rate of 89.38%. The study concluded that the high-level features outperformed handwrought features and the contextual features helped in improving the hierarchical classification even further.

A study of the breast cancer detection used a novel approach of patch based classification followed by heatmap based post-processing for generating tumor probability to detect metastatic breast cancer [Wang et al., 2016]. The study emerged as one of the solutions from Camelyon16 - ISBI challenge on cancer metastasis detection in lymph nodes ("CAMELYON16", 2018). The data was collected from two different institutes, Radboud University Medical Center (Nijmegen, the Netherlands), and the University Medical Center Utrecht (Utrecht, the Netherlands). Table 1. shows the data distribution.

Institutes	Train (cancer)	Train (benign)	Test
Radboud UMC	90 (22.5%)	70 (17.5%)	80 (20%)
UMC Utrecht	70 (17.5%)	40 (10%)	50 (12.5%)
Total	160 (40%)	110 (27.5%)	130 (32.5%)

Table 1: Camelyon16 data distribution.

Two different evaluation metrics were used in this study: slide-based evaluation (SBE) and lesion-based evaluation (LBE). SBE was measured using ROC-AUC score, based on the probabilities of each test slide containing cancer. LBE was measured using probability of a certain (x,y) pixel for each (predicted) cancer lesion for each whole slide image. The performance was measured as the average true positive rate for detecting all true cancer lesions in a whole-slide image (WSI) across 6 false positive rates: 0.25 , 0.5 , 1, 2, 4, and 8 false positives per WSI. As mentioned above the approach used in this study was two pronged, classification based on patch and headmap (post-processing). Patch-based classification used WSI as input with the label, indicating ROI of each WSI containing metastatic cancer. Training was done using millions of small positive and negative patches, randomly generated from the training data. Patches in the ROI containing tumors were labeled positive otherwise negative. The post-processing stage included tumor probability heatmap generation, the probability heatmap was further used to compute evaluation score based on both slide as well as lesion for each WSI. As opposed to developing a custom deep network from scratch, GoogleNet [Szegedy et al., 2015] was the architecture of choice for training. The solution stood first in both competitions, with an impressive area under the receiver operating curve (AUC) of 0.925 for WSI classification and for tumor localization it achieved a score of 0.7051.

2.2 State of the Deep Neural Network Architectures

Since the problem area comes under the umbrella of image classification; therefore, it is of paramount importance to review the state of the art in image classification. The most prominent image classification solutions based on deep convolutional neural networks emerged as winners of the yearly Large Scale Visual Recognition Challenge (ILSVRC) [Russakovsky et al., 2014]. Three architectures from the competition were chosen for this study.

2.2.1 Inception V1

In 2014, a research team from Google won the ILSVRC14. The codename of their architecture was inception [Szegedy et al., 2015]. One of the key features of the solution was the improved utilization of computing resources. Conventionally, deep neural networks are deep only depth-wise, this solution proposed a way to make the best use of depth as well as width on a constant computational budget. This architecture was inspired by network-in-network approach [Lin et al., 2013] to boost the representational power of the network which is the very core of the inception module. One particular instance of this study is called GoogleNet consisting of layers. GoogleNet is very efficient in terms of memory and power use, for instance, a reflection of its efficiency is the fact that it takes 12 times lesser parameters as compared to Alexnet winner of ["ILSVRC2012", 2018].

At the heart of the GoogleNet is the inception module which is the core of the whole architecture. The inner working of this module is non-trivial. The module applies a 1x1 convolutional layer to already existing convolutional layers followed by linear activation [Krizhevsky et al., 2012]. However, in inception, 1x1 convolution is multipurpose, it is mainly used for the purpose of dimensionality reduction and it allows not only to increase the depth, but also the width of networks, without affecting the performance so much. The inception architecture primarily focuses on dense components to efficiently map local sparse structures in a CNN. The architecture also takes inspiration from a layer by layer approach [Arora et al., 2013] where it clusters units in groups with high correlation statistics in the last layer. Based on already perceived information that image patches in lower layers are highly correlated, these can be covered by 1x1 convolutions. Clusters that are spatially spread-out can be covered by applying bigger convolutions of 3x3 and 5x5, and patches decrease with the increase in region size. Also, it can be observed in the architecture that all the convolutional filters 1x1, 3x3 and 5x5 are concatenated before they are passed to the next layer, as illustrated in Figure 8.

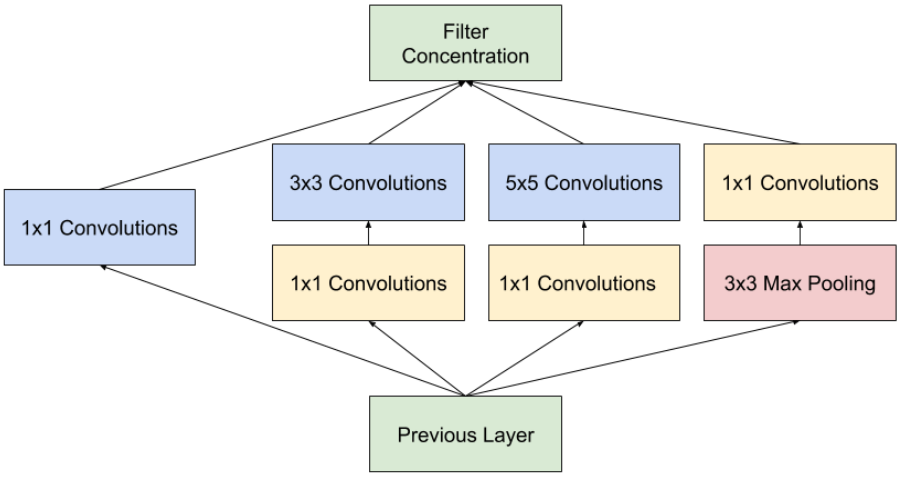

Figure 8: An inception module with dimensionality reduction.

GoogleNet won the 2014 ILSVRC with a 6.67% top-5 error. The most successful version of GoogleNet (an incarnation of inception architecture) is described in Table 2. The first column contains the types of layers or modules used in the architectures. The 'patch size/stride' column describes the filter (patch) size along with stride size (height x width/stride), where applicable. The column 'output size' describes the output shape and size of each layer. The column 'depth' represents the depth of each layer/module. The column '# 1x1' represents the number of 1x1 filters used in a specific layer. Similarly, columns '# 3x3' and '# 5x5' represent the number of 3x3 and 5x5 filters respectively. Columns '# 3x3 reduce' and '# 5x5 reduce' represent 1x1 filters applied before 3x3 and 5x5 filters respectively. The column 'Pool proj' represents the number of 1x1 filter applied after max pooling. Params and ops columns represent the number of parameters and operations per layer respectively.

type	patch size/ stride	output size	depth	# 1x1	# 3x3 reduce	# 3x3	# 5x5 reduce	# 5x5	Pool proj	params	ops
convolution	7 x 7 / 2	112 x 112 x 64	1							2.7K	34M
max pool	3 x 3 / 2	56 x 56 x 64	0								
convolution	3 x 3 / 1	56 x 56 x 192	2		64	192				112K	360M
max pool	3 x 3 / 2	28 x 28 x 192	0								
inception(3a)		28 x 28 x 256	2	64	96	128	16	32	32	159K	128M
inception(3b)		28 x 28 x 480	2	128	128	192	32	96	64	380K	304M
max pool	3 x 3 / 2	14 x 14 x 480	0								
inception(4a)		14 x 14 x 512	2	192	96	208	16	48	64	364K	73M
inception(4b)		14 x 14 x 512	2	160	112	224	24	64	64	437K	88M
inception(4c)		14 x 14 x 512	2	128	128	256	24	64	64	463K	100M
inception(4d)		14 x 14 x 528	2	112	144	288	32	64	64	580K	119M
inception(4e)		14 x 14 x 832	2	256	160	320	32	128	128	840K	170M
max pool	3 x 3 / 2	7 x 7 x 832	0								
inception(5a)		7 x 7 x 832	2	256	160	320	32	128	128	1072K	54M
inception(5b)		7 x 7 x 1024	2	384	192	384	48	128	128	1388K	71M
avg. pool	7 x 7 / 1	1 x 1 x 1024	0								
dropout(40%)		1 x 1 x 1024	0								
linear		1 x 1 x 1000	1							1000K	1M
softmax		1 x 1 x 1000	0								

Table 2: GoogleNet incarnation of Inception architecture [Szegedy et al., 2015].

2.2.3 InceptionV3

In 2016, researchers at Google published an improved version of inception architecture called InceptionV2 [Szegedy et al., 2016]. The idea behind the improvement is to disintegrate the convolutions in the inception module even further. The idea was further refined in revised version InceptionV3. InceptionV3 first suggests 5x5 convolutions to be are replaced by consecutive 3x3 convolutions and then it introduces the idea of factorising the 3x3 convolutions even further into asymmetric convolutions of nx1. Moreover, it introduces the

concept of batch normalised auxiliary classifier of fully connected layers. Figure 3. illustrates the architecture of InceptionV3 network. The study suggests four design principles based on large scale experimentation on a wide variety of deep convolutional neural networks, the principles are as follows:

- ➢ While designing an architecture, avoid too much data compression in the earlier layers. In general the compression of data representation should take place uniformly as we move towards the output.
- ➢ In order to increase the training efficiency of a network, high dimensional data can be processed locally to retrieve segregated features by increasing activation per tile in a convolutional network.
- ➢ Dimensionality reduction, before spatial aggregation, is more efficient. The reason for that is believed to be the high correlation between adjacent units, which results in much less loss of information during dimensionality reduction.
- ➢ Striking a balance between the dimensions of the network, in terms of width and depth. A network can be optimized for performance by balancing the number of filters per stage and its depth.

The improvements in the actual inception architecture were brought about using the same design principles. Variants of InceptionV3 modules are illustrated in Figure 9.

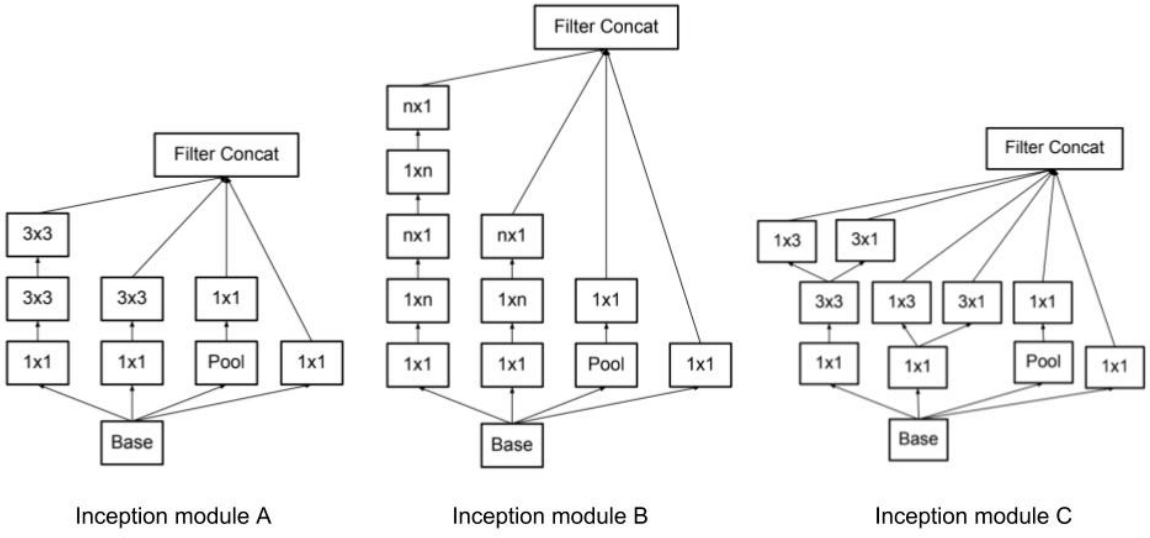

Figure 9: Inception modules [Szegedy et al., 2016].

Inception module A in Figure 9, shows a replacement of the one of the first inception modules where 5x5 convolutions are replaced by two 3x3 convolutions. Inception module B factorizes the nxn convolutions to 1xn and nx1 according to the third design principle. Finally, Inception module C represents the second design principle, locally processing high dimensional data by spatial aggregation (1x1 convolutions).

Table 3. provides an abstracted representation of the architecture of the proposed network. It makes use of the inception modules in Figure 9.

type	patch size/stride or remarks	Input size
conv	3x3/2	299x299x3
conv	3x3/1	149x149x32
conv padded	3x3/1	147x147x32
pool	3x3/2	147x147x64
conv	3x3/1	73x73x64
conv	3x3/2	71x71x80
conv	3x3/1	35x35x192
3 x Inception	Inception module A	35x35x288
5 x Inception	Inception module B	17x17x768
2 x Inception	Inception module C	8x8x1280
pool	8 x 8	8x8x2048
linear	logits	1x1x2048
softmax	classifier	1x1x1000

Table 3: InceptionV3 model [Szegedy et al., 2016].

The column 'type' contains all layers/modules used in the architecture. The column 'patch size/stride or remarks' contains information regarding filter size, step size and if applicable remarks about the corresponding layers. 'Input size' represents the input dimensions of the layer/module: height x width x filter-count.

InceptionV3 was trained and tested on ILSVRC12 data and it achieved 21.2% top-1 and 5.6% top-5 error for single crop evaluation.

2.2.4 Xception

In 2015, Francois Chollet, an artificial intelligence researcher at Google and the developer of the famous neural network python API Keras, proposed yet another deep neural network architecture called Xception. It was inspired by the inception modules. In his study he broke down the interpretation of inception module as in intermediary between normal convolutions and depthwise separable convolution operation, which eventually lead to the ideation of a novel deep convolutional neural network architecture. The depth-wise separable convolutions are at the heart of this architecture. [Chollet, 2016].

The advent of inception-like architecture was a paradigm shift from Visual Geometry Group (VGG) style [Simonyan & Zisserman, 2014] architecture. VGG style networks were primarily stacks of simple convolutional layers, whereas inception-like architectures used inception modules as building blocks. Although, inception modules bear conceptual similarity to convolution, they learn a far richer representation of data with less parameters. The power of inception module lies in the decoupling of spatial and cross-channel correlations. Xception architecture pushed the boundaries even further by introducing depth-wise separable convolutions, which is similar to the idea of extreme inception: to

separately map spatial correlations for each channel. However, they differ from each other in two ways

- ➢ Inception performs 1x1 convolution first, whereas depth-wise separable convolution performs channel-wise spatial convolution first.
- ➢ In inception, both operations are followed by piecewise linearity, rectified linear unit (ReLu), whereas depth-wise convolutions do not strictly rely on piecewise linearity.

An abstract representation of the Xception architecture is described in Figure 10.

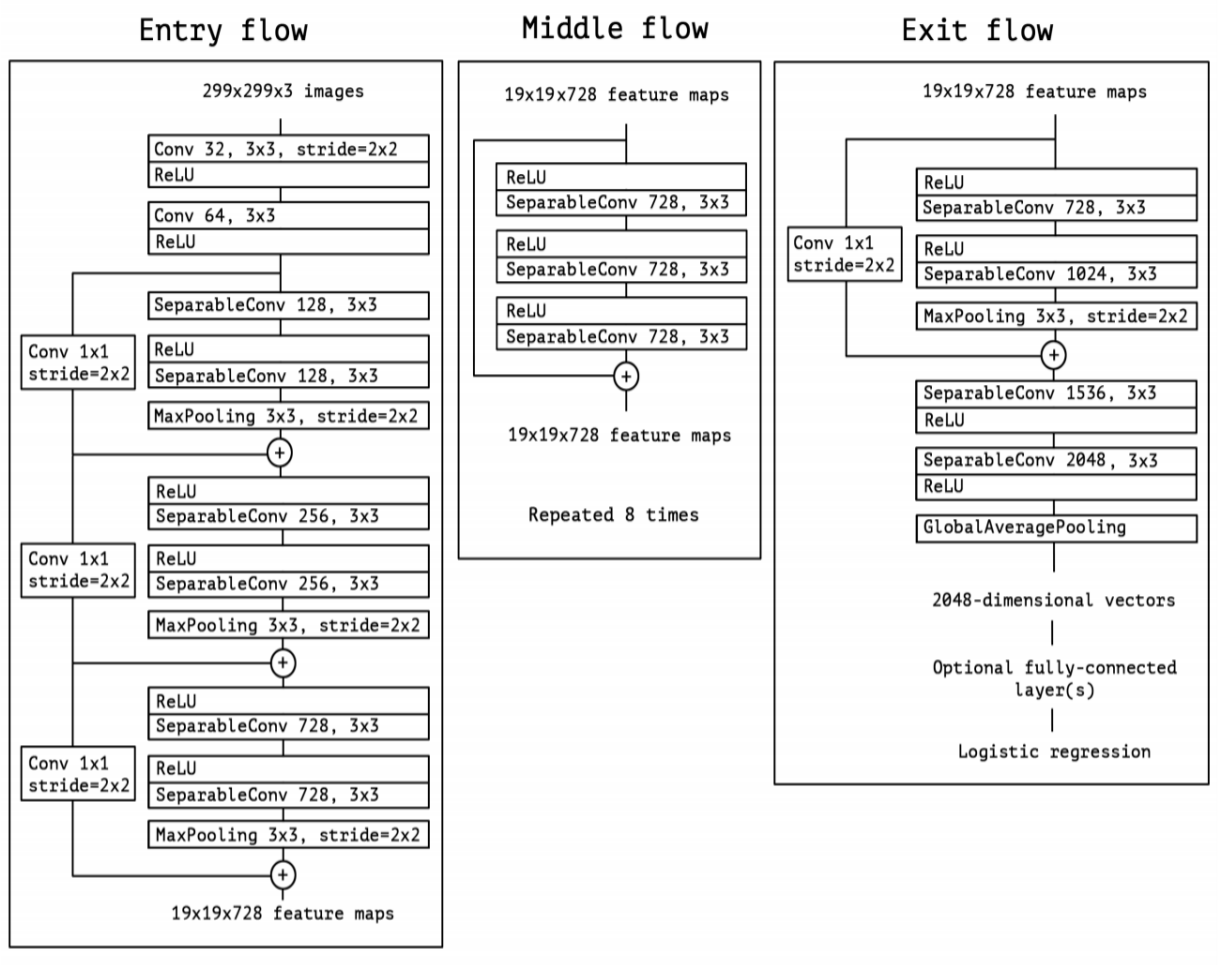

Figure 10: Xception architecture [Chollet, 2016].

Conceptually, Xception architecture is distributed into 3 main parts: entry flow, middle flow and exit flow. The data is first passed through the entry flow, then iteratively processed 8 times through the middle flow, and finally passed through the exit flow. In this process flow, batch normalization is performed on the data after every Convolution and SeparableConvolution layer, which is explicitly shown in Figure 10. Each SeparableConvolution employs a depth multiplier of 1.

Xception architecture was compared with InceptionV3 architecture on a large image classification dataset comprising of 350 million images and 17,000 classes. Since both architectures have the same number of input parameters, the improved performance of Xception is the result of model efficiency. Xception achieved a top-1 accuracy of 0.790 and top-5 accuracy of 0.945.

2.2.5 Deep Residual Learning for Image Recognition

Residual networks brought back very deep convolutional networks back to limelight. In 2015, researchers at Microsoft proposed the idea of deep residual learning for image recognition [He et al., 2015]. The idea was to use residual blocks as the core concept of the network to tackle vanishing and exploding gradients in order to improve the learning process of the network. It became evident from experiments that this approach not only made it easier to optimize the network, but yielded better accuracy with increased depth as well. These networks were significantly deeper than their predecessors in deep networks such as VGG nets [Simonyan & Zisserman, 2014]. One incarnation of residual networks has 152 layers which is 8x the depth of VGG nets.

At the heart of residual network, there is the residual block. In a nutshell, the function of a residual block is to perform an identity mapping through a skip connection. In this residual learning framework, the idea is to explicitly let the layers fit a residual mapping instead of relying on stacked layers to directly fit a desired mapping. Let the desired underlying mapping of the residual be H(x). We set the residual block such that H(x) = F(x) + x where F(x) is linear product of the residual block and x is the identity mapping, H(x) is the combined output of these two. Figure 11 graphically represents a residual block.

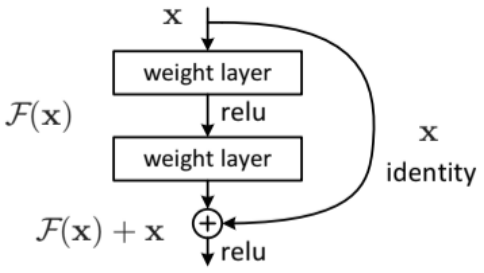

Figure 11: Residual Learning: a building block [He et al., 2015].

A combination of these residual networks achieved 3.57% error on the ImageNet test dataset, securing first position in ILSVRC 2015 classification task. Due to the deep representation of residual networks, the solution gained 28% improvement on the COCO dataset. A 34-layer residual network is illustrated in Figure 12. The architecture starts with a layer containing 64 '7x7' convolution filters and then it is further divided into 4 parts, the first uses layers with 64 '3x3' convolution filters, the second part uses layers with 128 '3x3' convolution filters, the third part uses layers 256 '3x3' convolution filters and the fourth part uses layers with 512 '3x3' filters. The fourth part is followed by a fully connected dense layer with 1000 neurons.

Altogether, the network consists of 34 layers. The residual block in Figure 11 is the building of block of the network.

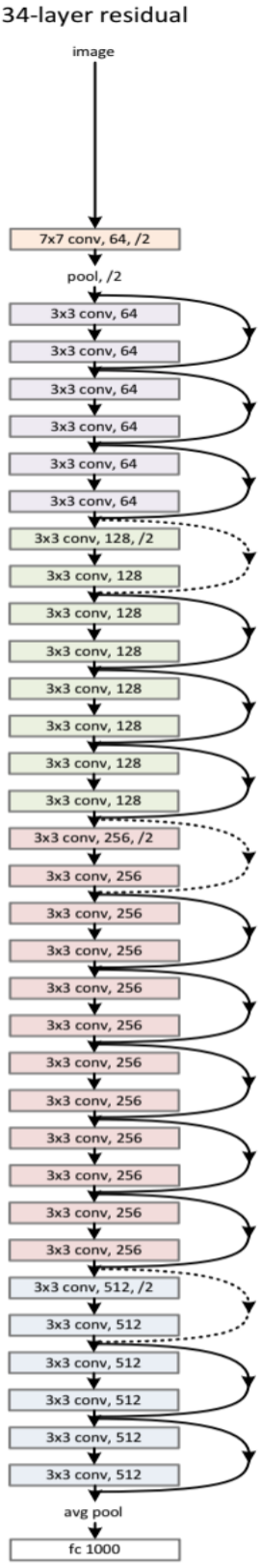

Figure 12: 34-Layer Resnet [He et al., 2015].

3. DATASET AND FRAMEWORKS

3.1 Dataset

The dataset used in this thesis work was made public in a 2017 online competition called ProstateX2 [“SPIE-AAPM-NCI Prostate MR Gleason Grade Group Challenge”, 2018]. The dataset comprised of mpMRI scans of 99 patients. The mpMRI scans had four different modalities:

- ➢ T2-weighted transaxial images (DICOM format).
- ➢ T2-weighted sagittal images (DICOM format).
- ➢ Ktrans images (mhd format)
- ➢ Apparent diffusion coefficient (ADC) images (DICOM format).

These cases contained 112 lesions. Each lesion had a known pathology-defined Gleason Grade Group [Epstein et al., 2016]. The total Gleason score (GS) was calculated based on the appearance of the cells under the microscope, the first half of the score was the representation of most common cell morphology, and the second half was the second most common cell pattern. Both halves score ranged from 1 to 5. Gleason grade groups are illustrated in Figure 13 and are following:

Gleason Grade Group 1 (GS less than 6): Well-formed separable glands.
Gleason Grade Group 2 (GS 7[3+4]): Mainly well-formed glands with sporadic presence of poorly-formed glands as the smaller component.
Gleason Grade Group 3 (GS 7[4+3]): Mainly poorly-formed or fused glands with sporadic presence of well-formed glands as the smaller component.
Gleason Grade Group 4 (GS 8[4+4]; 8[3+5]; 8[5+3]): (1) Only poorly-formed glands or (2) mainly well-formed glands and lacking glands as the smaller component or (3) mainly lacking glands and the smaller component of well-formed glands.
Gleason Grade Group 5 (GS 9 or 10)**:** There is dearth of gland formation in this group including or excluding poorly-formed glands.

Gleason grade group 2 and 3 both had a total score of 7, however, they differed depending on the scores of the most common and second most common cell morphology. [“SPIE-AAPM-NCI Prostate MR Gleason Grade Group Challenge”, 2018].

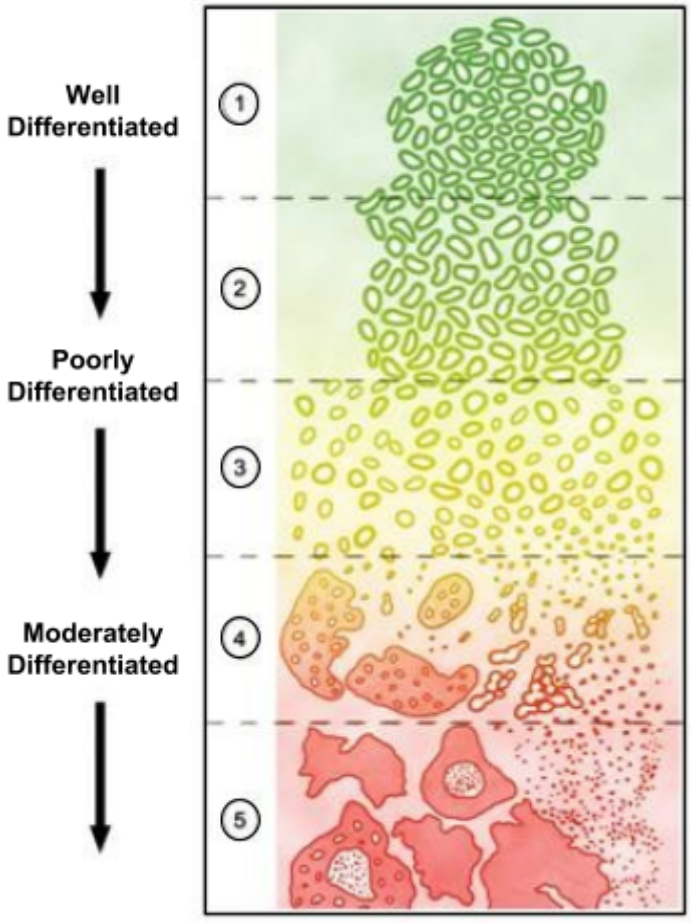

Figure 13: Gleason grade groups ["How is Prostate Cancer Diagnosed?", 2018].

3.1.1 Exploratory Analysis

Exploratory analysis of data included basic visualisation techniques to understand the scans better. These visualisations included, region of interest detection, analysis of zone information distribution and Gleason grade group distribution. Exploratory analysis helped in detecting class imbalance, random order of prostate slices (which was fixed by reading the metadata from DICOM headers), lack of standardization of image resolution across different cases as well as modalities. Figure 14 shows region of interest highlighted with a red bounding box in four different image artifacts.

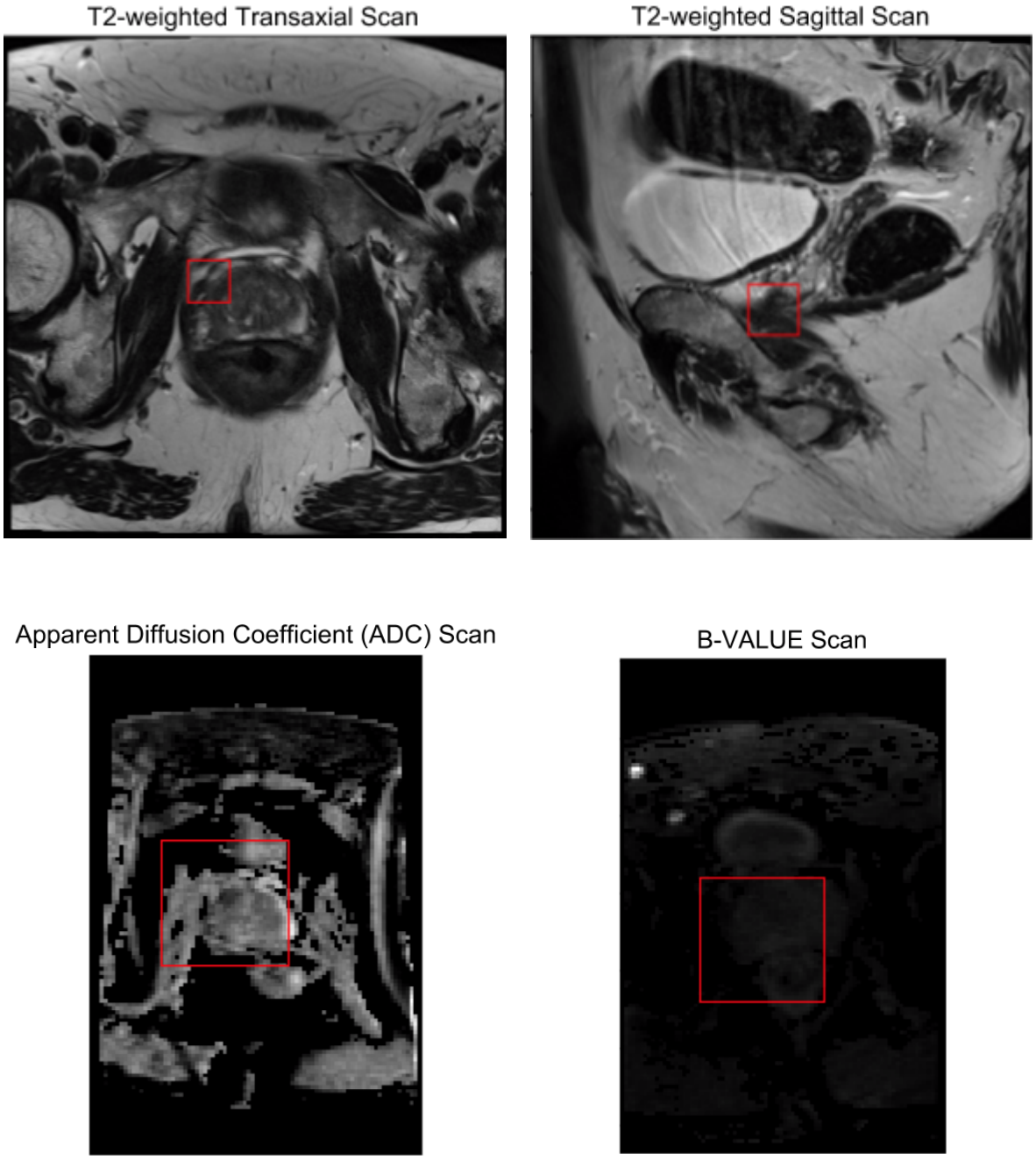

Figure 14: T2- weighted scan (sagittal and transaxial plane), apparent diffusion coefficient and b-value scan with ROI highlighted with a red bounding box.

Figures 15 & 16 show the distribution of zone information and Gleason grade group, respectively.

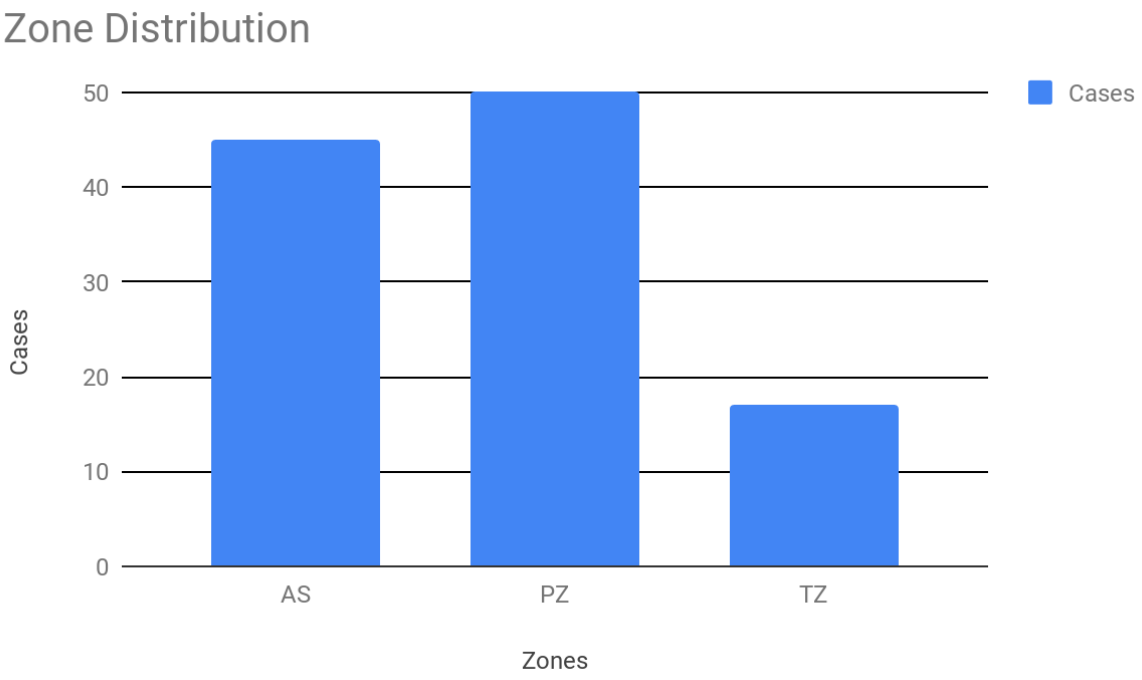

Figure 15: Bar chart showing zone information distribution, AS: anterior fibromuscular stroma, PZ: peripheral zone, TZ: transition zone.

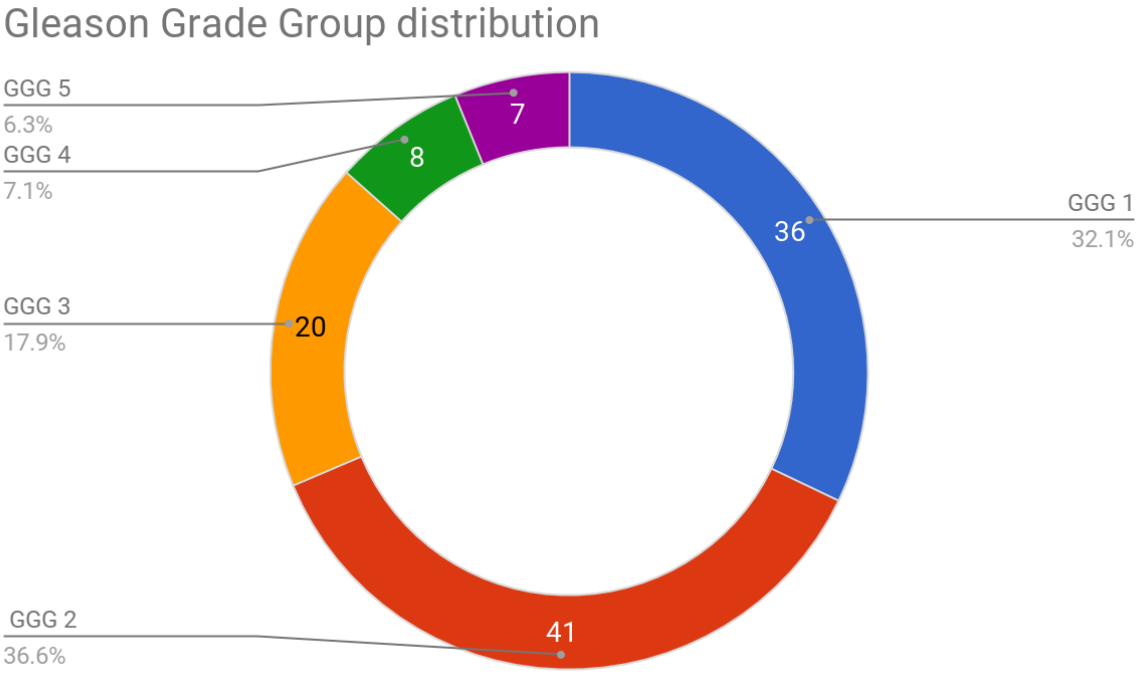

Figure 16: Pie chart showing Gleason grade groups distribution.

3.2 Technology and Frameworks

Python 3.5.3 was the programming language chosen for running all the experiments. There is a big collection of libraries in Python for different purposes. The experiments involved a number of different libraries, however, the major ones are mentioned in this section:

- *Numpy (version 1.14.0)* is used for different mathematical, in particular matrices operations.
- *Scikit-Learn (version 0.19.1)* implements a variety of different conventional machine learning algorithms such as logistic regression, linear discriminant analysis and k-nearest neighbors classification.
- *Tensorflow (version 1.4.1)* is a powerful artificial neural network library developed by Google, it is primarily used for deep learning.
- *Keras (version 2.1.3)* is a wrapper library for implementing big neural network models, it is easy to understand and manage. It uses Tensorflow as a low level library for building neural networks.
- *Pandas (version 0.22.0)* is a very efficient data processing library.
- *ImgAug (version 0.2.5)* is an image augmentation library which is used to extend small image datasets using different image processing techniques, such as rotation, inversion, and by adding Gaussian noise.
- *PyDicom (version 0.0.9)* is a library for reading DICOM files and their metadata.
- *OpenCV (version 3.4.0.12)* is one of the most popular computer vision or image processing library which is common across a number of different programming languages.
- *Matplotlib (version2.1.1)* is visualisation library, generally used to draw plots.

3.3 Data Wrangling

3.3.1 Preprocessing

All four modalities required preprocessing, before they were used for any feature extraction or training process. Preprocessing was required primarily because of lack of image resolution standardization, modalities sharing multiple regions of interest with different Gleason grade group and DICOM sequence irregularities.

Padding and ROI patch alignment: A 32x32 patch was extracted from every image and was centered to the ROI. Experiments which utilized extended augmentation used bigger patches of size 48x48 with additional padding to ensure data integrity while rotations.

DICOM sequencing: The naming convention used to name slices in DICOM sequences did not match the original order of the images as per the metadata file. DICOM headers were read to put slices in order.

Normalization: DICOM images were grayscale images and were required to be normalized to make them compliant with the image processing and machine learning algorithms. Each image pixel was divided by 255 to normalize the value.

3.3.2 Image Augmentation

As mentioned in the previous chapters, one of the biggest problems with healthcare solutions is the scarcity of data. ProstateX2 dataset was also riddled with similar problems, small size, lack of standardization and lack of details in metadata. However, in order to tackle the small sized dataset problem, standard image processing techniques were employed to augment the dataset. Python library imgaug was used [“Github - Imgaug”, 2018] for two different types of augmentation:

Standard Augmentation: Standard augmentation used 5 variants of the image using standard image processing techniques: flip (left-right), flip (up-down), dropout , invert and orthogonal rotation. Figure 17 illustrates different augmentation techniques including the ones employed in this project.

Extended Augmentation: Extended augmentation generated 72 variants of the image by rotating it 360˚ by increment of 5˚. There was also an option of pre-augment which applies standard augmentation before applying extended augmentation. There was definitely some redundant augmentation; however, optimization was not a concern.

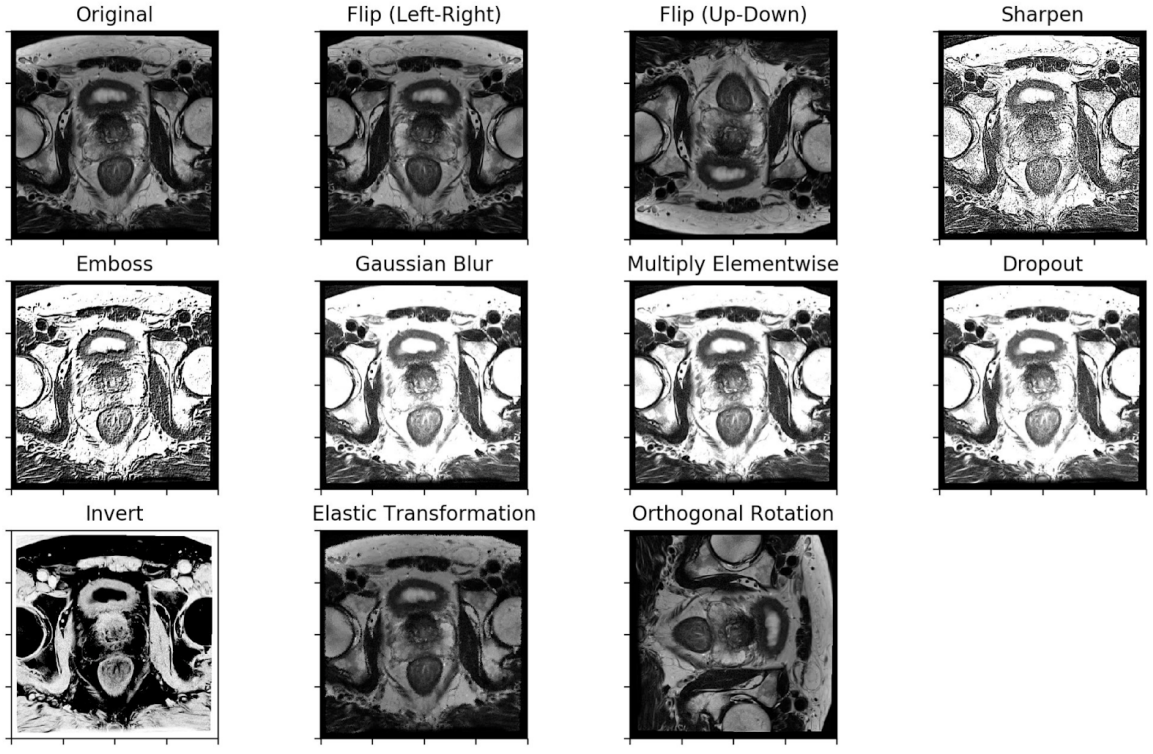

Figure 17: Image augmentation techniques.

3.4 Evaluation Metrics

3.4.1 Accuracy Score

Accuracy score is a standard metric to compute the fraction of correctly predicted labels. If

$\hat{y}_k$ is the predicted label of the k-th sample and y_k is the corresponding true label, then the fraction of correct predictions over total number of samples is defined as:

$$accuracy(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{k=0}^{n_{samples}-1} 1(\hat{y} = y_k)$$

[“Model evaluation: quantifying the quality of predictions”, 2018].

3.4.2 Cohen’s Kappa

The evaluation metric Cohen’s Kappa is used to compute inter-rater agreement score, not necessarily between a classifier and the ground truth. Cohen’s Kappa score varies on a spectrum between -1 and 1, any score above 0.8 is generally considered a strong agreement, anything below zero implies random labeling. Kappa score can be used for both binary as well as multiclass classification problems, however, it cannot be used for multilabel problems. [“Model evaluation: quantifying the quality of predictions”, 2018]. Cohen’s Kappa can be mathematically expressed with the following equation :

$$\kappa = (p_o - p_e) / (1 - p_e)$$

where p_o is the empirical probability of agreement on the label assigned to any sample (the observed agreement ratio), and p_e is the expected agreement when both annotators assign labels randomly. p_e is estimated using a per-annotator empirical prior over the class labels [Cohen, 1960].

4. PROPOSED METHODS

This thesis project was an effort to draw a comparison between classification prowess of conventional machine learning models and deep learning models on a small dataset of 112 lesions. This chapter sheds light on image processing algorithms used for feature engineering, conventional machine learning algorithms that used those features and deep convolutional neural network models.

4.1 Feature engineering

Two sets of features were used in training classification models: histogram of oriented gradients (HOG) [Dalal & Triggs, 2005] and local binary pattern (LBP) [Ojala et al., 1996].

4.1.1 Histogram of Oriented Gradient (HOG)

Histogram of Oriented Gradient descriptor uses the distribution of edge directions and intensity gradients to highlight the appearance and shape of local object within an image. The image is divided into smaller contiguous segments called cells, following that a histogram of gradient direction is computed for the set of pixels in each cell. The descriptor is generated by concatenating the previously computed histograms. In order to improve accuracy, contrast normalisation is carried out of local histograms by computing intensity measure across larger region of the image, called a block. Then all cells are normalized using the calculated value within that block. The normalisation makes the image prone to changes in brightness.

There are some benefits of using HOG descriptors as compared to other descriptors. Since it operates on local level, it is not susceptible to geometric and photometric transformations, excluding the object orientation. Changes like those only appear in large spatial regions. [Dalal & Triggs, 2005].

4.1.2 Local Binary Pattern (LBP)

Local Binary Pattern is a visual descriptor primarily used in image processing. It is a very efficient image feature used for texture classification. It performs pixel labeling by fixing neighborhood radius of each pixel and considers the result as a binary output. LBP is computationally inexpensive and has strong discriminative ability, which makes it a very widely used technique in a broad range of applications.

The fundamental idea behind LBP operator was to introduce a way to describe the two-dimensional surface by two additional measures: local spatial patterns and grayscale contrast. The LBP operator forms labels for the image pixels by thresholding the 3 x 3

neighborhood of each pixel with the center value and considering the result as a binary outcome. [Ojala et al., 1996].

Figure 18a shows the visual output of histogram of oriented gradients, for a T2-weighted transaxial scan. Figure 18b shows visual output of local binary pattern, for a T2-weighted transaxial scan.

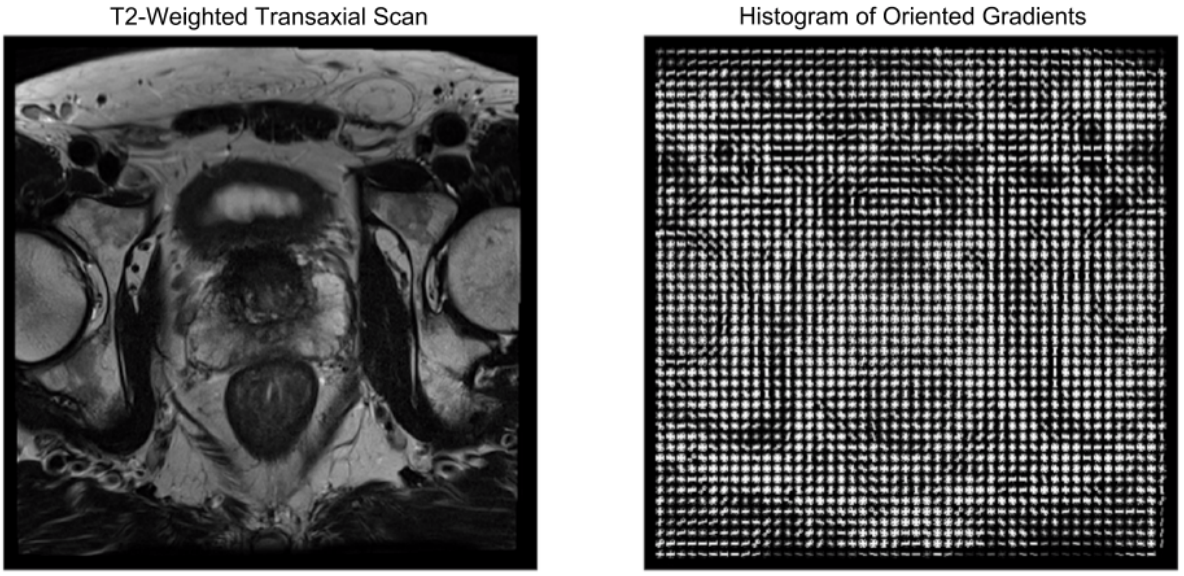

Figure 18a: Histogram of oriented gradients calculated from T2-weighted transaxial scan.

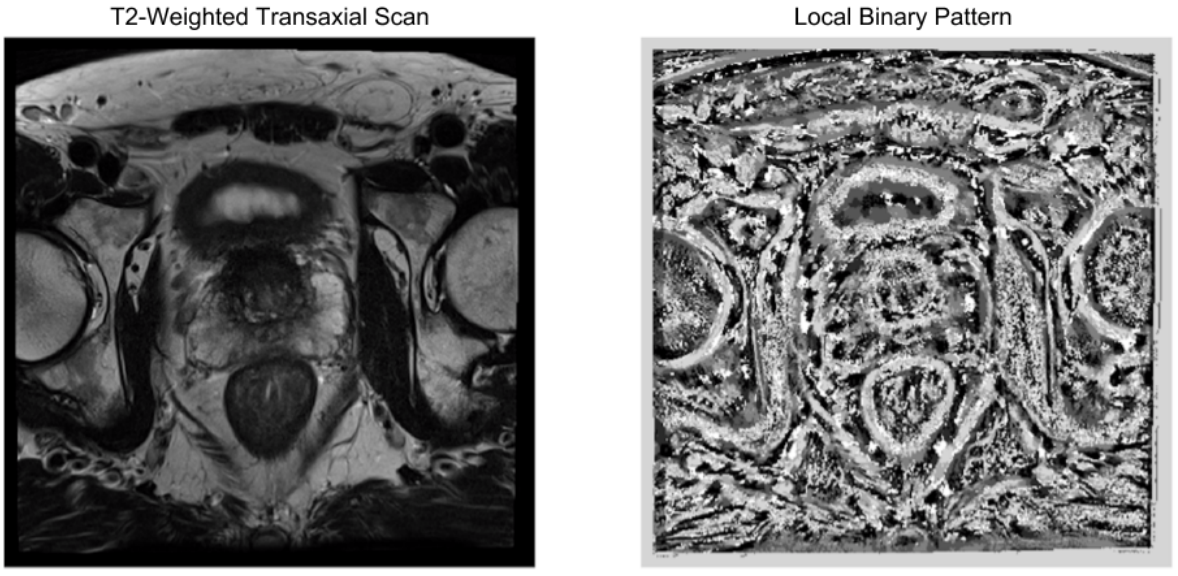

Figure 18b: Local binary pattern calculated from T2-Weighted transaxial scan.

4.2 Conventional Machine Learning

The idea of this research was to organically start from simpler techniques to attain a benchmark accuracy and then to move towards more complex techniques. Following the same line of logic, eight conventional machine learning classifiers were used. The classifiers are as follows.

4.2.1 Bernoulli Naive Bayes

Bernoulli Naive Bayes is type of supervised learning algorithm which uses Naive Bayes [Zhang, 2004] training as the building block of the algorithm. Naive Bayes predicts the likelihood of the occurrence of an event based on the prior information or evidence present in the data. Bernoulli Naive Bayes is used for the classification of multivariate data and the variables are interpreted as binary valued variables. Bernoulli Naive Bayes is based on the following equation:

$$P(x_i \mid y) = P(i \mid y)x_i + (1 - P(i \mid y))(1 - x_i)$$

["Bernoulli Naive Bayes", 2018]

It can be deduced from the equation above that the classifier penalizes the absence of a feature that is highly correlated to a certain class. ["Bernoulli Naive Bayes", 2018]. One of the applications of Bernoulli Naive Bayes is text classification. The classifier can use vectors representing word occurrences as opposed to word count for training. [McCallum & Nigam, 1998]

4.2.2 Passive Aggressive Algorithm

Passive Aggressive algorithm is a type of online learning algorithms [Smale & Yao, 2006]. The idea of the passive aggressive algorithms are quite simple and similar to perceptron algorithm [Block, 1962]. Both algorithms don't necessitate a learning rate. However, unlike perceptron, passive aggressive algorithms use regularization parameter ["Passive Aggressive Algorithms", 2018].

The working of a Passive Aggressive algorithm can be explained with the help of an example. Consider a dataset:

$$\begin{cases} X = \{\bar{x}_0, \bar{x}_1, \dots, \bar{x}_t, \dots\} \; where \; \bar{x}_i \in \mathbb{R}^n \\ Y = \{y_0, y_1, \dots, y_t, \dots\} \; where \; y_i \in \{-1, +1\} \end{cases}$$

In the equation above, t signifies the temporal dimension. However, in this case, the data points can be generated perpetually. In theory, if they are extracted from the same distribution, the algorithm will continue to learn, if the distribution changes, the weights will

slowly forget the previous one and be tuned according to new one. Let's also assume a binary classification process for the sake of simplicity.

Let the *w* be the weight vector, then the prediction will be:

$$\tilde{y}_t = sign(\bar{w}^T \cdot \bar{x}_t)$$

Hinge loss function is one of the foundational elements of this algorithm:

$$L(\bar{\theta}) = max\big(0,1 - y \cdot f(\bar{x}_t; \bar{\theta})\big)$$

The loss L is between 0 (representing complete match) and K depending on the function $f(x(t),\theta)$ with $K>0$ (representing completely invalid prediction). A generic Passive-Aggressive framework is based on the following update rule:

$$\begin{cases} \bar{w}_{t+1} = argmin_{\bar{w}} \dfrac{1}{2} \|\bar{w} - \bar{w}_t\|^2 + C\xi^2 \\ L(\bar{w}; \bar{x}_t, y_t) \leq \xi \end{cases}$$

ξ represents the slack variable which is an important component of this framework. Let's assume that $\xi=0$ and L is fixed to be 0. Given a sample $x(t)$, the sign is determined by current weight vector. Loss function outputs 0 when the sign is correct and the argmin becomes $w(t)$. This shows that the algorithm is passive on the occurrence of a correct classification event. In the case of misclassification, the update rule turns into an aggressive penalizing rule, because it recalibrates *w* to be close to the previous one as possible (otherwise the existing knowledge is immediately lost), but it must satisfy $L=0$ (in other words, the classification must be correct).

The slack variable allows to have margins as areas and regularization controlled by the parameter *C*. Aggressiveness of of the algorithm is controlled by the parameter *C*, higher values of *C* result in more aggressiveness which makes the algorithm prone to inconsistency in case of noisy data. And lower values of *C* help algorithm adapt better. Online algorithms like Passive Aggressive should be robust enough to handle noisy data (wrongly annotated). After solving both update conditions, we get a closed-form update rule:

$$\bar{w}_{t+1} = \bar{w}_t + \frac{\max\left(0,1 - y_t(\bar{w}^T \cdot \bar{x}_t)\right)}{\|x_t\|^2 + \dfrac{1}{2C}} y_t \bar{x}_t$$

["ML Algorithms addendum: Passive Aggressive Algorithms", 2018; Crammer et al., 2006]

4.2.3 K-Nearest Neighbor Classification

K-nearest neighbors (KNN) classification is a type supervised learning method. In pattern recognition, KNN algorithm is a method for classifying objects based on the closest training examples in the feature space. It is a type of instance based learning or lazy learning where the function is approximated locally and all computation is delayed until classification. KNN is one of the most fundamental classification techniques when there is little or no prior knowledge about the distribution of data.

The K in KNN is the number of nearest neighbors that the classifier will consider to make its prediction. Standard Euclidean distance is typically used to measure the distance between neighbors, however, KNN accepts a number of different distance measures. In order to classify a sample, voting is done based on the classes of the k nearest neighbors of the sample and the majority class is assigned to the sample. There are also some disadvantages of using KNN, it is highly susceptible to bias learned from local structure of the dataset. Moreover, majority voting doesn't always yield the desired results, especially when the class distribution is skewed. For instance, if the class distribution is skewed, the majority class will affect predictions of new samples. [Coomans & Massart, 1982]

KNN has been successfully used in a number of different classification and regression problems, despite is simplistic nature. KNN is very useful in scenarios where the decision boundaries lack regularity.

4.2.4 Random Forest

Random forest algorithm is a type of supervised learning method. It is a powerful algorithm which is used for both, classification as well as regression. The algorithm is loosely based on the concept of generating a forest using a number of decision trees. In general, the more trees in the forest, the more robust the prediction and higher the accuracy.

The algorithm was first designed by an IBM engineer, Tin Kam Ho. The idea was based on decision trees because of their time efficient execution. However, lack of generalization ability on unseen data due to high complexity and suboptimal accuracy on training data due to lack of complexity was the biggest challenges of decision trees. The solution, therefore, was based on a stochastic modeling principle, which proposed to build multitude of trees in randomly selected subspaces of original feature space. The classification prowess of the classifiers is strengthened when the features subspace coverage is increased. [Ho, 1995]. Random forest is sometimes referred to as ensemble method because the forest is an ensemble of trees generated from samples drawn with replacement from the training data.

4.2.5 Support Vector Machine

Support vector machine (SVM) is a type of supervised learning method, capable of performing both classification as well as regression. The idea is based on the non-linear mapping of vectorized input data to very high-dimensional feature space. Following that a linear decision hyperplane is generated in the features space, nearest to the extreme samples in the dataset. The decision hyperplane is also called the decision boundary. The decision boundary is a frontier that segregates two classes. Then support vectors from both classes are computed based on the maximum margin between the vectors and the decision boundary. The math behind SVM classification can be generalised with the following equations:

$$min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i$$

$$\text{Subject to } y_i(w^T \varphi(x_i) + b) \geq 1 - \zeta_i$$

$$\zeta_i \geq 0, \; i = 1, ..., n$$

Its dual can be shown as:

$$min_{\alpha} \frac{1}{2} \alpha^T Q \alpha - e^T \alpha$$

$$\text{Subject to } y^T \alpha = 0$$

$$0 \leq \alpha_i \leq C, \; i = 1, ..., n$$

where $x_i \in R^p$, $i=1,..., n$, in two classes, $y \in \{1, -1\}^n$ represent training vectors, e represents a vector consisting of ones, $C>0$ represents the upper bound, Q represents an nxn positive

semidefinite matrix, $Q_{ij} \equiv y_i \, y_j \, K(x_i, x_j)$, where $K(x_i, x_j) = \phi(x_i) T \phi(x_j)$ represents the kernel. In this case, using function ϕ the training vectors are mapped into a space with higher dimensions. The decision function is represented by the following equation:

$$sgn(\sum_{i=1}^{n} y_i \alpha_i K(x_i, x) + \varrho)$$

[“Support Vector Machines”, 2018; Cortes & Vapnik, 1995]

Support vector machines are very effective for problems involving high dimensional spaces, even if the number of features exceed the number of samples. It doesn’t use the whole

training data for computing the decision boundary, only a subset of it. [“Support Vector Machines”, 2018]

4.2.6 Logistic Regression

Logistic regression is a type of supervised learning method. It is another machine learning algorithm borrowed from the field of statistics. It is one of the most commonly used method for the task of classification, it is similar to linear regression in terms of estimating the values of parameter coefficients yielding the function that best describes the relationship between known input and output values. However, unlike linear regression the prediction of the output is transformed using a nonlinear function called the logistic function, also known as the sigmoid function. The logistic function was developed by a statistician to describe the properties of the population growth in an ecology rising quickly and maxing out at the carrying capacity of the environment [Cox, 2018]. The output of the sigmoid function looks like an S-shaped curve, it can take any real valued number and map it at any value between 0 and 1, but never exactly at those limits. The mathematical equation of the logistic function is as follows:

$$f(x) = \frac{L}{1+e^{-k(x-x_0)}}$$

where e represents the Euler's number, x_0 is the x value of the sigmoid midpoint, L represents the curve's maximum value and k represents the steepness of the curve [Cox, 2018].

4.2.7 Linear and Quadratic Discriminant Analysis

Linear and quadratic discriminant analysis are supervised learning methods. The methods were proposed by Ronald A. Fisher in 1936. It is evident from the name that the classifiers use linear and quadratic decision surfaces for the task of classification. Both LDA and QDA are commonly found interesting because of their closed form solution and ability to perform supervised dimensionality reduction.

Linear discriminant analysis works on the principle of projecting the dataset onto a lower dimensional space with good class separability in order to avoid over-fitting, curse of dimensionality and computational cost. The goals of projection in LDA is to increase inter-class separability and decrease intra-class variance. [Welling, 2005]. Both LDA and QDA can be derived using probabilistic models, Bayes’ rule can be followed to model the conditional probability of data $P(X|y=k)$, where k represents class.

$$P(y=k|X) = \frac{P(X|y=k)P(y=k)}{P(X)} = \frac{P(X|y=k)P(y=k)}{\Sigma_l P(X|y=l).P(y=l)}$$

the class k is chosen based on maximum conditional probability. And for quadratic discriminant analysis the class conditional distribution is modeled as a multivariate normal distribution with density:

$$P(X|y=k) = \frac{1}{(2\pi)^{d/2}|\Sigma_k|^{1/2}} \exp(-\tfrac{1}{2}(X-\mu_k)^t \Sigma_k^{-1}(X-\mu_k))$$

where d represents the number of features.

The model can be used as a classifier by estimating the class priors $P(y=k)$. The class means and covariance matrices from the training dataset. For LDA, the Normals for each class share the same covariance matrix, which results in linear decision surfaces best understood by comparing the log-probability ratios:

$$\log[P(y=k|X)/P(y=l|X)] = \log[(P(X|y=k).P(y=k))/(P(X|y=l).P(y=l))] = 0 \iff$$

$$(\mu_k - \mu_l)^t \Sigma^{-1} X = \tfrac{1}{2}(\mu_k^t \Sigma^{-1} \mu_k - \mu_l^t \Sigma^{-1} \mu_l)^t - \log[P(y=k)/P(y=l)]$$

For QDA, there are no assumptions on the covariance matrices. [“Linear and Quadratic Discriminant Analysis” 2018; Friedman et al., 2001].

4.2.8 Xgboost

Xgboost is a type of gradient boosting algorithm [Friedman, 2002], which falls under the category of supervised learning. Xgboost is an extreme implementation of gradient boosting machines, which parallelizes the traditional sequential tree generation process. The underlying idea was to develop an algorithm which was both memory and processing efficient. Xgboost can be best understood by it’s features:

1. **Split finding algorithm:** As the data size increases, it becomes more and more difficult to manage the entire dataset in the memory. Xgboost uses an approximate algorithm to find the best split over a continuous feature. It uses feature distribution based percentiles to propose candidate split points. Then the candidate split points are used to bin the continuous features. And finally using aggregated stats on the buckets, the best solution is chosen for candidate split points.
2. **Block structure:** Data sorting is the most time intensive task of tree based learning. Xgboost uses 'blocks', the in-memory units, for data storage. Data columns are stored in their respective blocks, based on the corresponding feature value. This is a one-off computation, required before training. Block sorting can be parallelised using CPU threads. Similarly, split finding can also be parallelised.

3. **Weighted quantile sketch for approximate tree learning:** Xgboost uses weighted quantile sketch to propose candidate split points among weighted datasets. Quantile summaries are then merged and pruned over the data.
4. **Sparsity-aware algorithm:** Xgboost is a sparsity-aware algorithm which is capable of handling missing data.
5. **Cache-aware access:** It chooses 2^16 examples per block to ensure parallelization and prevent cache miss during split finding.
6. **Out of core computation:** It divides the data into multiple blocks and stores it on the secondary memory, if the data doesn't fit into the memory. Blocks are compressed with respect to columns and while reading the data from secondary memory, it is decompressed on the fly.
7. **Regularised learning objective:** Xgboost uses the following regularization term to penalize the model complexity:

   $$Obj(\Theta) = L(\theta) + \Omega(\Theta)$$

   where L is the loss function and Ω represents the regularization term which most algorithms tend to neglect in the formulation of objective function. Whereas, Xgboost controls complexity and prevents overfitting by explicitly using regularization.

Xgboost uses all seven features mentioned above to provide scalability and efficient resource utilization. [“XGBoost: A Concise Technical Overview”, 2018; Chen & Guestrin, 2016].

4.3 Convolutional Neural Networks

Convolutional neural networks are one of most efficient and widely used neural networks in computer vision applications. The idea of convolutional neural network was inspired by the processing of image in an actual brain. The part of brain responsible for processing images is called the visual cortex. Neurons in this specific region of the brain respond to specific stimuli in the receptive field. A complete visual field is cumulative overlap of different neuron fields. This translates into activation of a specific set of neurons when a specific feature is in the visual field like a curve. Together a set of features fire up a set of different neurons (rest being inactive) which then shape complete object in our visual field. [Hubel & Wiesel, 1962].

Humans accomplish the task of image recognition by breaking down image into smaller components or features and recognise those features firsts, and human mind is incredibly fast at this. For instance in order to recognise a car, we recognise it’s features first such as tyres, windscreen, doors, lights, indicators. Since it happens instantaneously and at subconscious level, therefore, the process is not always very noticable. Convolutional neural networks work in a similar fashion, however, they start from learning very granular (or abstract) features such as curves, edges, etc.

Convolutional neural networks usually comprise of several different types of layers, the layers are listed below:

- **Convolutional Layer** - Used to learn feature map.
- **Non-Linearity Layer** - Introduces non-linearity to the network.
- **Pooling Layer** - Reduces the input size and curbs over-fitting.
- **Flattening Layer** - Flattens the multi-dimensional intermediate data.
- **Fully-connected Layer** - Typical neural network .

["CS231n Convolutional Neural Networks for Visual Recognition", 2018]

4.3.1 Vanilla Model and Variations

Vanilla model is a convolutional neural network with several convolutional layers interleaved with rectified linear unit (ReLU) activation, followed by a couple of dense layer, followed by a softmax activation layer for classification. There are four different variations of vanilla network to test the effect of dense layers and batch normalization on classification accuracy. Following is a brief description of all four versions:

- Baseline version: contains a set of 4 convolutional layers followed by ReLU activation and 2D maxpool.
- Version 0: extends baseline version with 2 dense layers.
- Version 0.1: extends baseline version with 2 dense layers followed by batch normalization.
- Version 0.2: extends baseline version with 3 dense layers followed by batch normalization.

Each variation ends with a softmax layer for classification. The reason it is called Vanilla Network is that there is no specific reason behind the arrangement or order of these layers, it is intuitive and experimental in nature. Version 0, 0.1 and 0.2 were built on top of the Baseline version. Figure 19 shows the visual representation of the Vanilla model and its variations.

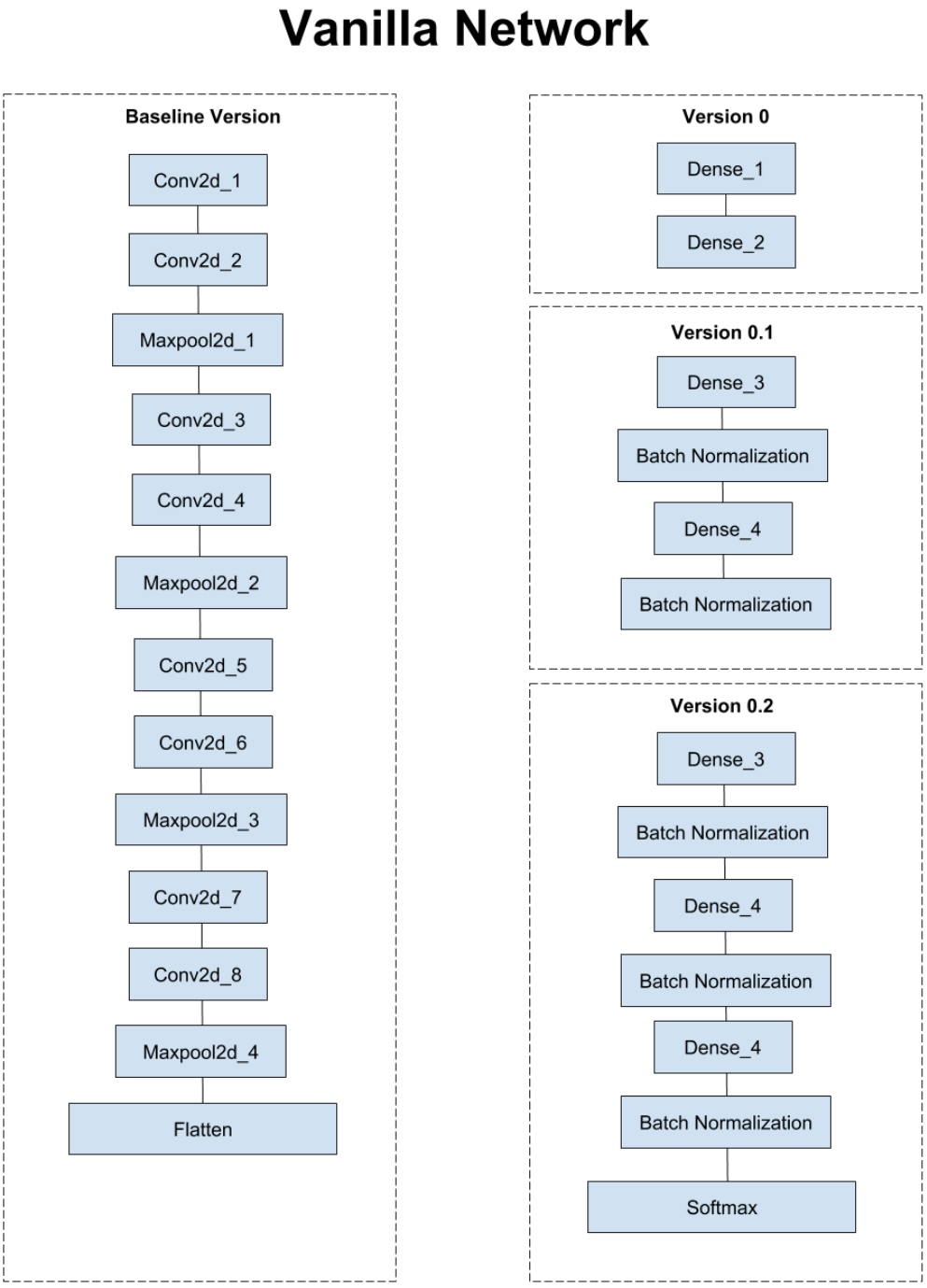

Figure 19: Vanilla model network diagram.

4.3.2 Xmasnet Model

Xmasnet model is the implementation of [Liu et al., 2017], mentioned earlier in Literature Review chapter. Although the model was primarily used on a different dataset for binary classification, due to the similarity of the datasets, it was decided to use the model for multi-class classification. Xmasnet architecture is illustrated in Figure 20.

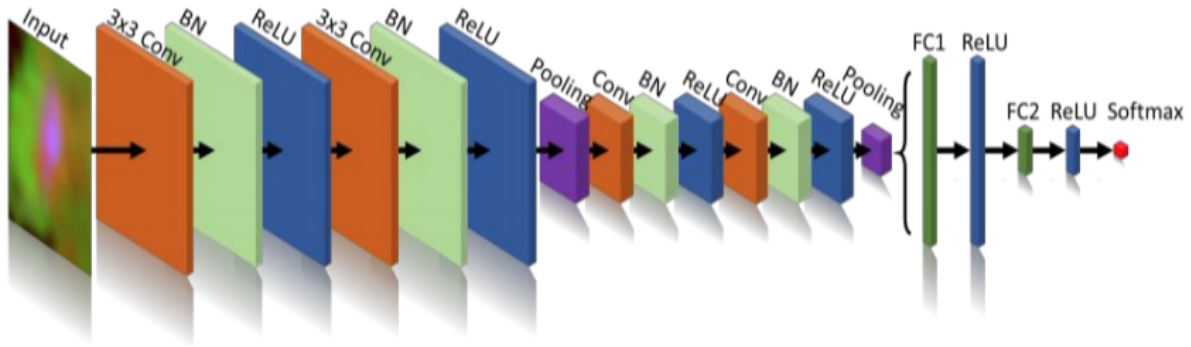

Figure 20: The architecture of the XmasNet. Conv: convolutional layer; BN: batch normalization layer; ReLU: rectified linear unit; Pooling: max pooling layer; FC: fully connected layer.

4.4 Pre-trained State-of-the-Art Deep Networks

The next step was to use state-of-the-art pretrained networks from Imagenet Large Scale Visual Recognition Challenge (ILSVRC) [Russakovsky et al., 2014]. One of the most commonly used deep learning Python package Keras, provides implementation of popular networks such as VGG19 [Simonyan & Zisserman, 2014] and ResNet50 [He et al., 2015].

InceptionV3 [Szegedy et al. 2016] and Xception [Chollet, 2017] were decided to be used to understand distribution of data based on Gleason grade group. The classification layer from these networks were removed in order to retrieve bottleneck features. The 2048 bottleneck features were reduced to 50 features using principal components analysis [Tipping & Bishop, 1999] and further reduced to 5 features using t-distributed stochastic neighbor embedding.

The results of dimensionality reduction of pretrained networks, the bottleneck features, are shown in Figure 21a & 21b. ggg in the labels represents Gleason grade group. It is evident from the visualisation that everything is haywire in the original dataset features, whereas some sort of cluster formation can be observed in case of augmented dataset. However, after a quick inspection, it can be concluded that the cluster formation is not based on different labels, instead the cause of cluster formation is the same augmentation technique. Furthermore, digging deeper into the aforementioned networks was also not useful because even intermediate layers output yielded similar confounding results. It can be deduced from the results that transfer learning does not help much in segregating different Gleason grade groups as none of the groups/classes formed any cluster.

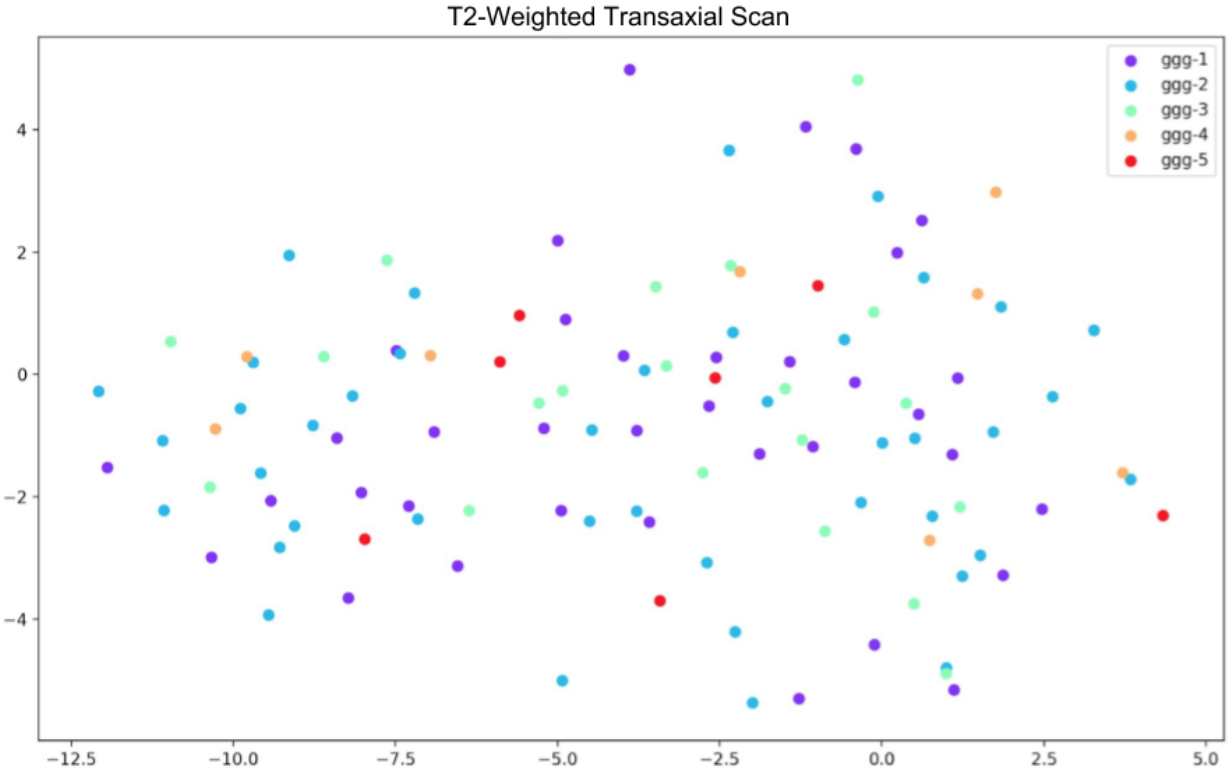

Figure 21a: Dimensionality reduction original dataset (T2-weighted transaxial scan).

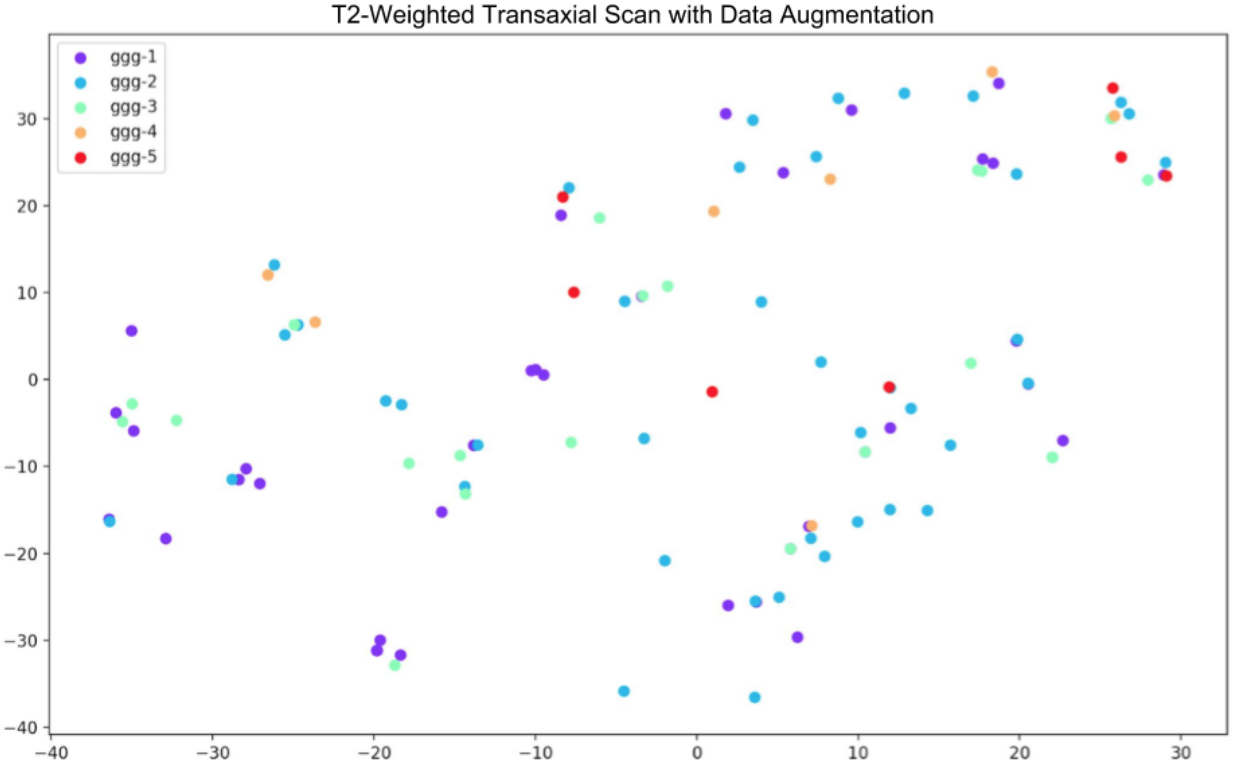

Figure 21b: Dimensionality reduction augmented dataset (T2-weighted transaxial scan).

5. EXPERIMENTS AND RESULTS

5.1 Conventional Classifiers with Image features

In order to employ an organic approach to solve this problem, the initial idea was to apply image processing algorithms to extract features from different image modalities and then use these features as input to a set of conventional machine learning algorithms. When the datasets are not big enough, conventional machine learning is still the method of choice with handwrought features [Fehr et al., 2015; Hussain et al., 2018; Litjens et al., 2015].

Two different types of experiments were conducted to understand the efficacy of data size and segmentation, namely single slice ROI and multi-slice ROI. Grid search with 5-fold cross validation was used to determine the best combination of hyperparameters.

5.1.1 Single Slice ROI 5-Fold Cross Validation

Single slice experiment was the most basic of all experiments, since only one slide from the whole set was used for feature extraction and classification. In this experiment a 48x48 region of interest patch was extracted from a slice out of the series of 19 slices, the one which contained the ROI information. Subsequently HOG and LBP features were calculated of the region of interest. This was done for all four modalities. Three sets of features were separately tested:

- HOG only features
- LBP only features
- HOG and LBP features concatenated

Table 4, 5 and 6 show the results of models training on the three aforementioned features. It can be seen that HOG features of T2-weighted transaxial (tse_tra) modality achieved the highest accuracy 47.32%. And it can be observed in Figure 22 that accuracy and Kappa score do not have a strong correlation.

HOG Desc.	tse_tra		ADC		BVAL		tse_sag	
	KS	Acc.%	KS	Acc.%	KS	Acc.%	KS	Acc.%
BernoulliNB	0.1147	44.6429	0.0503	36.6071	0.0497	38.3929	0.0483	36.6071
PssAgg	0.0146	34.8214	-0.0282	30.3571	-0.0032	32.1429	-0.0631	33.9286
KNeighbors	0.0808	47.3214	0.0897	39.2857	0.1043	41.0714	0.0574	39.2857
RandomForest	0.0234	41.0714	0.0294	33.9286	0.0545	31.2500	0.0233	38.3929
SVC	0.1005	44.6429	0.0390	36.6071	0.1076	36.6071	0.0023	39.2857
LR	0.0245	43.7500	0.0089	36.6071	0.0790	36.6071	0.0243	37.5000
LDA	0.0550	23.2143	0.1523	25.8929	0.0323	25.8929	-0.0570	30.3571
QDA	0.1064	36.6071	0.0685	35.7143	0.0110	36.6071	0.0385	36.6071

Table 4: Conventional classifiers accuracy and Kappa score for HOG descriptors computed on single slice ROI.

LBP Desc.	tse_tra		ADC		BVAL		tse_sag	
	KS	Acc.%	KS	Acc.%	KS	Acc.%	KS	Acc.%
BernoulliNB	0.0867	36.6071	0.1045	37.5000	0.2565	37.5000	0.0669	39.2857
PssAgg	0.1250	33.0357	0.0827	33.0357	0.2167	32.1429	0.0036	32.1429
KNeighbors	0.1016	39.2857	0.1991	40.1786	0.1328	42.8571	0.0225	37.5000
RandomForest	-0.0061	29.4643	0.0868	37.5000	0.0756	37.5000	-0.0046	32.1429
SVC	0.0227	40.1786	0.1552	40.1786	0.0761	43.7500	0.0880	37.5000
LR	0.0205	36.6071	0.0000	38.3929	0.1141	36.6071	0.0823	36.6071
LDA	-0.1553	25.8929	0.0194	30.3571	0.0428	27.6786	-0.0286	25.8929
QDA	0.2156	32.1429	0.0473	33.9286	0.0089	35.7143	0.0263	36.6071

Table 5: Conventional classifiers accuracy and Kappa score for LBP descriptors computed on single slice ROI.

LBP+HOG	tse_tra		ADC		BVAL		tse_sag	
	KS	Acc.%	KS	Acc.%	KS	Acc.%	KS	Acc.%
BernoulliNB	0.1856	46.4286	0.0215	37.5000	0.0740	41.0714	0.0474	38.3929
PssAgg	-0.0097	34.8214	-0.0468	26.7857	-0.0199	32.1429	-0.0539	34.8214
KNeighbors	0.0498	42.8571	0.0763	42.8571	0.1080	41.0714	0.0773	38.3929
RandomForest	0.0461	40.1786	0.0350	35.7143	0.0151	32.1429	0.0359	34.8214
SVC	0.1176	41.9643	0.0000	40.1786	0.0957	36.6071	0.0047	38.3929
LR	0.0371	41.0714	0.0348	36.6071	0.0891	36.6071	0.0000	36.6071
LDA	0.0512	24.1071	0.0436	25.0000	0.1165	25.8929	-0.1374	25.0000
QDA	0.1278	32.1429	0.0229	32.1429	0.0125	32.1429	0.0364	32.1429

Table 6: Conventional classifiers accuracy and Kappa score for LBP+HOG descriptors computed on single slice ROI.

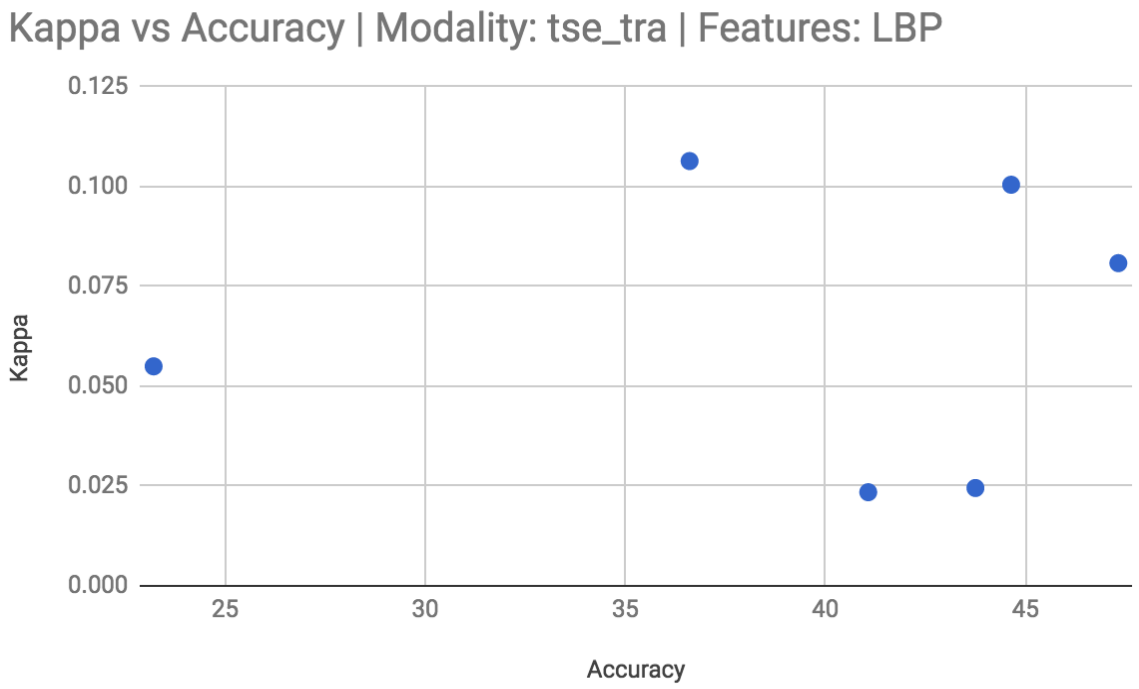

Figure 22: Kappa score vs. Accuracy scatter plot for T2-weighted transaxial scan.

5.1.2 Multi-Slice ROI 5-Fold Cross Validation

Similar to the previous experiment, in this experiment too 48x48 region of interest patches were extracted from the slice which contained the ROI information and also from one slice before and after the ROI annotated slice. Altogether, three consecutive patches were processed. Subsequently HOG and LBP features were calculated of each patch and then the

features were concatenated. The process was done for all four modalities. Three set of features were separately tested:

- HOG only features
- LBP only features
- HOG and LBP features concatenated

It can be observed from table 7, 8 and 9 that multi-slice experiments did not perform as well as single slice experiments accuracy wise. For instance, not even a single classifier for HOG features could yield accuracy over 40%, Figure 23 draws a comparison between single slice and multi-slice experiments for T2-weighted transaxial (tse_tra) modality with HOG features.

HOG Desc.	tse_tra		ADC		BVAL		tse_sag	
	KS	Acc.%	KS	Acc.%	KS	Acc.%	KS	Acc.%
BernoulliNB	0.2053	33.0357	0.0502	36.6071	0.0423	35.7143	0.0236	37.5000
PssAgg	0.0435	17.8571	0.0146	11.6071	-0.0161	15.1786	-0.0471	24.1071
KNeighbors	0.2106	34.8214	0.1859	38.3929	0.1876	32.1429	0.1250	37.5000
RandomForest	0.0586	24.1071	0.1064	27.6786	0.0574	23.2143	0.0666	35.7143
SVC	0.1022	36.6071	0.0803	36.6071	0.0502	37.5000	0.0373	36.6071
LR	0.1879	36.6071	0.1556	36.6071	0.1578	36.6071	0.0709	36.6071
LDA	SVD err.	SVD err.	0.0403	20.5357	-0.0341	25.8929	SVD err.	SVD err.
QDA	0.0235	32.1429	0.0133	32.1429	0.0072	32.1429	0.0255	32.1429

Table 7: Conventional classifiers accuracy and Kappa score for HOG descriptors computed on multi-slice ROI.

LBP Desc.	tse_tra		ADC		BVAL		tse_sag	
	KS	Acc.%	KS	Acc.%	KS	Acc.%	KS	Acc.%
BernoulliNB	0.2419	36.6071	0.1623	36.6071	0.2262	36.6071	0.1567	36.6071
PssAgg	0.3173	29.4643	0.1065	29.4643	0.0436	28.5714	0.0688	27.6786
KNeighbors	0.1961	29.4643	0.1085	39.2857	0.0191	42.8571	0.1783	37.5000
RandomForest	0.1786	22.3214	0.1076	29.4643	0.0332	25.0000	0.0836	33.9286
SVC	0.2170	36.6071	0.1419	39.2857	0.0373	41.0714	0.1611	36.6071
LR	0.1239	36.6071	0.0984	38.3929	0.0456	37.5000	0.2463	36.6071
LDA	0.0810	21.4286	0.0198	26.7857	-0.0633	23.2143	0.0282	26.7857
QDA	0.0757	32.1429	0.1537	32.1429	0.0000	32.1429	0.0186	32.1429

Table 8: Conventional classifiers accuracy and Kappa score for LBP descriptors computed on multi-slice ROI.

HOG+LBP	tse_tra		ADC		BVAL		tse_sag	
	KS	Acc.%	KS	Acc.%	KS	Acc.%	KS	Acc.%
BernoulliNB	0.2053	33.9286	0.0301	36.6071	0.0401	36.6071	0.0236	39.2857
PssAgg	0.0435	16.9643	-0.0026	15.1786	-0.0048	17.8571	-0.0471	21.4286
KNeighbors	0.2106	35.7143	0.1726	38.3929	0.1640	36.6071	0.1445	38.3929
RandomForest	0.0417	25.0000	0.0550	29.4643	0.0827	20.5357	0.0664	33.9286
SVC	0.1022	36.6071	0.0909	37.5000	0.0521	36.6071	0.0373	36.6071
LR	0.1959	36.6071	0.1512	36.6071	0.1568	36.6071	0.1171	36.6071
LDA	SVD err.	SVD err.	-0.0012	24.1071	-0.0265	14.2857	SVD err.	SVD err.
QDA	0.0478	32.1429	0.0658	32.1429	0.0545	32.1429	0.0609	32.1429

Table 9: Conventional classifiers accuracy and Kappa score for LBP+HOG descriptors computed on multi-slice ROI.

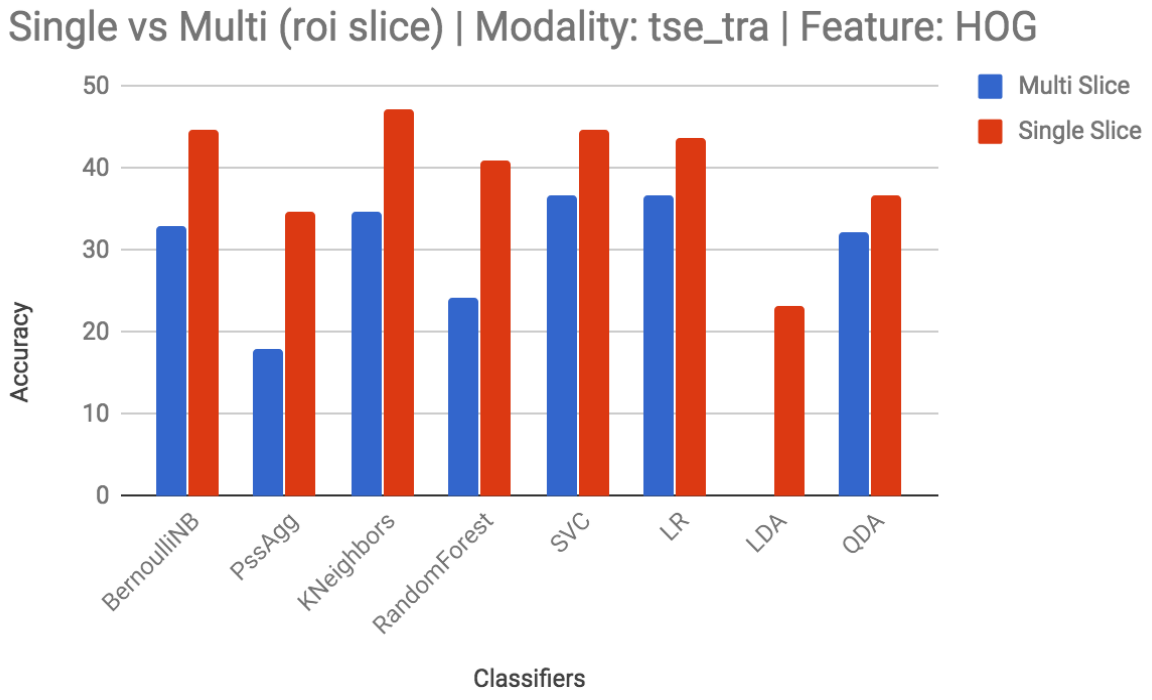

Figure 23: Single vs. Multi-slice bar chart for T2-weighted transaxial scan.

5.2 Partially Trained InceptionV3

The use of pre-trained and partially trained state-of-the-art convolution neural network is a growing academic trend, observed in recent past. More and more studies are incorporating transfer learning and fine tuning of the state-of-the-art algorithms in their approach and it has been fairly successful [Wang et al., 2016; Campanella et al., 2018].

In this experiment Google InceptionV3 was chosen for the partial training. The reason why it was partially trained was because of the sheer size of the model and the memory it needed to be fully trained. Therefore, it was decided to train only top two inception blocks. 140x140 patches of MRI scan centered to ROI were chosen as input to the model and extended augmentation was used in order to yield maximum benefit of the enormity of the dataset. The modality chosen for this experiment was T2-weighted sagittal (tse_sag), the experiment was 5-fold cross validated and the model was trained for 200 epochs.

The experiment was very resource and time exhaustive. As compared to the resources it required and time it took to train the model, the outcome was rather disappointing. The network achieved accuracy of 36.67% in a 5-fold cross validation experiment as shown in Table 10. The performance of each fold can be visualised in Figure 24.

Folds	Acc.%
1st Fold	36.00
2nd Fold	34.78
3rd Fold	36.36
4th Fold	38.10
5th Fold	38.10
5-Fold CV	36.67 (+/- 1.28)

Table 10: InceptionV3 140x140 ROI from T2-weighted sagittal plane scan 5-fold CV.

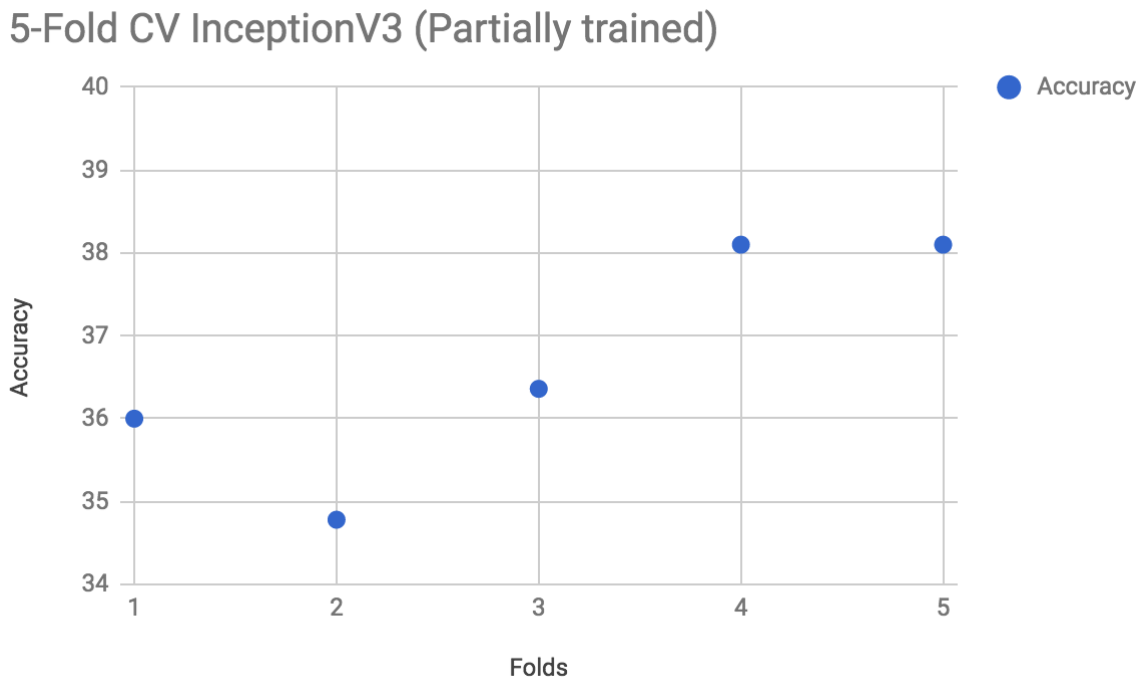

Figure 24: Partially trained InceptionV3, fold wise accuracy.

5.3 Vanilla Models

Vanilla models were the basic incarnation of deep convolutional neural network, as mentioned in the previous chapter. The underlying idea behind using Vanilla models was to utilize the power and efficacy of deep neural network, with architectures that are not as dense as the state-of-the-art architectures like Inception [Szegedy et al., 2015] or Xception [Chollet, 2016].

In this experiment ROI patches of resolution 48x48 were chosen as the input of Vanilla models. The modality chosen for this experiment was T2-Weighted images in sagittal plane. The model hyperparameter configuration are shown in Table 11.

Hyperparameters:	Epochs	Learning Rate	Optimizer
Values:	200	1e-4	Adam

Table 11: Vanilla model configuration.

Table 12 shows the 5-fold cross validated results from four different versions of Vanilla model varying primarily in depth with respect to the number of dense layers:

vbl (baseline = no dense)		v0 (2 dense no BN)		v0.1 (2 dense BN)		v0.2 (3 dense BN)	
Acc.%	KS	Acc.%	KS	Acc.%	KS	Acc.%	KS
44.000	-0.047	36.000	-0.007	32.000	-0.182	40.000	-0.072
52.174	0.240	43.478	0.091	39.130	0.074	26.087	-0.072
50.000	-0.068	36.364	-0.006	27.273	-0.136	31.818	-0.174
42.857	0.191	52.381	0.041	28.571	-0.195	42.857	0.370
52.381	-0.092	47.619	-0.268	42.857	-0.150	47.619	0.141
48.282 (+/-4.066)	0.402	43.168 (+/-6.363)	-0.0201	33.966 (+/-6.058)	-0.142	37.676 (+/- 7.742)	0.0052

Table 12: Vanilla model accuracy and Kappa score for T2-weighted sagittal plane scan.

Figure 19 shows that the baseline version (vbl) is the least dense Vanilla Model as compared the rest of the versions; however, it outperforms all other versions both in terms of accuracy as well as Kappa score as shown in Table 12. Other than that a decreasing trend in accuracy can also be observed as the model depth increases. This implies that the amount of data to train more dense networks, is not sufficient to yield better results.

5.4 Xmasnet Model

Xmasnet is the implementation of the method in a paper written by one of the teams participated in ProstateX competition [Liu et al., 2017]. The problem was to find if an MRI scan has clinically significant cancer or not, although the problem was binary classification, it was deemed appropriate to test the same model for multi-class classification because of the similar nature of the data. 48x48 patches of MRI scan centered to ROI were chosen as input to the model and extended augmentation was used in order to yield maximum benefit of the enormity of the dataset. The modality chosen for this experiment was T2-weighted sagittal (tse_sag), the experiment was 5-fold cross validated and the model was trained for 200 epochs.

Folds	Acc.%	KS
1st Fold	48.000	0.121
2nd Fold	34.783	0.000
3rd Fold	45.445	0.112
4th Fold	42.857	-0.051
5th Fold	52.381	0.310
5-Fold CV	44.695 (+/- 5.869)	0.105

Table 13: Xmasnet model accuracy and Kappa score for T2-weighted sagittal plane scan.

Despite being a smaller model in terms of depth, Xmasnet outperformed InceptionV3 both in accuracy as well as Kappa score as shown in Figure 25.

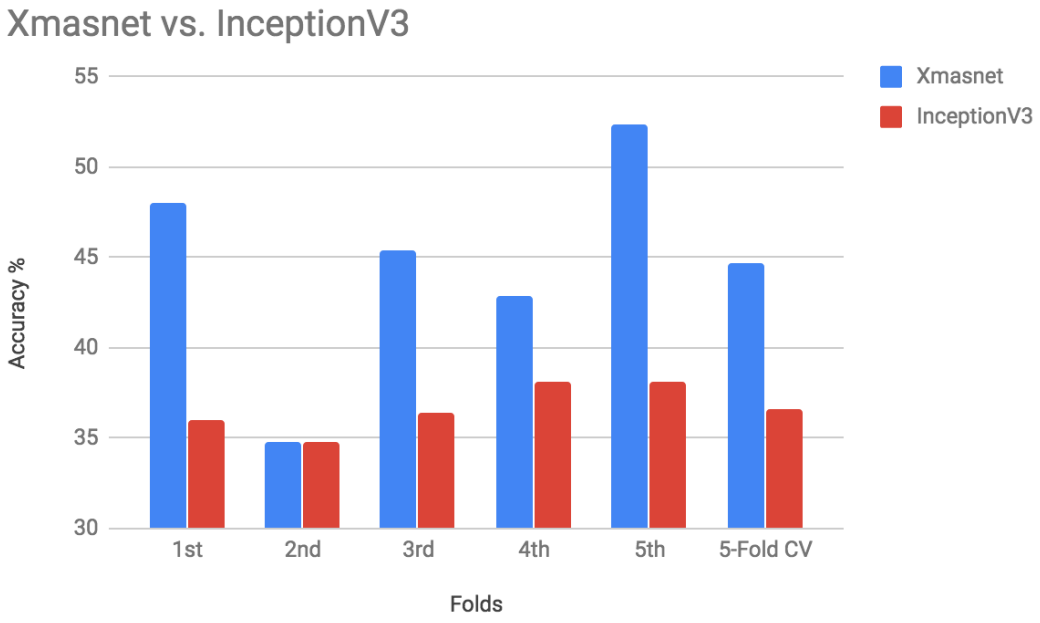

Figure 25: Xmasnet model vs. Inception accuracy bar chart for T2-weighted sagittal plane scan.

5.5 Multi-Channel Experiments

All the mpMRI scan modalities in the dataset were grayscale or single channel images by default. The purpose of multi-channel experiment was to enrich the contrast in the images, so that convolutional networks could exploit the contrast depth and learn more as compared to grayscale input. A couple of studies used the same approach of unifying modalities to create multi-channel input to train convolutional neural networks for binary classification and yielded very impressive results [Liu et al., 2017; Tsehay et al., 2017].

Unfortunately, only two modalities in ProstateX2 dataset contained same plane information, because of which true RGB images could not be generated. The same Vanilla Model and its variants were used to evaluate channel enriched inputs.

5.5.1 Slices to Channel

Three patches from ADC and tse_tra slices were chosen for this experiment, keeping the patch with ROI in the centre. The slices placed adjacently in an array to simulate an RGB image. 48x48x3 ROI patches were used as input to the models. And the models were trained for 200 epoch.

Modality	Fold	Baseline		Version 0		Version 0.1		Version 0.2	
		Acc.%	KS	Acc.%	KS	Acc.%	KS	Acc.%	KS
	1st	44.000	0.141	40.000	-0.033	36.000	-0.064	36.000	0.045
	2nd	43.478	0.028	39.130	0.068	26.087	0.023	39.130	0.133
	3rd	36.364	0.000	36.364	0.000	40.909	-0.013	22.727	0.135
	4th	38.095	0.000	33.333	0.000	33.333	0.019	23.810	-0.082
	5th	52.381	0.272	38.095	-0.154	9.524	-0.037	47.619	-0.008
ADC	5-Fold CV	42.864 (+/- 5.608)	0.0821	37.385 (+/- 2.360)	0.0087	29.171 (+/- 10.929)	-0.0533	33.857 (+/- 9.451)	0.054
		Acc.%	KS	Acc.%	KS	Acc.%	KS	Acc.%	KS
	1st	48.000	-0.028	36.000	-0.076	32.000	0.246	20.000	0.105
	2nd	43.478	0.005	34.780	0.000	34.780	0.000	34.780	0.006
	3rd	40.909	-0.041	31.820	0.000	31.820	-0.039	31.820	0.000
	4th	38.095	0.000	38.100	0.208	28.570	-0.023	42.860	0.202
	5th	42.857	0.012	47.620	-0.003	42.860	0.113	33.330	-0.152
tse_tra	5-Fold CV	42.668 (+/- 3.260)	-0.0184	37.663 (+/- 5.377)	0.0171	34.006 (+/- 4.843)	0.1018	32.558 (+/- 7.347)	0.0767

Table 14: Accuracy and Kappa score for slices to channel experiment.

The multi-channel (slice to channel) didn't perform better than single channel input using Vanilla Models. However, the same trend of decreasing accuracy with increasing depth of the model was observed in slice to channel experiments as shown in Table 14.

5.5.2 Modality to Channel

Only ADC and BVAL were the two modalities which captured prostate image in the same plane (transaxial), therefore, these two modalities were merged to create a multi-channel input, in order to visualise the output, one of the modalities were repeated in the third channel to create a RGB image. 48x48x2 ROI patches were used as input for the models. And the models were trained for 200 epoch.

Modality	Fold	Baseline		Version 0		Version 0.1		Version 0.2	
		Acc.%	KS	Acc.%	KS	Acc.%	KS	Acc.%	KS
	1st	36.000	0.000	32.000	0.000	36.000	0.000	32.000	0.000
	2nd	34.780	0.000	34.780	0.000	30.440	0.000	30.440	0.000
	3rd	36.360	0.000	36.360	-0.114	36.360	0.000	31.820	0.000
	4th	38.100	0.000	38.100	-0.113	38.100	0.000	38.100	0.000
	5th	38.100	-0.322	47.620	-0.235	38.100	0.000	19.050	0.036
ADC + BVAL	**5-Fold CV**	36.667 (+/- 1.278)	-0.086	37.772 (+/- 5.316)	-0.076	35.798 (+/- 2.817)	-0.029	30.279 (+/- 6.206)	-0.033

Table 15: Accuracy and Kappa score for modality to channel experiment.

Modality to channel experiment performed even worse than slice to channel in terms of accuracy. The former yielded 36.667% accuracy for baseline version of vanilla model and the latter yielded 42.68% accuracy for the baseline version of Vanilla Models as shown in Table 15 and 14 respectively.

5.6 Feature Extraction with Shallow Network and training with Conventional Machine Learning Algorithms

This was one of the last experiments conducted, keeping in mind the small size of the dataset and inability of deep networks to learn despite extended augmentation. The experiment was designed to exploit the representational power of neural networks and robustness of classifiers like SVM [Cortes & Vapnik, 1995] and Xgboost [Chen & Guestrin, 2016]. Similar approaches have been used in contemporary research where the dataset was small for deep networks [Rakhlin et al., 2018].

The idea was to create a shallow network of 3 convolutional layers, the last layer sandwiched between max-pooling layer and the output was flattened to get a set of features to be fed to

conventional machine learning algorithms. The network was trained for 200 epochs. Two machine learning algorithms were selected for this task: Support Vector Machine (SVM) and Xgboost. Two different modalities were used to evaluate the experiment pipeline. 48x48 ROI patches were used as input to the models. The results are shown in Table 16.

Classifiers	tse_sag		ADC	
	Acc.%	KS	Acc.%	KS
SVM	36.600	0.121	36.600	0.242
Xgboost	33.920	0.067	33.920	0.277

Table 16: Xgboost and SVM results trained on shallow network bottleneck features.

6. CONCLUSION

Medical data is difficult to obtain in any country. Finland, in particular, is one of those countries because of patient information confidentiality and privacy laws. The project was done in collaboration with Helsinki University Hospital, but unfortunately they were not able to resolve the patient information confidentiality issue during the time period of the project development. Nonetheless, their contribution in transferring domain knowledge is praise worthy. Due to the unavailability of data from a trusted source, it was decided to choose an open dataset as opposed to be bogged down by the problem. It was very difficult to find an open dataset where quality and quantity coexisted. ProstateX2 dataset was not the best dataset, however, it was a good starting point.

The dataset was plagued with many problems such as class imbalance which impeded machine learning models to generalise well. Dataset contained MRI scans of patients with multiple tumor findings of different Gleason grade groups, which rendered full-resolution images useless. Therefore, 32x32 and 48x48 ROI patches were chosen as input to the models, in order to differentiate tumors of different Gleason grade groups from a single patient. A common practice in image processing is to superimpose multiple modalities (e.g. the same plane sagittal, transaxial) to create 3-channel artifacts. Unfortunately ProstateX data only had two modalities per plane which was not enough to create 3-channel artifacts. This is particularly very useful when using convolutional networks because it helps networks to learn meaningful information from sharp contrast, specifically in the tumorous regions.

In deep learning experiments accuracy started decreasing with the increase in the number of dense layers. Batch-normalisation also did not seem to have any serious impact on accuracy. These results make it evident that the dataset was too small to employ deep learning and yield any meaningful results from it. Not only that the dataset lacked resolution and brightness standardisation, some scans were abnormally large as compared to the rest of the dataset. Standard image augmentation techniques were used to generate more data but that did not help much. Extended augmentation which used 360° rotation with 5° shift was completely fruitless. In future advanced data augmentation techniques can be used to generate synthetic yet valid ROI patches, one of these techniques uses Generative Adversarial Networks (GAN) [Goodfellow et al., 2014]. Synthetic ROI can help increase the generalising ability of deep convolutional networks for the least abundant class.

In a nutshell the dataset was a tough one to solve a multi-class classification problem. Even according to the competition organizers, out of 21 teams that participated in the competition, none of them were able to disprove the null hypothesis that the agreement is merely by chance. [“SPIE-AAPM-NCI Prostate MR Gleason Grade Group Challenge “PROSTATEx-2”: Performance Evaluation”, 2018]

7. REFERENCES

aleju/imgaug. Retrieved June 8, 2018, from https://github.com/aleju/imgaug

Alpaydin, E. (2010). Introduction to Machine Learning. [Sl].

Arora, S., Bhaskara, A., Ge, R., & Ma, T. (2013, October 23). *Provable Bounds for Learning Some Deep Representations*. Retrieved from http://arxiv.org/abs/1310.6343

Artificial Neural Networks as Models of Neural Information Processing | Frontiers Research Topic. Retrieved May 8, 2018, from https://www.frontiersin.org/research-topics/4817/artificial-neural-networks-as-models-of-neural-information-processing

Bailey, K. (1994). Numerical taxonomy and cluster analysis. *Typologies and Taxonomies*, *34*.

Bengio, Y., Courville, A., & Vincent, P. (2012, June 24). *Representation Learning: A Review and New Perspectives*. Retrieved from http://arxiv.org/abs/1206.5538

Bernoulli Naive Bayes. Retrieved November 27, 2018, from https://scikit-learn.org/stable/modules/naive_bayes.html#bernoulli-naive-bayes

Block, H. D. (1962). The perceptron: A model for brain functioning. i. *Reviews of Modern Physics*, *34*(1), 123.

CAMELYON16 - Home. Retrieved May 22, 2018, from https://camelyon16.grand-challenge.org/

Campanella, G., Silva, V. W. K., & Fuchs, T. J. (2018). Terabyte-scale Deep Multiple Instance Learning for Classification and Localization in Pathology. *arXiv preprint arXiv:1805.06983*.

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *16*, 321–357.

Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794). ACM.

Chollet, F. (2017). Xception: Deep Learning with Depthwise Separable Convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Cohen, J. (1960). A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement*, *20*(1), 37–46.

Coomans, D., & Massart, D. L. (1982). Alternative k-nearest neighbour rules in supervised pattern recognition: Part 1. k-Nearest neighbour classification by using alternative voting rules. *Analytica Chimica Acta*, *136*, 15-27.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

Cox, D. R. (2018). *Analysis of survival data*. Routledge.

Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., & Singer, Y. (2006). Online passive-aggressive algorithms. *Journal of Machine Learning Research*, *7*(Mar), 551-585.

Dalal, N., & Triggs, B. Histograms of Oriented Gradients for Human Detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*.

Deng, L., & Yu, D. (2014). Deep Learning: Methods and Applications. Retrieved from https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/DeepLearning-NowPublishing-Vol7-SIG-039.pdf

Docrates is the first oncology center in Finland to offer new radiation therapy for metastatic prostate cancer. Retrieved May 7, 2018, from https://www.docrates.com/en/docrates-is-the-first-oncology-center-in-finland-to-offer-new-ra

diation-therapy-for-metastatic-prostate-cancer/

Dong-chen He, He, D.-C., & Wang, L. (1990). Texture Unit, Texture Spectrum, And Texture Analysis. *IEEE Transactions on Geoscience and Remote Sensing: A Publication of the IEEE Geoscience and Remote Sensing Society*, *28*(4), 509–512.

Efron, B. (1979). Bootstrap Methods: Another Look at the Jackknife. *Annals of Statistics*, *7*(1), 1–26.

Epstein, J. I., Egevad, L., Amin, M. B., Delahunt, B., Srigley, J. R., Humphrey, P. A., & Farsad, M., Schiavina, R., Castellucci, P., Nanni, C., Corti, B., Martorana, G., … Fanti, S. (2005). Detection and localization of prostate cancer: correlation of (11)C-choline PET/CT with histopathologic step-section analysis. *Journal of Nuclear Medicine: Official Publication, Society of Nuclear Medicine*, *46*(10), 1642–1649.

Fehr, D., Veeraraghavan, H., Wibmer, A., Gondo, T., Matsumoto, K., Vargas, H. A., … Deasy, J. O. (2015). Automatic classification of prostate cancer Gleason scores from multiparametric magnetic resonance images. *Proceedings of the National Academy of Sciences of the United States of America*, *112*(46), E6265–E6273.

Frakes, W. B., & Baeza-Yates, R. (Eds.). (1992). *Information retrieval: Data structures & algorithms* (Vol. 331). Englewood Cliffs, NJ: prentice Hall.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* (Vol. 1, No. 10). New York, NY, USA:: Springer series in statistics.

Friedman, J. H. (2002). Stochastic gradient boosting. *Computational Statistics & Data Analysis*, *38*(4), 367-378.

Freedman, D. A. (2009). *Statistical models: theory and practice*. cambridge university press.

Geman, S., & Geman, D. (1984). Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine*

Intelligence, *PAMI-6*(6), 721–741.

Global PI-RADS Standardization of Prostate MRI. Retrieved May 17, 2018, from http://www.admetech.org/global-pi-rads-standardization-of-prostate-mri/

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., … Bengio, Y. (2014). Generative Adversarial Nets. In *Advances in Neural Information Processing Systems* (pp. 2672–2680).

Grading Committee. (2016). The 2014 International Society of Urological Pathology (ISUP) Consensus Conference on Gleason Grading of Prostatic Carcinoma: Definition of Grading Patterns and Proposal for a New Grading System. *The American Journal of Surgical Pathology*, *40*(2), 244–252.

Haralick, R. M., Shanmugam, K., & Dinstein, I. 'hak. (1973). Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics*, *SMC-3*(6), 610–621.

He, K., Zhang, X., Ren, S., & Sun, J. (2015, December 10). *Deep Residual Learning for Image Recognition*.

Hinton, G. E. (2009). Deep belief networks. *Scholarpedia Journal*, *4*(5), 5947.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, *313*(5786), 504–507.

Ho, T. K. (1995, August). Random decision forests. In *Document analysis and recognition, 1995., proceedings of the third international conference on* (Vol. 1, pp. 278-282). IEEE.

How is Prostate Cancer Diagnosed? Retrieved June 5, 2018, from https://prostatecancer.chesapeakeurology.com/about-prostate-cancer/how-is-prostate-cancer-diagnosed/

Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, *160*(1), 106–154.

Hussain, L., Ahmed, A., Saeed, S., Rathore, S., Awan, I. A., Shah, S. A., … Awan, A. A. (2018). Prostate cancer detection using machine learning techniques by employing combination of features extracting strategies. *Cancer Biomarkers: Section A of Disease Markers*, *21*(2), 393–413.

ImageNet Large Scale Visual Recognition Competition 2012 (ILSVRC2012). Retrieved May 23, 2018, from http://www.image-net.org/challenges/LSVRC/2012/results.html

Karpathy, A. CS231n Convolutional Neural Networks for Visual Recognition. Retrieved November 27, 2018, from http://cs231n.github.io/convolutional-networks/

Key Statistics for Prostate Cancer | Prostate Cancer Facts. Retrieved May 7, 2018, from https://www.cancer.org/cancer/prostate-cancer/about/key-statistics.html

Kitchen, A., & Seah, J. (2017, August 1). *Deep Generative Adversarial Neural Networks for Realistic Prostate Lesion MRI Synthesis*. Retrieved from http://arxiv.org/abs/1708.00129

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems* (pp. 1097–1105).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Lin, M., Chen, Q., & Yan, S. (2013, December 16). *Network In Network*. Retrieved from http://arxiv.org/abs/1312.4400

Litjens, G. J. S., Barentsz, J. O., Karssemeijer, N., & Huisman, H. J. (2015). Clinical evaluation of a computer-aided diagnosis system for determining cancer aggressiveness in prostate MRI. *European Radiology*, *25*(11), 3187–3199.

Linear and Quadratic Discriminant Analysis. Retrieved November 27, 2018, from https://scikit-learn.org/stable/modules/lda_qda.html

Liu, S., Zheng, H., Feng, Y., & Li, W. (2017). Prostate cancer diagnosis using deep learning

with 3D multiparametric MRI. In *Medical Imaging 2017: Computer-Aided Diagnosis*.

Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, *60*(2), 91–110.

Machine learning | artificial intelligence. Retrieved May 7, 2018, from https://www.britannica.com/technology/machine-learning

Mehrtash, A., Sedghi, A., Ghafoorian, M., Taghipour, M., Tempany, C. M., Wells, W. M., … Fedorov, A. (2017). Classification of clinical significance of MRI prostate findings using 3D convolutional neural networks. In *Medical Imaging 2017: Computer-Aided Diagnosis*.

McCallum, A., & Nigam, K. (1998, July). A comparison of event models for naive bayes text classification. In *AAAI-98 workshop on learning for text categorization* (Vol. 752, No. 1, pp. 41-48).

ML Algorithms addendum: Passive Aggressive Algorithms. Retrieved November 27, 2018, from https://www.bonaccorso.eu/2017/10/06/ml-algorithms-addendum-passive-aggressive-algorithms/

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2012). *Foundations of machine learning*. MIT press.

Moore, G. E. (2006). Cramming more components onto integrated circuits, Reprinted from Electronics, volume 38, number 8, April 19, 1965, pp.114 ff. *IEEE Solid-State Circuits Society Newsletter*, *11*(3), 33–35.

Ojala, T., Pietikäinen, M., & Harwood, D. (1996). A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, *29*(1), 51–59.

Passive Aggressive Algorithms. Retrieved November 3, 2018, from https://scikit-learn.org/stable/modules/linear_model.html#passive-aggressive

PROSTATEx Grand Challenge | SPIE Medical Imaging. Retrieved May 17, 2018, from http://spie.org/PROSTATEx

Rakhlin, A., Shvets, A., Iglovikov, V., & Kalinin, A. A. (2018, June). Deep Convolutional Neural Networks for Breast Cancer Histology Image Analysis. In *International Conference Image Analysis and Recognition* (pp. 737-744). Springer, Cham.

Rote, G. (1991). Computing the minimum Hausdorff distance between two point sets on a line under translation. *Information Processing Letters*, *38*(3), 123-127.

Roweis, S. T., & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *science*, *290*(5500), 2323-2326.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., … Fei-Fei, L. (2014, September 1). *ImageNet Large Scale Visual Recognition Challenge*. Retrieved from http://arxiv.org/abs/1409.0575

Samuel, A. L. (1988). Some Studies in Machine Learning Using the Game of Checkers. I. In *Computer Games I* (pp. 335–365). Springer, New York, NY.

Schmidhuber, J. (2015). Deep Learning. *Scholarpedia Journal*, *10*(11), 32832.

Simonyan, K., & Zisserman, A. (2014, September 4). *Very Deep Convolutional Networks for Large-Scale Image Recognition*. Retrieved from http://arxiv.org/abs/1409.1556

Smale, S., & Yao, Y. (2006). Online learning algorithms. *Foundations of computational mathematics*, *6*(2), 145-170.

Soldea, O., Unel, M., & Ercil, A. (2010). Moments of Elliptic Fourier Descriptors. In *2010 20th International Conference on Pattern Recognition* (pp. 3521–3524). Istanbul, Turkey: IEEE.

SPIE-AAPM-NCI Prostate MR Gleason Grade Group Challenge “PROSTATEx-2”: Performance Evaluation. Retrieved September 16, 2018, from

http://amos3.aapm.org/abstracts/pdf/127-38304-424554-126249.pdf

SPIE-AAPM-NCI Prostate MR Gleason Grade Group Challenge. Retrieved August 19, 2018, from https://www.aapm.org/GrandChallenge/PROSTATEx-2/

Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images - IEEE Journals & Magazine. Retrieved May 15, 2018, from https://ieeexplore.ieee.org/document/4767596/

Support Vector Machines. Retrieved November 27, 2018, from https://scikit-learn.org/stable/modules/svm.html#svm-classification

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., … Rabinovich, A. (2015). Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2015.7298594

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr.2016.308

The Basics of Prostate Cancer. Retrieved May 5, 2018, from https://www.webmd.com/prostate-cancer/guide/understanding-prostate-cancer-basics

Tipping, M. E., & Bishop, C. M. (1999). Probabilistic Principal Component Analysis. *Journal of the Royal Statistical Society. Series B, Statistical Methodology*, *61*(3), 611–622.

Tsehay, Y. K., Lay, N. S., Roth, H. R., Wang, X., Kwak, J. T., Turkbey, B. I., … Summers, R. M. (2017). Convolutional neural network based deep-learning architecture for prostate cancer detection on multiparametric magnetic resonance images. In S. G. Armato & N. A. Petrick (Eds.), *Medical Imaging 2017: Computer-Aided Diagnosis* (Vol. 10134, p. 1013405). Orlando, Florida, United States: SPIE.

Tucker, A. B. (Ed.). (2004). *Computer science handbook*. CRC press.

Wang, D., Khosla, A., Gargeya, R., Irshad, H., & Beck, A. H. (2016, June 18). *Deep Learning for Identifying Metastatic Breast Cancer*. Retrieved from http://arxiv.org/abs/1606.05718

Welling, M. (2005). Fisher linear discriminant analysis. *Department of Computer Science, University of Toronto*, *3*(1).

XGBoost: A Concise Technical Overview. Retrieved November 27, 2018, from https://www.kdnuggets.com/2017/10/xgboost-concise-technical-overview.html

Yu, L., Yang, X., Chen, H., Qin, J., & Heng, P. (2017). Volumetric ConvNets with Mixed Residual Connections for Automated Prostate Segmentation from 3D MR Images. *AAAI*.

Zhang, H. (2004). The optimality of naive Bayes. *AA*, *1*(2), 3.

Zhu, Y., Wang, L., Liu, M., Qian, C., Yousuf, A., Oto, A., & Shen, D. (2017). MRI-based prostate cancer detection with high-level representation and hierarchical classification. *Medical Physics*, *44*(3), 1028–1039.

2.8. Density Estimation. Retrieved September 15, 2018, from http://scikit-learn.org/stable/modules/density.html

3.3. Model evaluation: quantifying the quality of predictions — scikit-learn 0.19.1 documentation. Retrieved June 8, 2018, from http://scikit-learn.org/stable/modules/model_evaluation.html#accuracy-score

3.3. Model evaluation: quantifying the quality of predictions — scikit-learn 0.19.1 documentation. Retrieved June 8, 2018, from http://scikit-learn.org/stable/modules/model_evaluation.html#cohen-kappa