

American University in Cairo

AUC Knowledge Fountain

Theses and Dissertations

Student Research

Winter 1-31-2021

Deep Learning for Multi-Tissue Cancer Classification of Gene Expressions

Tarek Khorshed

tarek_khorshed@aucegypt.edu

Follow this and additional works at: <https://fount.aucegypt.edu/etds>



Part of the [Artificial Intelligence and Robotics Commons](#), [Bioinformatics Commons](#), [Data Science Commons](#), and the [Genomics Commons](#)

Recommended Citation

APA Citation

Khorshed, T. (2021). *Deep Learning for Multi-Tissue Cancer Classification of Gene Expressions* [Doctoral Dissertation, the American University in Cairo]. AUC Knowledge Fountain.

<https://fount.aucegypt.edu/etds/1493>

MLA Citation

Khorshed, Tarek. *Deep Learning for Multi-Tissue Cancer Classification of Gene Expressions*. 2021. American University in Cairo, Doctoral Dissertation. *AUC Knowledge Fountain*.

<https://fount.aucegypt.edu/etds/1493>

This Doctoral Dissertation is brought to you for free and open access by the Student Research at AUC Knowledge Fountain. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of AUC Knowledge Fountain. For more information, please contact thesisadmin@aucegypt.edu.

PhD Dissertation

Academic and Dissertation Advisors

Prof. Mohamed Moustafa

Prof. Ahmed Rafea

Deep Learning for Multi-Tissue Cancer Classification of Gene Expressions

“This research is about saving lives of Cancer Patients”

MAY 2020

Tarek Khorshed

The American University in Cairo, Egypt

AUC SID 800-09-0336

ABSTRACT	6
ACKNOWLEDGMENTS	7
PUBLICATIONS	8
CHAPTER 1.....	9
1. INTRODUCTION	9
1.1 The Global Burden of Cancer	9
1.2 Challenges in Early Diagnosis and Treatment of Cancer	10
1.3 Cancer Genomics.....	10
1.4 Early Cancer Diagnosis using Gene Expressions	11
1.5 Complexity in Cancer Classification using Gene Expression Data.....	13
1.6 Deep Learning for Early Cancer Diagnosis	15
CHAPTER 2.....	16
2. LITERATURE SURVEY	16
2.1 Cancer Genomics.....	16
2.2 Next Generation Sequencing (NGS)	17
2.3 Gene Expression Analysis.....	17
2.4 Cancer Classification using Gene Expressions	19
2.5 Gene Feature Selection	19
2.5.1 Gene Filtering.....	20
2.5.2 Gene Wrapping	20
2.5.3 Gene Filtering using Ranking.....	21
2.6 Cancer Classification Methods using Gene Expressions	23
2.6.1 Support Vector Machines (SVM).....	24
2.6.2 K-Nearest Neighbors	24
2.6.3 Fuzzy Decision Trees (DT)	25
2.6.4 AdaBoost.....	25
2.6.5 Particle Swarm Optimization	26
2.6.6 Random Forests (RF)	26
2.6.7 Deep Neural Forest Models (DFN)	27
2.6.8 Self-training Subspace Clustering	28
2.6.9 One-Class Logistic Regression.....	29
2.6.10 Multilayer Recursive Feature Elimination	30
2.6.11 Graph Structure Algorithms	31
2.6.12 Genetic Algorithms (GA)	31
2.6.13 Ensemble Classifiers	31
2.7 Deep Learning	32
2.7.1 Learning using Deep Multilayer Architectures	32
2.7.2 Feature Extraction using Representation Learning.....	32
2.7.3 Training Deep Architectures using Gradient Descent & Backpropagation	33
2.8 Convolutional Neural Networks.....	34
2.8.1 CNN Overview.....	34
2.8.2 (ALEX-Net) ImageNet Classification with Deep Networks	35
2.8.3 (ZF-Net) Visualizing and Understanding Convolutional Networks	36
2.8.4 (VGG-Net) Very Deep Convolutional Networks for Image Recognition	37
2.8.5 Inception Model Architectures	38
2.8.6 (Google-Net) Going Deeper with Convolutions.....	39
2.8.7 Deep Residual Learning Framework.....	40
2.8.8 (ResNet) Deep Residual Learning Networks for Image Recognition	41
2.8.9 (Inception-ResNet) Impact of Residual Connections on Learning.....	42
2.8.10 (DenseNet) Densely Connected Convolutional Networks	43
2.8.11 (NasNet) Learning Transferable Architectures for Image Recognition.....	44
2.8.12 (MobileNet) Efficient CNNs for Mobile Vision Applications	45
2.9 ROC Analysis for Evaluation of Classification Performance.....	46

CHAPTER 3.....	48
3. RESEARCH METHODOLOGY	48
3.1 Problem Definition	48
3.2 Research Objectives	50
3.3 Approach.....	53
3.3.1 Using Deep Learning to Design a Multi-Tissue Cancer Classifier.....	53
3.3.2 Overcoming Complexity in Feature Extraction of Gene Expression Data	53
3.3.3 Deep Learning Architecture for Multi-Tissue Cancer Classification	54
3.3.4 Transfer Learning using Genomic Signatures of Multiple Tumors	55
3.3.5 Deep Reinforcement Learning to Discover the Optimal Network Architecture	56
3.3.6 Visualizing Genomic Relationships of Gene Expressions Across Multiple Tumors	57
CHAPTER 4.....	58
4. METHODS.....	58
4.1 Deep Learning System Architecture.....	58
4.2 Convolutional Neural Networks.....	59
4.3 Gene Expression Data Representation for CNNs	60
4.4 Learning Genomic Signatures using Convolutions	61
4.5 Gene eXpression Network (GeneXNet) Architecture	64
4.6 GeneXNet Building Block Formulation	65
4.7 Transfer Learning using Genomic Signatures Across Multiple Cancer Tumor Types	70
4.8 Gene eXpression Network Training and Optimization	73
4.8.1 Optimization Objective and Loss Function	73
4.8.2 Optimization Algorithms with Accelerated Gradient & Adaptive Learning	75
4.9 Visualizing Genomic Relationships of Gene Expressions Across Multiple Cancer Tumors	80
4.9.1 Visualizing Class-Discriminative Localization Maps of Gene Expressions	81
4.9.2 Visualizing Molecular Clusters of Intermediate Feature Maps	82
CHAPTER 5.....	85
5. EXPERIMENTS.....	85
5.1 Datasets	85
5.2 Classification Experiments	87
5.2.1 Experiment 1 - Multi-tissue Multi-class classification	87
5.2.2 Experiment 2 - Multi-Tumor Binary classification	87
5.2.3 Experiment 3 - Comparison between Transfer Learning and Full Training	87
5.2.4 Experiment 4 – Transfer Learning for Tumors Lacking Sufficient Training Data.....	87
5.2.5 Experiment 5 – Comparison between GeneXNet & State-of-the-art models	88
5.2.6 Experiment 6 – Comparison between different GeneXNet Architectures.....	88
5.3 Training, Optimization and Evaluation	89
5.4 Results	90
5.5 Analysis of Classification Results.....	94
5.6 Visualizing Class-Discriminative Localization Maps	95
5.7 Biological Significance of Visualizing Class-Discriminative Maps	96
5.8 Visualizing Molecular Clusters of Intermediate Feature Maps	97
5.9 Biological Significance of Visualizing Molecular Clusters of Intermediate Feature Maps.....	98
CHAPTER 6.....	99
6. CONCLUSIONS.....	99
6.1 Motivation.....	99
6.2 Contributions.....	99
6.3 Analysis	101
6.4 Biological Significance	102
6.5 Future work	104
6.6 COVID-19	105
REFERENCES	106

ABSTRACT

We contribute in saving the lives of cancer patients through early detection and diagnosis, since one of the major challenges in cancer treatment is that patients are diagnosed at very late stages when appropriate medical interventions become less effective and full curative treatment is no longer achievable. Cancer classification using gene expressions is extremely challenging given the complexity and high dimensionality of the data. Current classification methods typically rely on samples collected from a single tissue type and perform a prerequisite of gene feature selection to avoid processing the full set of genes. These methods fall short in taking advantage of genome-wide next generation sequencing technologies which provide a snapshot of the whole transcriptome rather than a predetermined subset of genes. We propose a Deep Learning framework for cancer diagnosis by developing a multi-tissue cancer classifier based on whole-transcriptome gene expressions collected from multiple tumor types covering multiple organ sites. We introduce a new Convolutional Neural Network architecture called Gene eXpression Network (GeneXNet), which is specifically designed to address the complex nature of gene expressions. Our proposed GeneXNet provides capabilities of detecting genetic alterations driving cancer progression by learning genomic signatures across multiple tissue types without requiring the prerequisite of gene feature selection. We design an end-to-end Deep Reinforcement Learning framework that automatically learns the optimal network architecture together with the associated optimal hyperparameters that maximizes the performance of our multi-tissue cancer classifier. Our framework eliminates the manual process of handcrafting the design of deep network architectures and the manual process of hyperparameter optimization on the target dataset. Our model achieves 98.9% classification accuracy on human samples representing 33 different cancer tumor types across 26 organ sites. We demonstrate how our model can be used for transfer learning to build classifiers for tumors lacking sufficient samples to be trained independently. We contribute in providing medical professionals with more confidence in using Deep Learning for medical diagnosis by introducing visualization procedures to provide biological insight on how our network is performing classification across multiple tumors. To our knowledge, this is the first effort to develop a multi-tissue cancer classifier based on a full set of whole-transcriptome gene expressions collected from tumors across different tissue types without requiring a prerequisite process of gene feature selection.

ACKNOWLEDGEMENTS

I would like to sincerely thank my academic and dissertation advisors Prof. Mohamed N. Moustafa and Prof Ahmed Rafea. They have continuously provided me with a wealth of in-depth technical knowledge in my research area. Their inspiration, motivation and continuous support is greatly appreciated. I would also like to thank the American University in Cairo and in particular the school of Sciences & Engineering, the office of Associate Dean for Graduate Studies & Research and the office of Dean of Graduate studies for the support they have provided me to complete my degree.

Finally, I would like to thank my wife Dina and my sons Omar and Alei for their patience, endurance and support during my years of studies.

PUBLICATIONS

The following journal and conference papers have been accepted for publication as part of the research towards this Ph.D. degree:

Date	Type	Publication	Paper Title
April 2020	Conference	IEEE International Conference on Big Data Computing and Machine Learning, Oxford, UK.	Multi-Tissue Cancer Classification of Gene Expressions using Deep Learning
May 2020	Journal	IEEE Access Journal	Deep Learning for Multi-Tissue Cancer Classification of Gene Expressions (GeneXNet) DOI: 10.1109/ACCESS.2020.2992907
July 2020	Conference	IEEE World Congress on Computational Intelligence (IEEE WCCI), Glasgow, UK.	Learning and Visualizing Genomic Signatures of Cancer Tumors using Deep Neural Networks

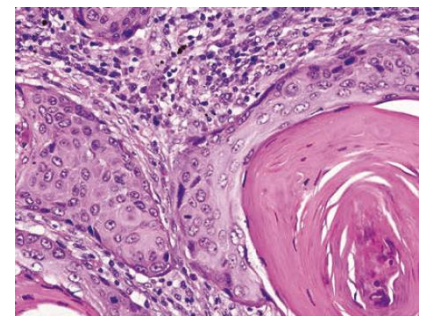
CHAPTER 1

1. INTRODUCTION

“This research is about saving lives of Cancer Patients”

1.1 The Global Burden of Cancer

The World Health Organization reports that cancer is an incurable disease which is considered one of the leading causes of death worldwide accounting for an estimated 9.6 million deaths in 2018 [1]. The cumulative risk of incidence indicates that 1 in 8 men and 1 in 10 women will develop the disease in a lifetime [2]. Lung cancer is the most commonly diagnosed cancer and the leading cause of cancer death, followed by breast, prostate, colorectal, stomach, and liver cancer [2].



Squamous cell carcinoma Lung cancer [1]

Cancer is a generic term for a large group of diseases that can affect any part of the body. Other terms used are malignant tumors and neoplasms. One defining feature of cancer is the rapid creation of abnormal cells that grow beyond their usual boundaries, and which can then invade adjoining parts of the body and spread to other organs, the latter process is referred to as metastasizing. Metastases are a major cause of death from cancer [3].

9.6 million Cancer deaths in 2018 worldwide

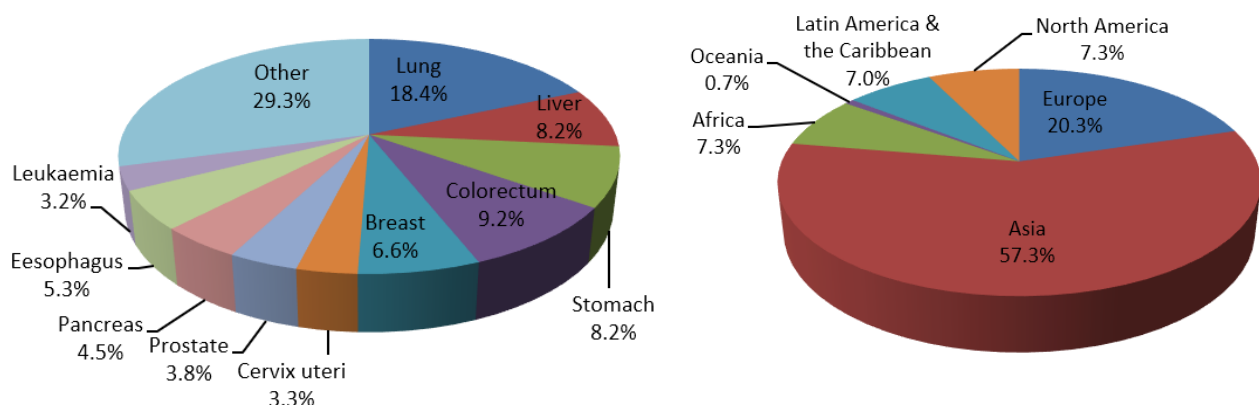


Figure 1.1 Estimated world cancer mortality in 2018 [1], [2]

1.2 Challenges in Early Diagnosis and Treatment of Cancer

Despite the dramatic impact of Cancer and high mortality rates, many of these deaths can be avoided. It is reported by the World Health Organization that between 30-50% of cancer death cases can be prevented through early detection and treatment [3]. A major challenge is that the disease is not diagnosed early enough to allow for appropriate and effective treatment. When Cancer patients are diagnosed at very late stages, appropriate treatment interventions become less effective and full curative treatment is no longer achievable [1].

The increasing complexity of the disease and its molecular biology has made it extremely difficult for medical experts to use traditional patient diagnosis and laboratory screening techniques to detect early signs and symptoms of cancer. In absence of any early detection or screening and treatment intervention, patients are diagnosed at very late stages when curative treatment is no longer an option [4].

One of the major challenges of Cancer treatment, especially when using Chemotherapy, is to maximize the drug efficiency but at the same time minimize the toxic effects on healthy cells. As a result, accurate classification and diagnosis of the Cancer tumor is crucial to successful treatment. Conventional laboratory screening techniques for Cancer classification usually rely on the biological insights of the medical experts and have primarily focused on the morphological appearance of the tumor. This has serious limitations, since tumors with similar histopathological appearance can follow significantly different clinical courses and show different responses to therapy [1].

Accordingly, advancements in cancer classification and prediction play an important role in early detection since a major challenge in cancer treatment is that patients are diagnosed at very late stages where appropriate interventions become less effective and full curative treatment is no longer achievable [4]. Cancer classification can be divided into two categories which are class discovery and class prediction. The task of class discovery is to identify a new tumor which was previously unrecognized. Class prediction is the task of diagnosing a tumor sample and assigning it to the correct predefined class [4].

1.3 Cancer Genomics

Technological advances in structural genomics have allowed studying the full set of DNAs in the human genome [4], [25]. DNA is a molecule in the cell nucleus that contains instructions for making proteins. A segment of DNA that contains information for making a protein is called a gene [4]. During the transcription process, DNA that makes up a gene is copied into a

complementary molecule called messenger RNA (mRNA). The mRNA moves from the nucleus to the cytoplasm where it interacts with ribosomes which are the protein factories of the cell [4]. DNA alterations can affect the structure, function and amount of corresponding proteins leading to a change in a cell's behavior from normal to cancerous [24]. Next generation sequencing (NGS) methods such as whole-genome DNA sequencing and Total RNA sequencing are considered revolutionary technologies for studying genetic changes in Cancer [22], [27]. These technologies provide great potential for cancer classification and better understanding of tumor progression given their ability to sequence thousands of genes at one time and detect multiple types of genomic and transcriptome gene expression alterations [20], [21], [25]. They provide capabilities for comparing the sequence of DNA and RNA in cancer cells with that in normal cells, such as blood or saliva to identify genetic changes that may be driving the growth of a tumor in addition to measuring the activity of genes to understand which proteins are abnormally active in cancer cells leading to uncontrolled growth [26]. Gene expression analysis using total RNA sequencing provides a snapshot of the whole transcriptome rather than a predetermined subset of genes, enables testing multiple genes simultaneously and can detect both coding plus multiple forms of noncoding RNA [22]. These methods have eliminated many limitations involved in microarray based experiments that were previously used for measuring gene expressions [22], [25], [27].

1.4 Early Cancer Diagnosis using Gene Expressions

Gene expressions have been extensively used in biological research and cancer classification [5], [6], [7], [8], [9], [10], [13], [11], [17]. Individual proteins determine the cell function and at the same time the protein synthesis is dependent on which genes are expressed by the cell. Accordingly, the expression pattern of a gene provides indirect information about a cell function [1]. Gene expression refers to the process of translating information in DNA into functional products including proteins and non-coding RNA [4]. While Microarrays have traditionally been used for gene expression analysis, they have shown many limitations since the snapshot of the transcriptome they provide is incomplete and they cannot detect previously unidentified genes or transcripts [21], [22], [25].

Gene expression quantification can be used to identify which genes are preferentially expressed in various tissues. Transcription produces what is referred to as precursor messenger RNA (pre-mRNA) which undergoes further modifications leading to mature mRNA [4].

By collecting mRNA samples for tumors of known classes, supervised learning can be used to build discriminative models which can learn the gene patterns of the underlying disease and then be used to predict the tumor class of new patient samples which were not previously diagnosed [1]. This is considered a great achievement as there are many Microarray experiments which demonstrate how it was possible to classify and distinguish between certain cancer types using data classification even though they are clinically indistinguishable [1], [72], [73].

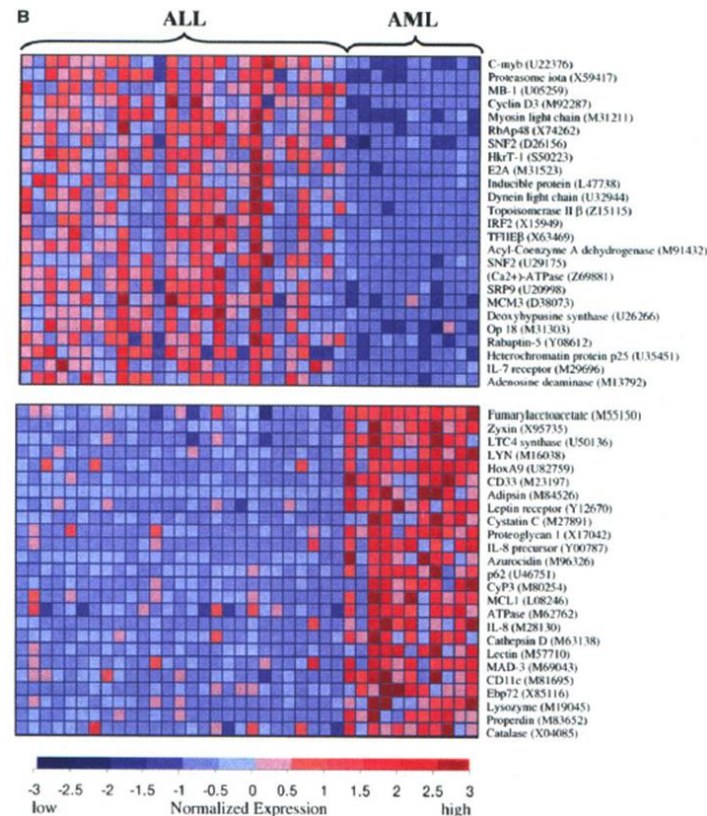


Figure 1.2 Classification of AML and ALL Leukemia using Gene Expression Data [73]

For example, Figure 1.2 shows the classification between two types of Leukemia Cancer (AML and ALL) which are clinically indistinguishable [73]. The figure shows how Clustering of microarray gene expression data was used to distinguish between Acute Myeloid Leukemia (AML) and Acute Lymphoblastic leukemia (ALL) using only data classification. Rows correspond to genes and columns correspond to human samples [73].

Another example of a microarray experiment is shown in Figure 1.3 which was used to analyse a total of 78 Breast Cancer patients to develop what is known as the PAM50 Breast Cancer Intrinsic Classifier which predicts the breast cancer type out of several classes [74]. This classifier predicts the Breast Cancer type out of several classes which are: Luminal A, Luminal B, Basal-like, Human Epidermal Growth Factor receptor 2 (HER2)+ [74].

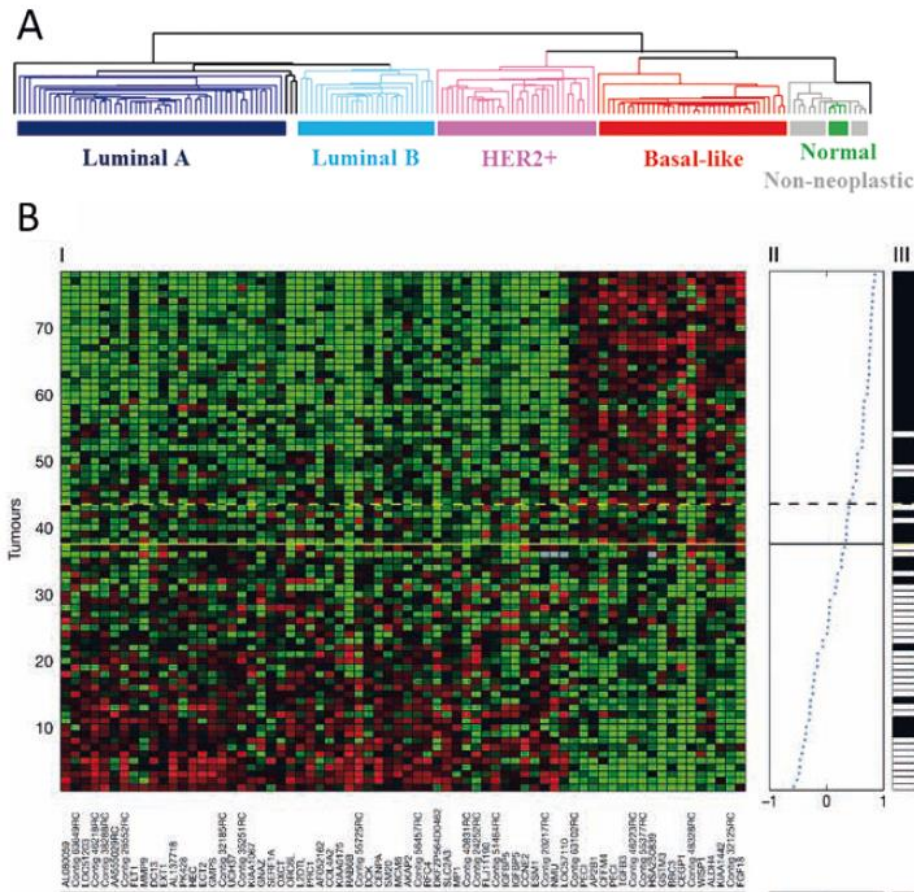


Figure 1.3 PAM50 Classifier for Classification of Breast Cancer using Gene Expression Data [74]

1.5 Complexity in Cancer Classification using Gene Expression Data

Despite all these potential capabilities, cancer classification using gene expressions produced from Total RNA sequencing is extremely challenging given the complexity and massive amount of genetic data that is produced [20], [21], [25], [26], [38]. The magnitude of variants obtained from RNA-Sequencing is exponential which makes it difficult for traditional bioinformatics and machine learning approaches to evaluate genetic variants for disease prediction [4], [22], [23]. Gene expression data is characterized by being very high in dimensionality in terms of having a very large number of features representing the genes, and a very small number of training data representing the patient samples [9], [22], [33]. Complexity is also due to the fact that only a small subset of genes might be influencing the cancer tumor being diagnosed [4], [29].

Current cancer classification methods avoid processing the full set of genes to overcome these complexities and are mainly based on performing a process of gene feature selection as a prerequisite to the classifier learning process [28], [29], [30], [31].

Gene feature selection is the process of selecting a small subset of informative genes which are discriminative among the full set of genes collected from the tumor samples [32], [33]. Gene feature selection will allow the learning process to proceed, but the resulting classifier will not have the opportunity to learn the molecular signatures of genes which have been excluded and their influence on the underlying cancer tumor [34], [35].

Current methods for cancer classification follow the approach of feature engineering and are based on applying innovative gene feature selection techniques as a prerequisite to the classifier learning process to discover a small subset of informative genes which are discriminative among the tumor being analysed [28], [29], [31]. Gene selection methods can be generally classified into filtering, wrapping and embedded methods [32], [33]. The accuracy of such a classifier depends heavily on the successful identification of these discriminative features [34], [35]. Furthermore, the same classification method might not succeed in achieving the same accuracy if applied on a tumor for a different tissue type which will most likely have a different subset of informative genes [1].

Substantial work has been done for cancer classification by performing gene feature selection and building on traditional machine learning methods such as Support Vector Machines [15], [18], [30], Random Forests [14], Decision Trees [16], AdaBoost [11], K-Nearest Neighbor [14] and Genetic algorithms [9], [11]. Current classification methods which are based on gene feature selection are not optimal for early cancer diagnosis. This is because these methods will fall short in taking the full advantage of DNA and RNA sequencing technologies to discover the correlated patterns between genes across the full set of DNAs in the human genome and to detect multiple types of genetic alterations that may be driving the growth of a tumor across the whole transcriptome rather than a predetermined subset of genes [5], [6]. Another limitation of current methods is that they typically rely on gene expressions collected mainly from a single cancer tissue type based on the same anatomical site of origin. This approach does not utilize the full potential of the recent emerging whole-genome sequencing technologies and data produced by large-scale genomic projects which are producing detailed molecular characterizations of thousands of tumors using genome-wide platforms [38]. Recent studies which have performed an integrated multiplatform analysis across multiple cancer types have revealed molecular classification within and across tissues of origin [5], [7]. The results of these studies have recommended that the traditional approach of anatomic cancer classification should be supplemented by classification based on molecular alterations shared by tumors across different tissue types [5].

1.6 Deep Learning for Early Cancer Diagnosis

This has motivated our research for early diagnosis of cancer by leveraging the latest deep learning methods to develop a comprehensive multi-tissue cancer classifier. Our proposed classifier is based on molecular signatures of whole-transcriptome wide gene expressions, that are collected from human samples representing multiple cancer tissue types covering multiple organ sites of origin. Our approach using deep learning eliminates the need for discovering a predefined subset of genes by combining the process of gene feature selection and classification into one end-to-end learning system. We propose a new Convolutional Neural Network architecture called “Gene eXpression Network” (GeneXNet) which is specifically designed to learn the complex nature of whole-transcriptome gene expressions and which gives the opportunity to design cancer classifiers with capabilities of detecting more complex types of genetic alterations by learning the genomic signatures shared across multiple cancer tissue types. To our knowledge, this is the first effort to develop a multi-tissue cancer classifier based on a full set of whole-transcriptome wide gene expressions collected from tumors across different tissue types without requiring a prerequisite process of gene feature selection. We demonstrate how our model can perform transfer learning to build classifiers for other types of cancer tumors which are lacking sufficient patient samples to be trained independently. We design an end-to-end Deep Reinforcement Learning framework to automatically learn the optimal Deep Neural Network architecture together with the associated optimal hyperparameters that maximizes the performance of our multi-tissue cancer classifier. We introduce visualization procedures to provide more biological insight on how our model is performing cancer classification across multiple tumor types. We visualize gene localization maps highlighting the important regions in the gene expressions influencing the tumor class prediction. We also visualize the molecular clusters formed by intermediate gene expression feature maps learned by the network which helps in revealing the genomic relationships of gene expressions that are influential in the tumor progression.

CHAPTER 2

2. LITERATURE SURVEY

2.1 Cancer Genomics

DNA is a molecule in the cell nucleus that contains instructions for making proteins. A segment of DNA that contains information for making a protein is called a gene [4]. During the transcription process, DNA that makes up a gene is copied into a complementary molecule called messenger RNA (mRNA). The mRNA moves from the nucleus to the cytoplasm where it interacts with ribosomes which are the protein factories of the cell [4]. DNA alterations can affect the structure, function and amount of corresponding proteins leading to a change in a cell's behaviour from normal to cancerous [24].

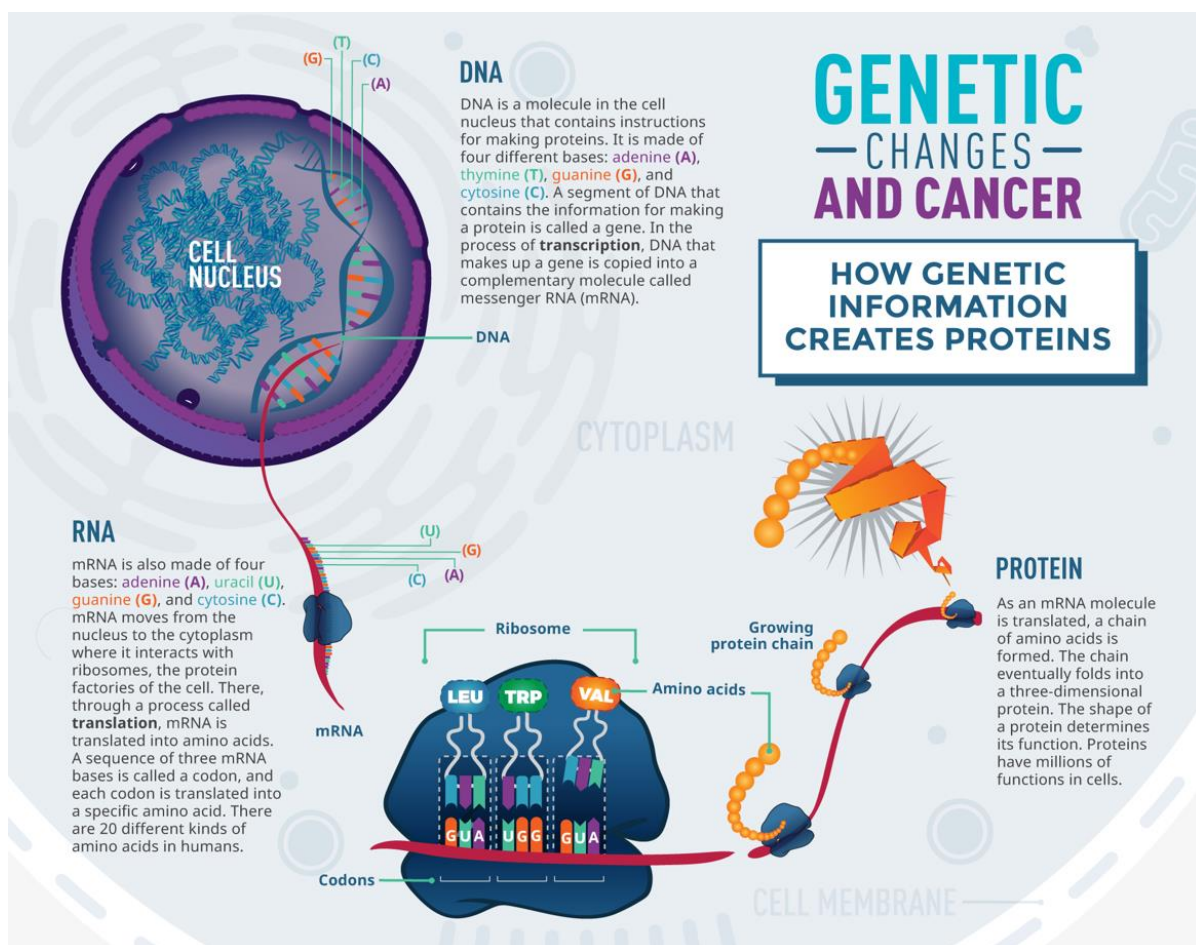


Figure 2.1 Genetic Changes and Cancer [4]

2.2 Next Generation Sequencing (NGS)

Technological advances in structural genomics have allowed studying the full set of DNAs in the human genome [4], [25]. Next generation sequencing (NGS) methods such as whole-genome DNA sequencing and Total RNA sequencing are considered revolutionary technologies for studying genetic changes in Cancer [22], [27]. These technologies provide great potential for cancer classification and better understanding of tumor progression given their ability to sequence thousands of genes at one time and detect multiple types of genomic and transcriptome gene expression alterations [20], [21], [25]. They provide capabilities for comparing the sequence of DNA and RNA in cancer cells with that in normal cells, such as blood or saliva to identify genetic changes that may be driving the growth of a tumor in addition to measuring the activity of genes to understand which proteins are abnormally active in cancer cells leading to uncontrolled growth [26]. Gene expression analysis using total RNA sequencing provides a snapshot of the whole transcriptome rather than a predetermined subset of genes, enables testing multiple genes simultaneously and can detect both coding plus multiple forms of noncoding RNA [22]. These methods have eliminated many limitations involved in microarray based experiments that were previously used for measuring gene expressions [22], [25], [27].

2.3 Gene Expression Analysis

The advances in Next generation sequencing (NGS) and DNA microarray technologies have provided the capabilities to measure the expression levels of thousands of genes during various biological processes, collected from different experimental samples and conditions [22], [27].

Gene expression refers to the process of translating information in DNA into functional products including proteins and non-coding RNA. Only a fraction of genes in a cell are expressed at a given time where a distinct set of regulators determine the expression profiles of each cell. Transcription produces what is referred to as precursor messenger RNA (pre-mRNA) which undergoes further modifications leading to mature mRNA. The formation of a malignant tumor is typically a transformation characterized by distribution of genetic information and irregular expression of multiple genes [1].

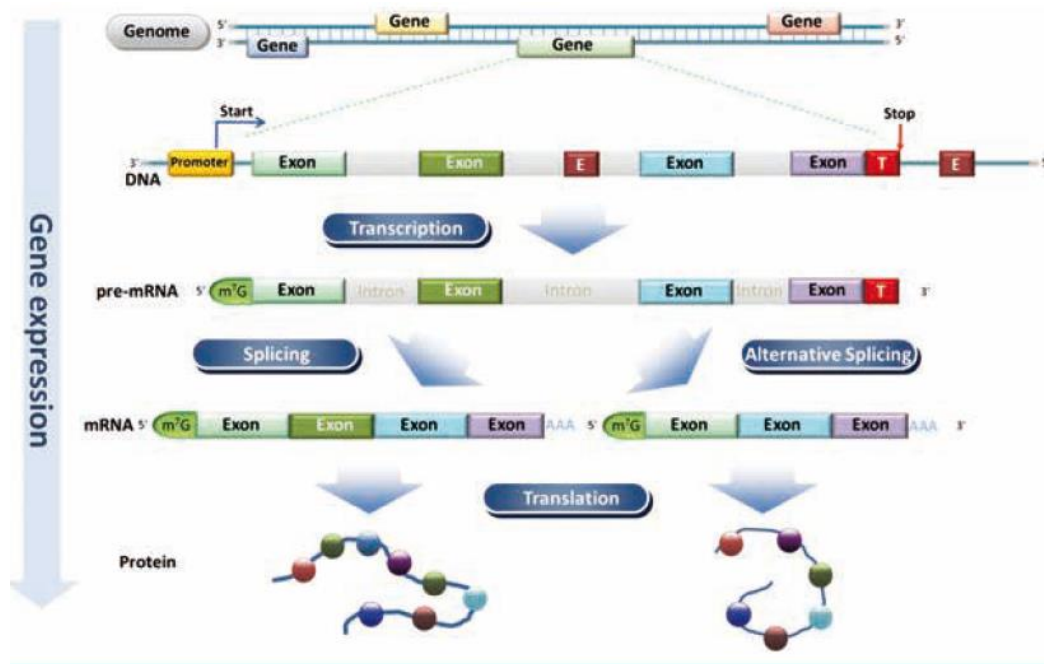


Figure 2.2 Schematic overview of protein-coding gene expression pathways [1]

During a Next Generation Sequencing experiment, DNA sequences under multiple conditions are captured for analysis where the collective data samples are commonly referred to as gene expression data [21]. The variations in conditions of the data samples could represent different time intervals in a specific biological process or they could represent different samples from different organs or tissues [25]. For example, the gene expressions could represent the DNA sequence progression of infected cancer cells at different stages, or they could represent samples from different tissues of healthy and infected patients [27].

Medical procedures for early cancer diagnosis and screening still depend heavily on clinical and histological analysis, which is the study of the microscopic anatomy of cells and tissues. But despite this common practice, there are many well-known research experiments which have proven that clinical and histological analysis are insufficient to distinguish between subclasses in several types of cancer [1], [4].

Analysis of Cancer gene expression data can have many objectives, but among the most common are class prediction and class discovery [1]. Class prediction is based on collecting mRNA samples for Cancer tumors of known classes, then using supervised learning and classification techniques to build discriminative models which can be used to learn the molecular signatures of the underlying tumor. These models can then be used to predict the tumor class of new patient samples which were previously unrecognized. Class discovery on the other hand is based on unsupervised learning to identify the molecular signature of a new subclass of a cancer tumor which was previously unknown [1], [4].

2.4 Cancer Classification using Gene Expressions

Gene expressions have been extensively used in biological research and cancer classification [5], [6], [7], [8], [9], [10], [13], [11], [17]. By collecting mRNA samples for tumors of known classes, supervised learning can be used to build discriminative models which can learn the gene patterns of the underlying disease and then be used to predict the tumor class of new patient samples which were not previously diagnosed [1]. This is considered a great achievement as there are many Microarray experiments which demonstrate how it was possible to classify and distinguish between certain cancer types using data classification even though they are clinically indistinguishable [1], [72], [73]. For example the classification between two types of Leukemia Cancer (AML and ALL) which are clinically indistinguishable [70]. Another example is the microarray experiment used to analyze a total of 78 Breast Cancer patients to develop what is known as the PAM50 Breast Cancer Intrinsic Classifier which predicts the breast cancer type out of several classes [71].

2.5 Gene Feature Selection

Current methods for cancer classification follow the approach of feature engineering and are based on applying innovative gene feature selection techniques as a prerequisite to the classifier learning process to discover a small subset of informative genes which are discriminative among the tumor being analysed [28], [29], [31]. Gene selection methods can be generally classified into filtering, wrapping and embedded methods [32], [33]. The accuracy of such a classifier depends heavily on the successful identification of these discriminative features [34], [35]. Furthermore, the same classification method might not succeed in achieving the same accuracy if applied on a tumor for a different tissue type which will most likely have a different subset of informative genes [1].

It is very common that the gene expression data produced from Next Generation Sequencing or microarray experiments will contain many data anomalies such as noise and missing values which are expected in any biological experimental procedure. Accordingly, preprocessing the gene expression data is a crucial step before attempting any analysis for disease diagnosis to ensure the quality and accuracy of the results. One of the biggest challenges in analyses of gene expression data is that only a small subset of the genes could be influencing the tumor being monitored and also it is possible that interesting features of the disease are only present in a subset of the data. Accordingly gene feature selection is an important preprocessing step. Other preprocessing tasks include data normalization and estimating missing values [32].

Gene expression data can be represented in a 2D matrix representation as shown in Figure 2.3. The matrix stores real values where each row represents the expression patterns of genes and each column represents the expression profiles of tumor samples such that the value in cell X_{ij} represents the expression level measured for gene (i) in the patient sample(j).

		N – Tumor Samples				
$G - \text{Genes}$		X_{11}	X_{12}	X_{13}	...	X_{1N}
		X_{21}	X_{22}	X_{23}	...	X_{2N}
		X_{31}				X_{3N}
	
					X_{ij}	
	
		X_{G1}	X_{G2}	X_{G3}	...	X_{GN}

Figure 2.3 Gene Expression 2D Matrix Representation (G x N)

2.5.1 Gene Filtering

Gene filtering is the process of selecting a small subset of genes which are discriminative among the full range of genes underlying the tumor being analyzed [32]. These genes are called informative genes, biomarkers or differentially expressed genes. Filter methods rely on pre-processing techniques which analyze potential overall gain of the selected features while ignoring performance of the learning algorithm [34]. Examples Principle component analysis (PCA) and Singular Value Decomposition SVD [30]. In general, there are two common filtering techniques which are widely used which are ranking methods and space search methods. In a ranking method, a scoring function is used to choose the top ranking genes. While in a space search method, the genes are selected by optimizing a certain cost function to provide a tradeoff between maximizing the information gain and minimizing the redundancy among the selected genes [32], [33].

2.5.2 Gene Wrapping

A drawback in filtering the gene expression data before building the classifier is that it produces a dataset where the genes might have a high level of correlation within the same class. This correlation might be resulting from shared upstream signaling of molecules which might result in misclassification. [34]. The process of Wrapping as opposed to filtering, attempts to solve this problem by embedding the feature selection step directly into the classifier. Gene wrapping relies on selecting a subset of features according to the performance gain they provide to the learning algorithm [32], [33].

2.5.3 Gene Filtering using Ranking

In a ranking method, a scoring function is used to choose the top ranking genes. The following is a summary of the steps used to filter the genes using ranking [37]:

- 1) Define a scoring function to measure the expression level differences between the various gene samples and rank the features based on the obtained scores.
- 2) Estimate the statistical significance of the obtained scores.
- 3) Select the top ranking genes which are statistically significant.
- 4) Validate the subset of selected genes.

Score Functions

There are a wide variety of ranking score functions available in the literature which is summarized in the tables below [33]. The first table describes the notations used for the definitions. The ranking score functions can be divided into the following groups :

- Rank Score Functions
- T-Test Functions
- Bayesian Functions
- Information Theory Functions
- Functions based on Probability Density Function (PDF-Based)
- Correlation Gene Class Label Functions

Estimating Statistical Significance

Calculating a score function is not enough for gene selection, but the statistical significance has to be estimated as a form of probability measure that a good score ranking has not been obtained by chance [33]. Statistical significance tests typically consist of running permutations of multiple tests which are identical with the distinction that the features or the class label can be chosen differently on each test [37].

The following is a summary of the most widely used score functions for Gene Filtering Ranking methods and their corresponding notations [32], [33], [37].

$X^{m \times n}$	Data set with m genes and n samples
$X_1^{m \times n_1}, X_2^{m \times n_2}$	subsets of X denoting samples from two different populations, where $n_1 + n_2 = n$
x, x_1, x_2	single gene expression across all samples, across samples in X_1 respectively X_2
$\bar{x}, \bar{x}_1, \bar{x}_2$	mean value of x, x_1 and x_2
$\sigma_x, \sigma_{x_1}, \sigma_{x_2}$	standard deviation of x, x_1 and x_2
P_x, P_{x_1}, P_{x_2}	probability density function of x, x_1 and x_2

Rank Score Family

Name	Metric
Wilcoxon rank sum	$S = \sum_{j=1}^k R_j, k = \min(n_1, n_2)$
Rank product	$S = (\prod_{j=1}^n R_j)^{1/n}$

t-Test Family

Name	Metric
Z-score	$S = \frac{\bar{x}_1 - \bar{x}_2}{\sigma_x}$
t-test	$S = \frac{\bar{x}_1 - \bar{x}_2}{\sigma_{x_1} + \sigma_{x_2}}$
Welch t-test	$S = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{\sigma_{x_1}^2}{n_1} + \frac{\sigma_{x_2}^2}{n_2}}}$
Modified t-test	$S = \frac{\bar{x}_1 - \bar{x}_2}{\sigma_x + \sigma_0}$ σ_0 - small positive constant

Bayesian Family

Name	Metric
Bayesian t-test	$S = \frac{\bar{x}_1 - \bar{x}_2}{\sigma_p}, \sigma_p^2 = \frac{\nu_0 \sigma_0^2 + (n-1) \sigma_x^2}{\nu_0 + n - 2}$ ν_0, σ_0 - prior degrees of freedom/variance
Regularized t-test	$S = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{\sigma_{x_1}^2}{n_1} + \frac{\sigma_{x_2}^2}{n_2}}}$ $\tilde{\sigma}_{x_1, x_2}^2 = \frac{\nu_0 \sigma_0^2 + (n_{1,2} - 1) \sigma_{x_1, x_2}^2}{\nu_0 + n_{1,2} - 2}$ ν_0, σ_0 - prior degrees of freedom/variance
Moderated t-statistics	$S = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\tilde{\sigma}^2 \left(\frac{1}{n_1} + \frac{1}{n_2} \right)}}, \tilde{\sigma}^2 = \frac{ds^2 + d_0 \sigma_0^2}{d + d_0},$ $s^2 = \frac{(n_1 - 1) \sigma_{x_1}^2 + (n_2 - 1) \sigma_{x_2}^2}{(n_1 - 1) + (n_2 - 2)}$ $d = n_1 + n_2 - 2$ d_0 and σ_0 are unknown and must be estimated from the data
B-statistics	$B = \log A \left[\frac{b + \sigma_x + (\bar{x}_1 - \bar{x}_2)^2}{b + \sigma_x + \frac{(\bar{x}_1 - \bar{x}_2)^2}{1 + nh}} \right]^{\nu + \frac{n}{2}}$ $A = \frac{p}{1-p} \frac{1}{\sqrt{1+nh}}$ b and ν - hyperparameters in the inverse gamma prior for the variance h - hyperparameters in the normal prior of the nonzero means p - fixed to sensible values (0.01 or 0.001)

CDF_{x_1}, CDF_{x_2}	cumulative density function of x_1 and x_2
c	class label feature or target annotation
c_1, c_2	labels corresponding to X_1, X_2 respectively
$R_{i,j}$	rank of gene i in the j -th sample
S	relevance index or score associated to a gene
$\Omega_m, \Omega_s, \Omega_s $	the whole set of genes, a subset of genes from Ω_m respectively the number of genes in Ω_s

Fold-Change Family

Name	Metric
Fold-change ratio	$S = \frac{\bar{x}_1}{\bar{x}_2}$
Fold-change difference	$S = \bar{x}_1 - \bar{x}_2$

Information Theory-Based Scoring Functions

Name	Metric
Info gain	$S = \text{Info}(X) - \text{Info}_x(X)$ $\text{Info}(X) = - \sum_{i=1}^k P(c_i, X) \times \log(P(c_i, X))$ $\text{Info}_x(X) = - \sum_{i=1}^v \frac{ V_i }{ X } \times \log(V_i)$ k - number of classes v - number of individual values of a gene x V_i - the set of instances whose values in gene x equal x_i $ V_i $ - number of samples in $V_i, X = n$
Mutual info	$S = \sum_{i=1} \log \left[\frac{P_{x_i}}{P_{x_1, i} P_{x_2, i}} \right]$

pdf-Based Scoring Functions

Name	Metric
K-S test	$S = \sup(CDF_{x_1} - CDF_{x_2})$
KL divergence	$S = \sum_{i=1} P_{x_1, i} \log \frac{P_{x_1, i}}{P_{x_2, i}}$
Bhattacharyya distance	$S = - \ln \sum_i \sqrt{P_{x_1, i} P_{x_2, i}}$

Correlation Gene-Class Label Family

Name	Metric
PCCs	$S = \frac{\sum_{i=1}^n (x_i - \bar{x})(c_i - \bar{c})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (c_i - \bar{c})^2}}$
2-class	$S = \sum_{i \in c_1} \sum_{j \in c_2} h(x_{1, i} - x_{2, j})$
KRCCs	where $h(x) = \begin{cases} 0, & \text{if } x \leq 0 \\ 1, & \text{if } x > 0 \end{cases}$

Figure 2.4 Score Functions for Gene Filtering Ranking Methods [32], [33], [37]

2.6 Cancer Classification Methods using Gene Expressions

Cancer classification is based on collecting samples for tumors of known classes and using supervised learning to build discriminative models which can learn the gene patterns of the underlying disease and then be used to predict the tumor class of new patient samples which were not previously diagnosed [1]. Current methods for cancer classification follow the approach of feature engineering and are based on applying innovative gene feature selection techniques as a prerequisite to the classifier learning process to discover a small subset of informative genes which are discriminative among the tumor being analysed [28], [29], [31]. The accuracy of such a classifier depends heavily on the successful identification of these discriminative features [34], [35].

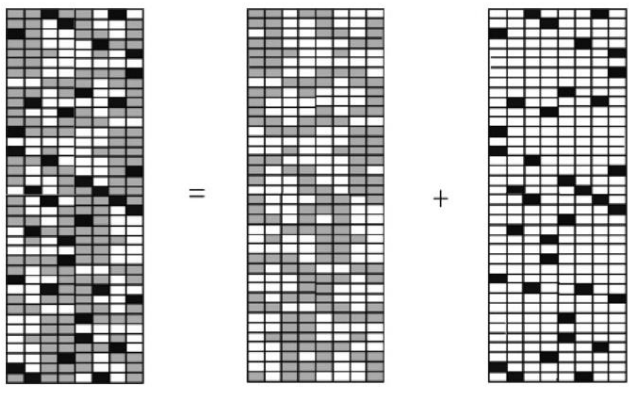
Substantial work has been done for cancer classification by performing gene feature selection and building on traditional machine learning methods such as Support Vector Machines [15], [18], [30], Random Forests [14], Decision Trees [16], AdaBoost [11], K-Nearest Neighbor [15] and Genetic algorithms [9], [11]. Many other techniques which combine gene feature selection and classification have also been proposed, for example: a hybrid method which integrates genetic programming and particle swarm optimization was used to build a scale-free complex network classifier using an ensemble of different gene feature sets [8]. A self-training subspace clustering algorithm was proposed by first applying a low-rank representation to extract discriminative features from gene expressions [13]. A deep neural forest model was used with a combination of fisher ratio and neighborhood rough set for dimensionality reduction of gene expressions [12]. An ensemble classifier was developed using a combination of k-means clustering, t-test, self-organizing maps and hierarchical clustering [10]. A classifier was developed using a multilayer recursive feature elimination method based on an embedded integer-coded genetic algorithm [9]. A gene expression graph structure was proposed for using the weight of graph edges to filter and determine significance of genes before classification [17]. A one-class logistic regression machine learning algorithm was used to identify stemness features extracted from transcriptomic and epigenetic data from cancer tumors to reveal clinical insight and potential drug targets for anti-cancer therapies [6].

The following sections present a survey of the state-of-the-art cancer classification methods using gene expression data.

2.6.1 Support Vector Machines (SVM)

Many studies have been proposed for Cancer classification using Support Vector Machines (SVM) and gene feature selection as a prerequisite to the learning process [15], [18], [30].

One of the proposed examples is based on robust principle component analysis (RPCA) and SVM to classify tumor samples of gene expressions [18]. First RPCA is used to extract the characteristic genes from gene expression data. In the second stage, Linear Discriminant Analysis (LDA) is then used to refine the subset of characteristic genes. Finally, SVM is then applied to classify the tumor samples of gene expressions based on the identified features [18].



$$\begin{aligned}
 &\text{minimize } \|\mathbf{A}\|_* + \lambda \|\mathbf{S}\|_1 \\
 &\text{subject to } \mathbf{D} = \mathbf{A} + \mathbf{S}, \\
 &L(\mathbf{A}, \mathbf{S}, \Phi, \mu) = \|\mathbf{A}\|_* + \lambda \|\mathbf{S}\|_1 \\
 &\quad + \langle \Phi, \mathbf{D} - \mathbf{A} - \mathbf{S} \rangle + \frac{\rho}{2} \|\mathbf{D} - \mathbf{A} - \mathbf{S}\|_F^2
 \end{aligned}$$

$$\begin{aligned}
 \mathbf{w} &= \arg \max_{\mathbf{w}} \frac{\mathbf{w}^T \mathbf{S}_b \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w \mathbf{w}} \\
 \mathbf{S}_b &= \sum_{k=1}^c m_k (\mu^{(k)} - \mu)(\mu^{(k)} - \mu)^T \\
 \mathbf{S}_w &= \sum_{k=1}^c \left[\sum_{i=1}^{m_k} (x_i^{(k)} - \mu^{(k)})(x_i^{(k)} - \mu^{(k)})^T \right]
 \end{aligned}$$

Figure 2.5 RPCA and LDA to extract characteristic genes from gene expressions before applying SVM. The gene expression matrices D, A, and S represent the observation matrix, low-rank matrix and sparse perturbation signals to decompose the gene expression data [18]

2.6.2 K-Nearest Neighbors

A comparative study was performed for applying different feature selection methods on the classification performance of cancer using DNA microarrays of leukemia, prostate and colon cancer data [15]. Feature selection of gene expressions was applied using the methods of Fisher, T-Statistics, SNR and ReliefF. Classification was then performed using K-Nearest Neighbors and Support vector machines. The study showed that the combination between SNR feature selection and SVM produced the highest accuracy for cancer classification [15].

2.6.3 Fuzzy Decision Trees (DT)

A Fuzzy decision tree algorithm was proposed for the classification of gene expressions since they have shown to outperform classical decision tree algorithms [16]. Classical decision trees have shown some disadvantages in that their performance tends to deteriorate with the increase of features and emergence of complex interactions as in gene expression data. Since most decision trees depend on dividing the search space into mutually exclusive regions, the resulting tree must include several copies of the same subtree to accurately represent complex data like gene expressions. This greedy approach is prone to overfitting on the training set in addition to irrelevant features and noise. On the other hand, Fuzzy decision trees do not require assigning a data instance with a single branch and can simultaneously assign more branches to the same instance with a gradual certainty. Using this approach, Fuzzy decision trees retain the symbolic tree structure and are able to represent concepts by producing continuous classification outputs with gradual transitions between classes [16].

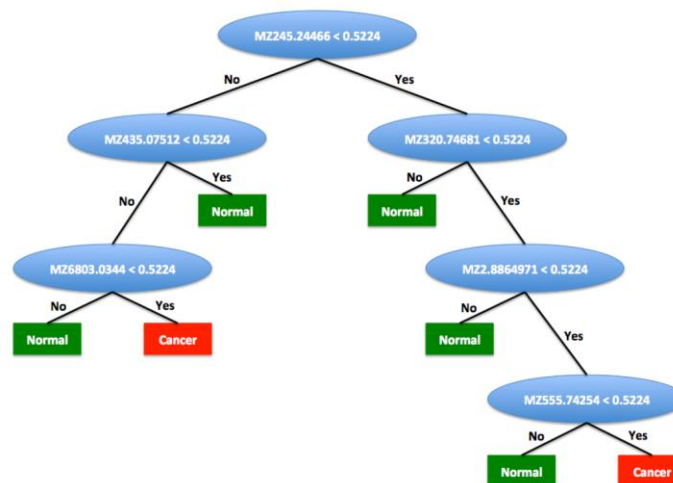


Figure 2.6 Fuzzy Decision Tree Classifier for Ovarian Cancer Gene Expressions [16]

2.6.4 AdaBoost

A hybrid ensemble algorithm combining AdaBoost and genetic algorithm (GA) was proposed for cancer classification with gene expression data [11]. A decision group is proposed to improve the diversity of base classifiers in the ensemble system and GA is used to optimize the weight of Adaboost's base classifier. In a traditional Adaboost algorithm, a single classifier is used as the base classifier and cannot be changed after selection. The introduction of a decision group as the base classifier of the Adaboost algorithm was used to improve the diversity of the base classifiers [11].

2.6.5 Particle Swarm Optimization

A hybrid method which integrates genetic programming and particle swarm optimization was used to build a scale-free complex network classifier using an ensemble of different gene feature sets [8]. A Complex Network (CN) classifier was used to implement the classification task. A Complex Network is different from a Neural Network in terms of topological structure. A CN has an uneven distribution of nodes, while a NN has an even distribution. CN models are used to simulate structural properties of many real-world networks like social networks and bibliographical index networks. An algorithm was used to initialize the structure, which allowed input variables to be selected over layered connections and different activation functions for different nodes. Then a hybrid method integrated Genetic Programming and Particle Swarm Optimization was used to identify an optimal structure with the parameters encoded in the classifier. The ensemble classifiers were constructed using different feature sets including Pearson's correlation, Spearman's correlation, euclidean distance, Cosine coefficient, and the Fisher-ratio [8].

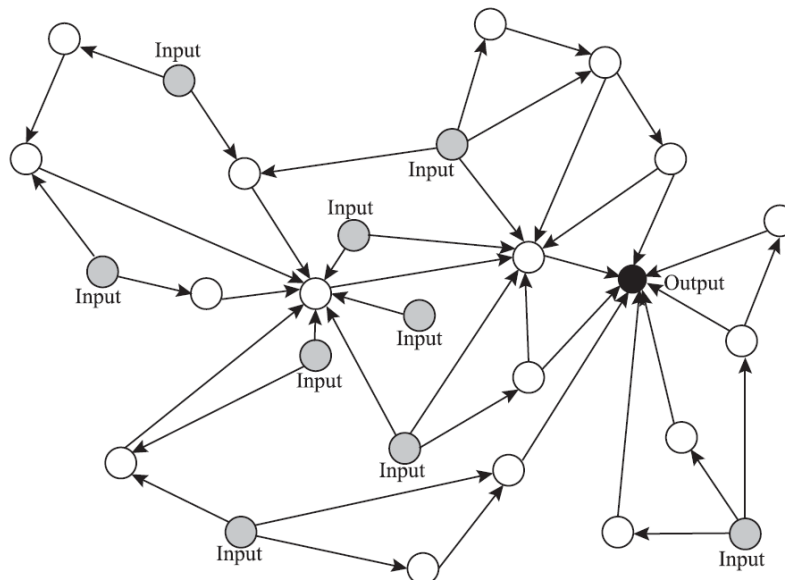


Figure 2.7 Topology of a Cancer Classifier implementing a scale-free Complex Network [8]

2.6.6 Random Forests (RF)

A method was proposed using Random Forest for cancer classification of miRNA gene expression data [14]. The method was used to overcome challenges in existing techniques caused by the extremely low miRNA count in body fluids and also problems related to cross contamination between cells and exosomes in sample preparation steps. The proposed system was able to successfully identify miRNA markers responsible for classification of cancer [14].

2.6.7 Deep Neural Forest Models (DFN)

A Deep Neural Forest (DFN) model was proposed for cancer classification with a combination of fisher ratio and neighborhood rough set for dimensionality reduction of gene expressions [12]. The motivation in using a DFN is to transform a multi-class classification problem into many binary classification problems in each forest. The cascade structure of the DFN is used to deepen the traditional Flexible Neural Tree (FNT) model so that the depth of the model is increased without introducing additional parameters. FNT is a special neural network with the advantage of automatic optimization of structure and parameters. Gene feature selection was first performed using a fisher ratio in combination with neighborhood rough set to select the most informative genes among the gene expression data. The fisher ratio was used to eliminate invalid genes and then neighborhood rough set is applied to reduce redundant genes. The fisher ratio method can effectively deal with noise in the gene expression data as it filters the noisy genes according to its contribution to classification. The neighborhood rough set has the characteristics of not requiring discretization of continuous data and avoids information loss caused by data discretization, which can eliminate redundant genes [12].

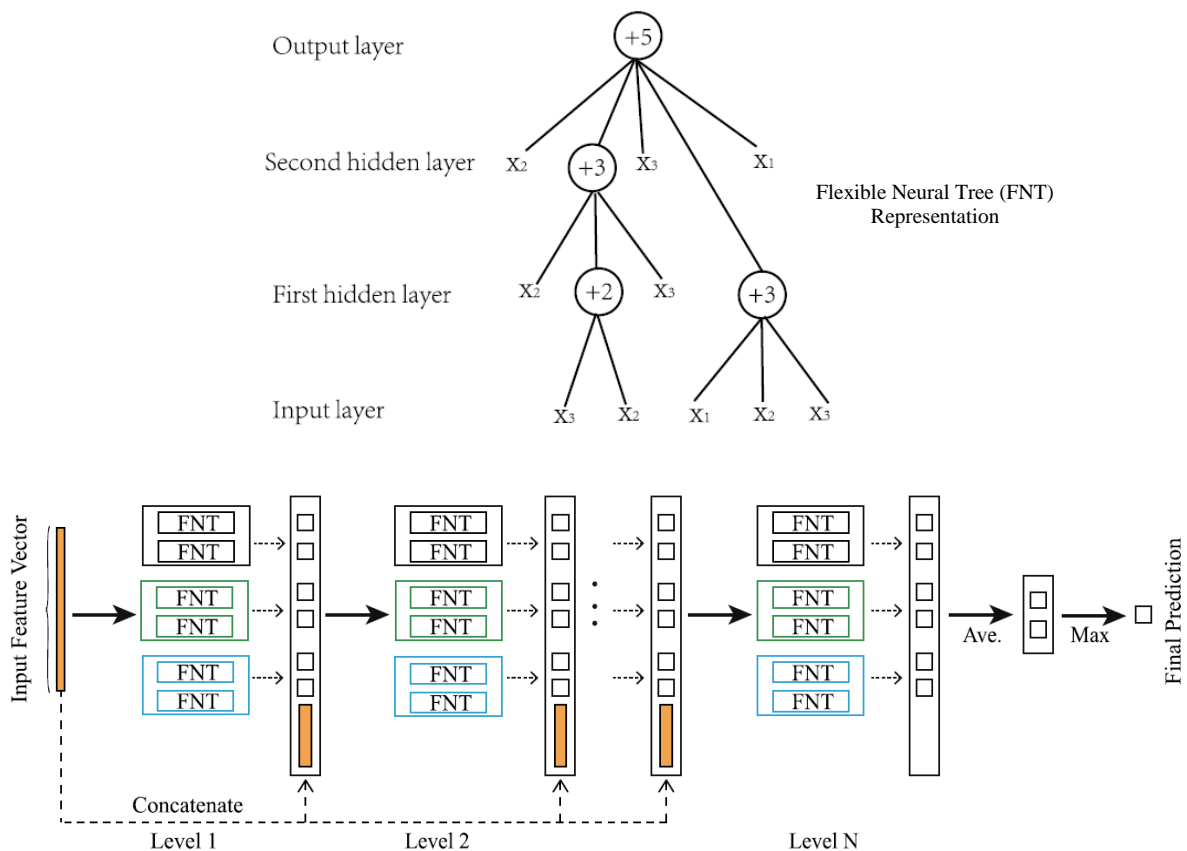


Figure 2.8 Deep Neural Forest Structure used for Cancer Classification [12]

2.6.8 Self-training Subspace Clustering

A self-training subspace clustering algorithm under low-rank representation (SSC-LRR) was proposed for cancer classification of gene expressions [13]. First, a Low-rank representation (LRR) is applied to extract discriminative features from the high-dimensional gene expression data. The self-training subspace clustering (SSC) method is then used to generate the cancer classification predictions. The advantage of combining these two methods is that the Low-rank representation is able to perform subspace segmentation which can reduce the dimension of the gene expression data, and then the enhanced semi-supervised self-training subspace clustering algorithm can effectively utilize both the labeled and unlabeled data. To analyse the results, the study performed a decomposition of the gene expression data matrix into a low-rank representation matrix and a sparse matrix and then visualized the results. It was shown that cancer samples belonging to the same class often have the same subspace structure. This means that the low-rank representation can unveil the intrinsic structure of data much better than the original data matrix and therefore the low-rank representation can provide more useful discriminative information leading to a better classification performance. From a biological point of view, different types of cancers are often associated with some specific genes and therefore the corresponding gene expression data may fall into specific feature subspaces, which can be unveiled by using LRR [13]. The proposed SSC-LRR method was tested on two separate cancer benchmark datasets in control with four state-of-the-art classification methods. The method showed that several genes (RNF114, HLA-DRB5, USP9Y, and PTPN20) were identified as new cancer identifiers that deserve further clinical investigation [13].

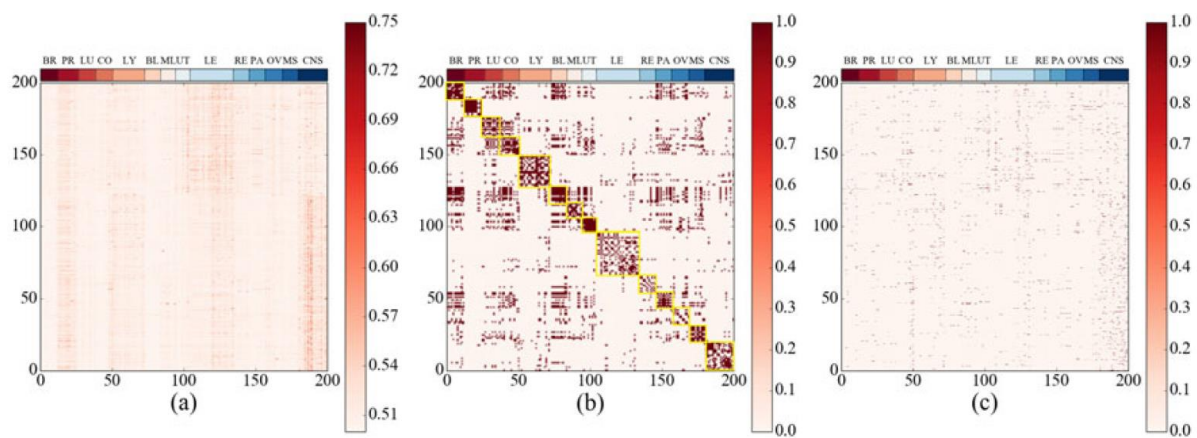


Figure 2.9 Illustration of Cancer Classification of Gene Expressions using Self-training Subspace Clustering and Low-Rank representation (SSC-LRR). (a) shows the original gene expression data matrix, (b) shows the decomposed low-rank representation and (c) shows the decomposed sparse representation [13].

2.6.9 One-Class Logistic Regression

A one-class logistic regression machine learning algorithm was used to identify stemness features extracted from transcriptomic and epigenetic data from cancer tumors to reveal clinical insight and potential drug targets for anti-cancer therapies [6]. Stemness is defined as the potential for self-renewal and differentiation from the cell of origin. Cancer progression involves gradual loss of a differentiated phenotype and acquisition of progenitor-like, stem-cell-like features [4]. The proposed study was based on an integrated analysis of cancer stemness in human tumors of different cancer types including gene expression data of mRNA and miRNA. By applying one-class logistic regression to molecular datasets from normal stem cells and their progeny, the method developed two different molecular metrics of stemness and then used them to classify epigenomic and transcriptomic features of the cancer tumors [6].

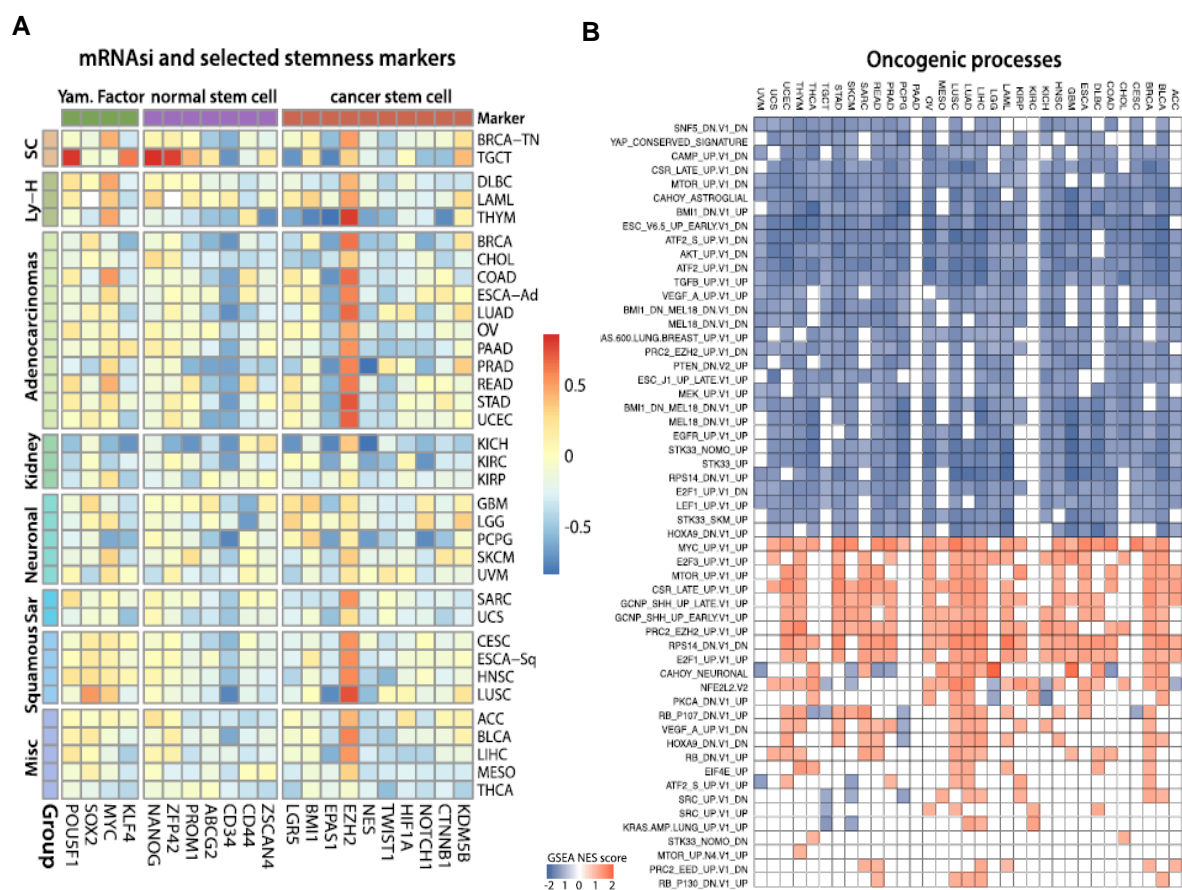


Figure 2.10 Results of One-class Logistic Regression used to identify biological processes associated with Cancer Stemness. (A) Correlation between mRNAsi and mRNA expression for published hallmarks of stemness. (B) Correlation between mRNAsi and selected oncogenic processes [6].

2.6.10 Multilayer Recursive Feature Elimination

A Multilayer Recursive Feature Elimination (MGRFE) method was proposed for cancer classification based on an embedded integer-coded genetic algorithm [11]. The feature elimination was aimed at selecting the gene combination with minimal size and maximal information. The method uses the filtering algorithms t-test and Maximal Information Coefficient (MIC) to reduce the feature range and generate a candidate feature set. MGRFE combines the advantages of both evolution calculation of genetic algorithms and the explicit Recursive Feature Elimination (RFE) to achieve the minimum discriminative gene subset with optimal classification ability. The experiments of the study showed that MGRFE outperforms state-of-the-art feature selection algorithms with better cancer classification accuracy and a smaller selected gene number on 19 benchmark microarray datasets including multiclass and imbalanced datasets [11].

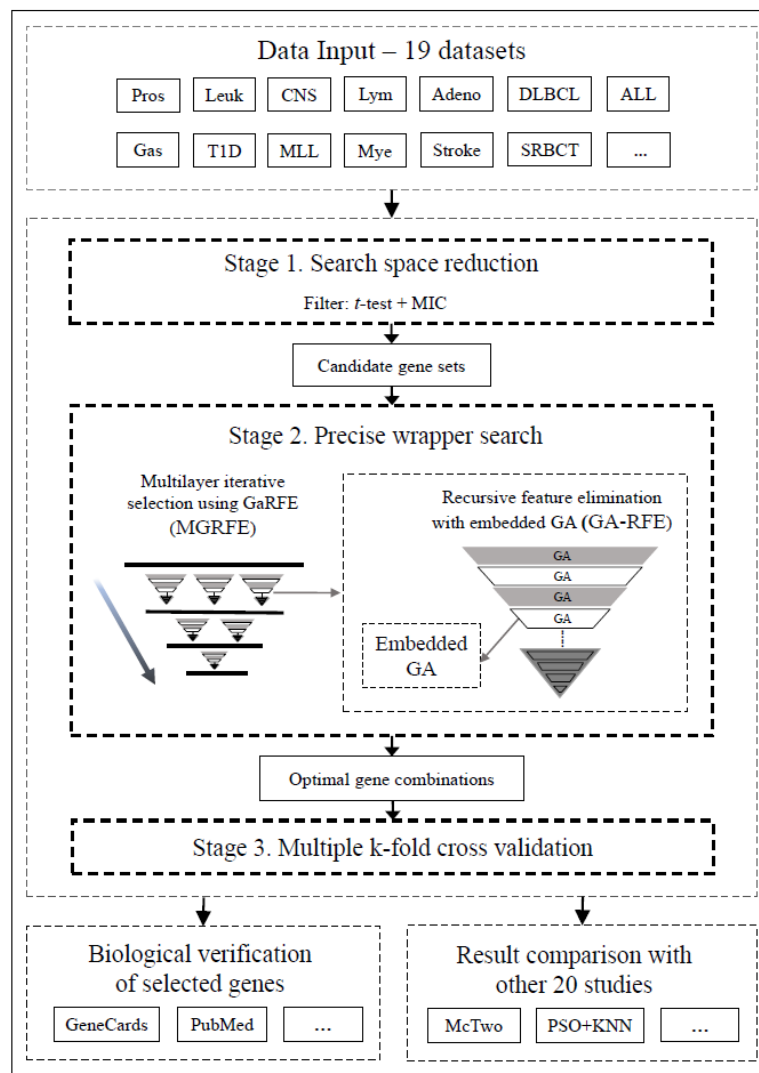


Figure 2.11 Cancer Classification using Multilayer Recursive Feature Elimination (MGRFE) based on an embedded integer-coded Genetic Algorithm (GA) [11].

2.6.11 Graph Structure Algorithms

A gene expression graph structure was proposed for cancer classification by using the weight of graph edges to filter and determine the significance of genes before classification [17]. The informative genes were selected by filtering the weight values between genes such that greater weights indicate a stronger relationship between two genes. The method was also able to detect out-of-class samples that do not belong to any trained class.

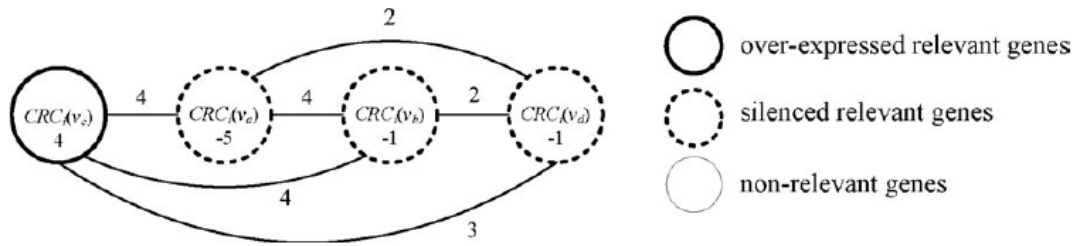


Figure 2.12 Cancer Classification using a Weighted Gene Expression Graph Structure [17].

2.6.12 Genetic Algorithms (GA)

Genetic algorithms (GA) have been frequently used for cancer classification of gene expressions by combining gene feature selection and other machine learning methods. For example, an embedded integer-coded genetic algorithm was used for cancer classification of 19 benchmark microarray datasets [9]. The approach relied on first applying a multilayer recursive feature elimination method based on the embedded integer-coded genetic algorithm with the aim of selecting the gene combination with minimal size and maximal information. Another example is the use of a hybrid ensemble algorithm combining genetic algorithms and AdaBoost for cancer classification with gene expression data [11]. A hybrid method was also used which integrates genetic programming and particle swarm optimization to build a scale-free complex network classifier using an ensemble of different gene feature sets [8].

2.6.13 Ensemble Classifiers

One of the common approaches in classification is to use an Ensemble of multiple classifiers to improve Classification accuracy. An Ensemble classifier was developed for classification of Lung Adenocarcinoma cancer (LUAD) into molecular subtypes using a combination of k-means clustering, t-test, Self-organizing Maps (SOM) and Hierarchical Clustering [10]. The method determined 24 differentially expressed genes which could be used as therapeutic targets, and five genes (RTKN2, ADAM6, SPINK1, COL3A1, and COL1A2) which could be potential novel markers for Lung cancer (LUAD).

2.7 Deep Learning

Traditional machine learning techniques have shown limitations in processing high dimensional data [59]. Recent neuroscience findings have provided additional insight into the principles governing information representation in the brain. The discovery motivated the emergence of deep machine learning which has since revolutionized the capabilities of processing high dimensional data [59]. Deep learning methods rely on building complex multi-layer network architectures capable of processing huge amounts of high dimensional data with minimal preprocessing requirements [46], [58]. Deep learning leverages spatial relationships among data to reduce the number of dimensions to be learned which dramatically improves the learning process in comparison to traditional machine learning methods [59], [67].

2.7.1 Learning using Deep Multilayer Architectures

Traditional shallow architectures such as 2-layer neural networks, SVMs and kernel machines have been shown to be universal learning machines. But deep multilayer architectures have the capability of representing more complex functions [59]. The approach using Deep Learning is through building architectures with multiple layers each with a non-linear function. Each layer transforms the input to increase the level of accuracy and invariance of the selected features. As the Deep Learning architectures increase in depth and layers, the learning procedure is capable of representing complex functions which are very sensitive to the slightest details in the input objects and which are also insensitive to any irrelevant variations [59], [67].

2.7.2 Feature Extraction using Representation Learning

Deep learning is commonly referred to as Representation Learning which is a technique based on using raw data as input with no feature extraction as a prerequisite. Deep learning relies on building architectures with multiple levels of representation by combining non-linear building blocks. Each level in the architecture transforms the input into a representation at a higher more abstract level which provides the capability of learning complex non-linear functions [59].

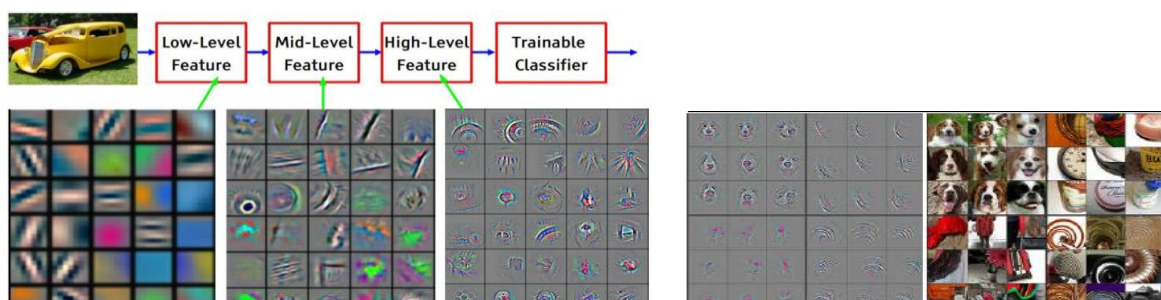


Figure 2.13 Visualization of Extracted Features from Deep Learning Networks [65]

2.7.3 Training Deep Architectures using Gradient Descent & Backpropagation

Supervised learning is used to train deep multilayer architectures. Training is performed by collecting a large amount of labelled data and presenting it to the network to produce an output score for each labelled category. By defining the appropriate objective function that measures the error between the network output and the desired output, we can adjust the network weights using Gradient descent optimization to reduce the classification error [59], [66].

Gradient descent optimization can be illustrated by considering the cost function, averaged over all the training data, as a very high dimensional landscape full of hills in the network weight space. The negative gradients represent the direction of steepest descent in this landscape which can be used to iteratively determine the local minimum [62].

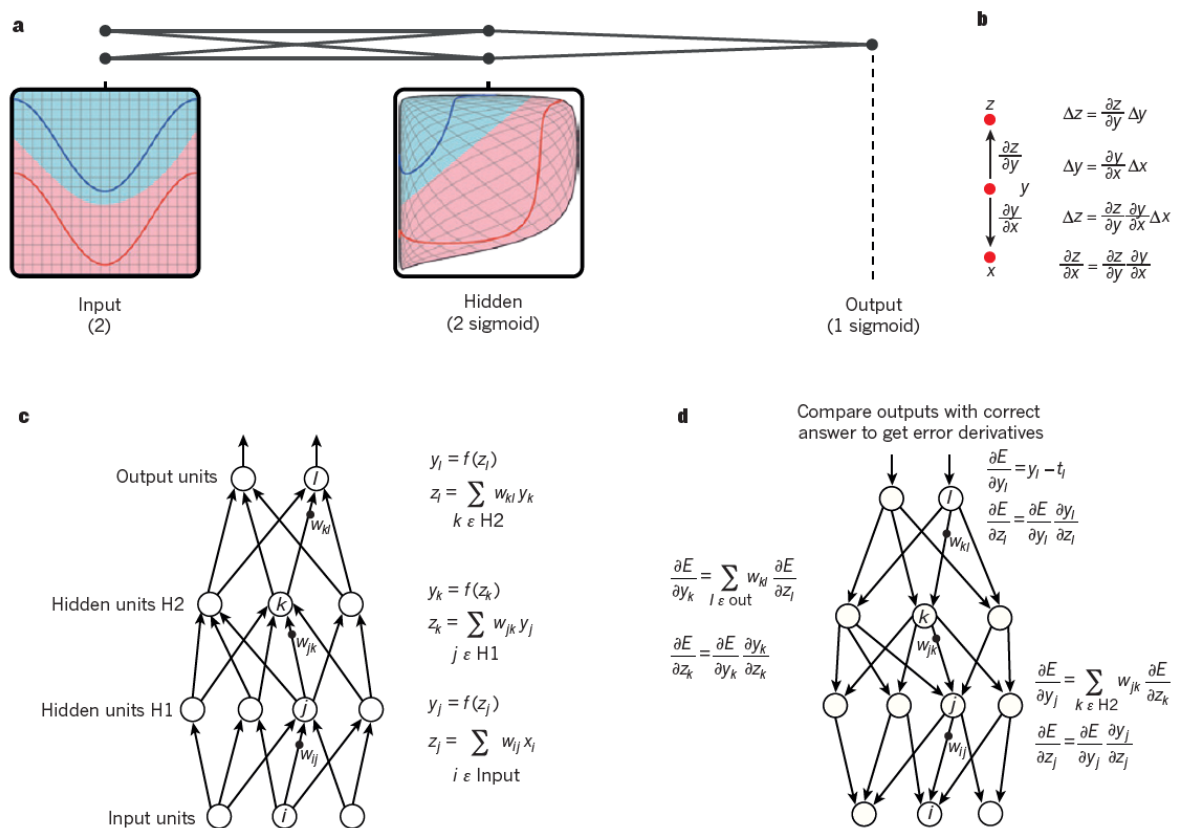


Figure 2.14 Training Multilayer Neural Networks using Backpropagation [59]

By training deep multilayer networks using Gradient descent and Backpropagation, the network learns to map an input of fixed size, such as an image, to an output which could represent probability scores of the classification categories [66]. A non-linear activation function is applied before passing the weighted sum of the inputs from one layer to the next. The hidden layers are considered to be performing a non-linear transformation of the input so that the classification of the output categories can become linearly separable [46].

Deep network architectures can be trained by means of stochastic gradient descent using backpropagation by application of the chain rule for derivatives. To adjust the network weights, we need to calculate the gradient of the error function with respect to all the weight parameters in the network. Backpropagation calculations can be used to propagate the gradients from the output layer back to the input by passing through the multiple layers [67].

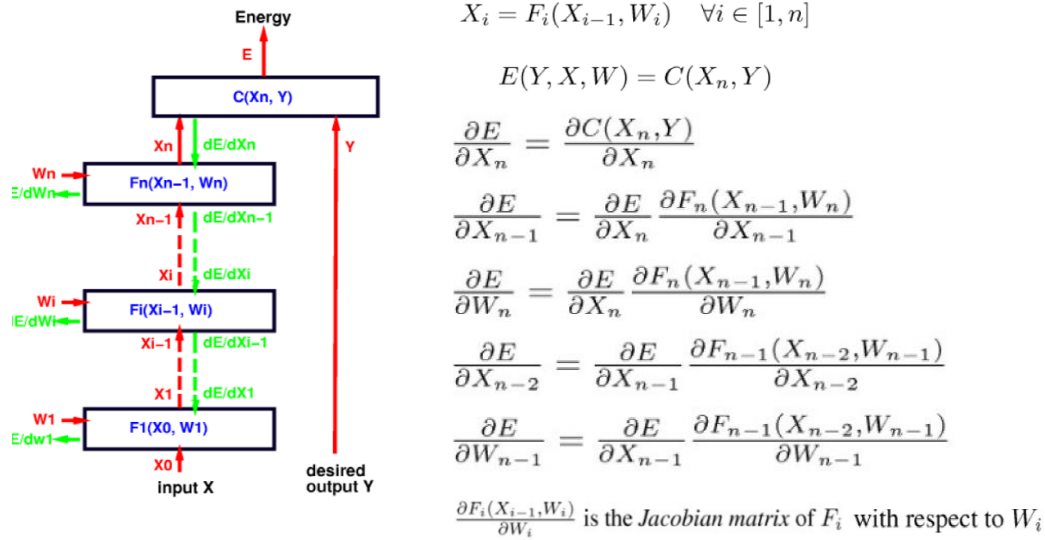


Figure 2.15 Computing Gradients for Deep Networks using Backpropagation [67]

2.8 Convolutional Neural Networks

2.8.1 CNN Overview

Convolutional Neural Networks (CNNs) are similar to traditional Neural Networks in having neurons with learnable weights and biases but differ greatly in the architecture and connectivity between the various layers. CNNs are made of multiple layers where each layer is arranged in the form of a 3D volume of neurons that has a specific width, depth and height. Each layer transforms the input 3D volume to an output 3D volume using a non-linear transformation function. The notion of depth is different from the number of layers of the network which was typically referred to as depth in traditional neural networks, but the depth in this context refers to the depth of the activation volume of neurons in a particular layer [59]. CNNs also differ in that the neurons in a particular layer will only be connected to a small region in the previous layer instead of the traditional fully connected networks, [66]. The following sections provide a survey of some of the current state-of-the-art convolutional neural networks used in computer vision and natural language processing applications.

2.8.2 (ALEX-Net) ImageNet Classification with Deep Networks

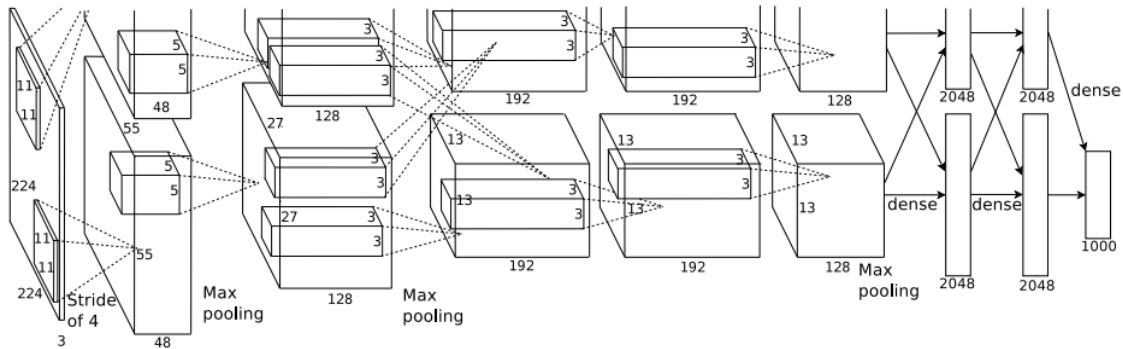


Figure 2.16 ALEX-NET Convolutional Neural Network [66]

ALEX-Net was among the first Convolutional Neural Networks that drew great attention to the capabilities of deep learning and managed to outperform some of the existing classification benchmarks with a relatively high margin [66]. It won the first place in the ImageNet 2012 competition. It was used to classify 1.2 million high resolution images covering 1000 different classes. The network had 5 convolutional layers followed by max pooling layers in some of them and then followed by 3 fully connected layers and a final 1000-way softmax for classification. The network had 60 million parameters and 650,000 neurons. A dropout regularization was used on the fully connected layers to reduce over fitting [66].



Figure 2.18 Examples of Convolutional Kernels learned by ALEX-NET in the first layer [66]

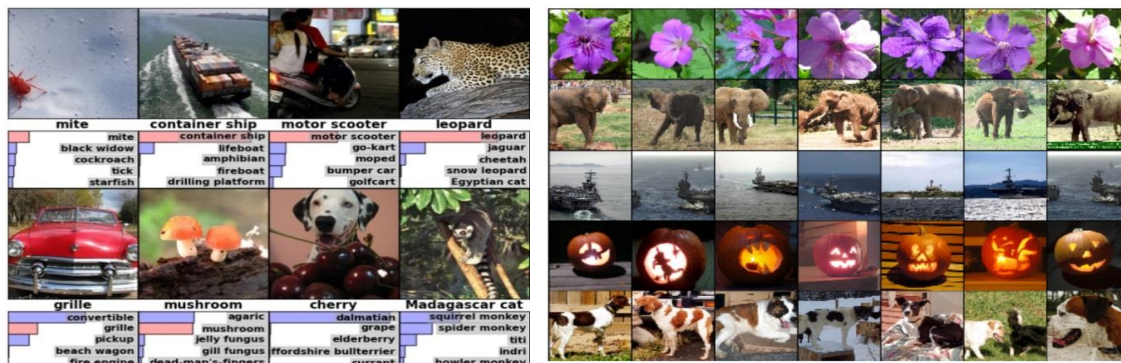


Figure 2.17 Classification of ImageNet 2012 Images using ALEX-NET [66]

2.8.3 (ZF-Net) Visualizing and Understanding Convolutional Networks

The ZF-Net network architecture provided a visualization technique to gain more insight on the functions of the intermediate feature layers of a Convolutional Neural Network and how it performs its classification [65]. This insight was used to further enhance the design of the network and enhance its classification performance and was able to achieve better performance in the ImageNet competition compared to ALEXNET. By using what is referred to as a De-convolutional Neural Network, it was possible to visualize the intermediate feature maps extracted by the intermediate layers of a Convolutional Neural Network which has provided more insight on the functions of the intermediate layers and their role in feature extraction [65].

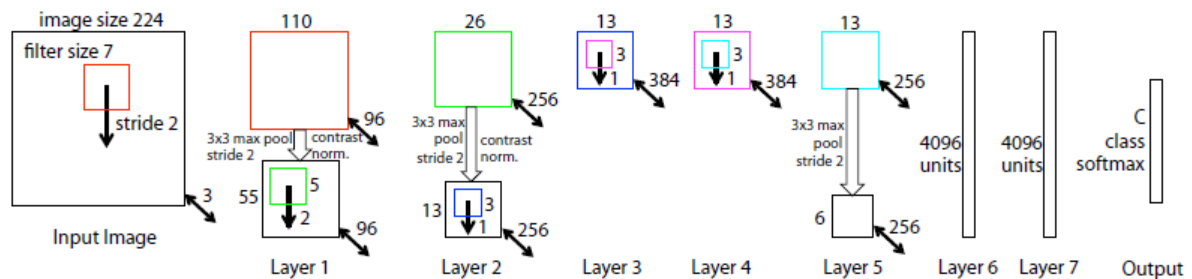


Figure 2.19 ZF-NET Convolutional Neural Network [65].

Using the ZF-NET, it was possible to visualize the top activation for any feature map projected back to the image pixel space. This visualization made it possible to reveal the different structures that excite the activation map and demonstrated how it is invariant to any input deformations. The visualization of features also demonstrated through experimental trials that the features extracted by CNNs are not just random patterns, but they have significant interpretations in how the class discrimination is performed. It also demonstrates how the network is able to extract features with desirable properties such as compositionality and increasing variance as the data is moved deeper into the network layers [65].

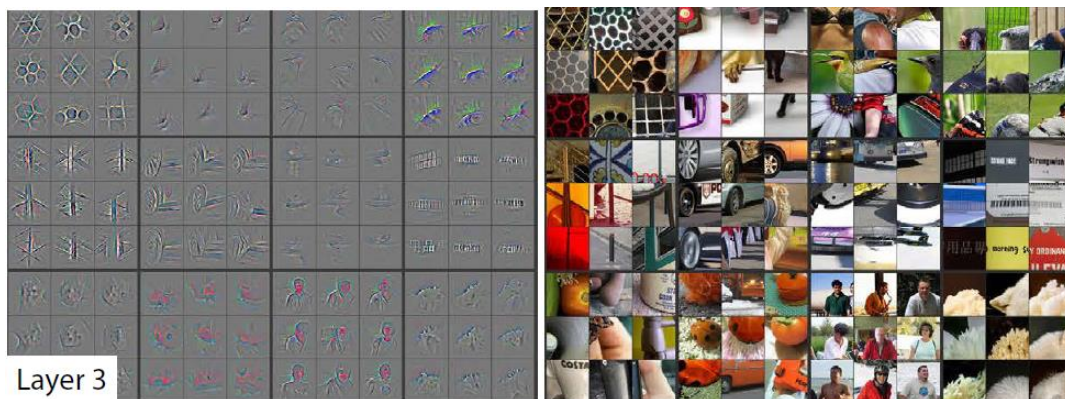


Figure 2.20 Visualization of layer 3 features for a fully trained ZF-NET network [65].

2.8.4 (VGG-Net) Very Deep Convolutional Networks for Image Recognition

The VGG-Net network attempts to further improve the architecture of CNNs by studying the variations of design in terms of the depth of the network. Experiments are performed to gradually increase the depth of the network by adding more convolutional layers while fixing other network parameters and using a very small 3x3 filter [64]. Variations in CNN depth configuration included networks starting from 11 weight layers (8 Conv. and 3 FC layers) up-to networks with 19 weight layers (16 Conv. And 3 FC layers). At the same time variations are applied to the width of the network by changing the number of filters used from 64 in the first layer and up-to 512. Changing the number of filters determines the volume width or number of channels of the stacked activation maps after convolution. These variations in network depth demonstrated that the representation depth is beneficial for the accuracy of the classification and that state-of-the-art performance can be achieved by a conventional CNN architecture with substantially increased depth [64].

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224×224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 2.21 VGG-NET Convolutional Neural Network configurations [64]

2.8.5 Inception Model Architectures

It has been shown that if a large sparse Deep Neural Network architecture can be used to represent the probability distribution of a dataset, then the optimal network architecture can be constructed on a layer by layer basis by analysis of the correlation statistics of the activations of the last layer and clustering neurons with highly correlated outputs [63].

The Inception model architecture attempts to find an optimal approximation for a local sparse structure that is covered by readily available dense components. Construction proceeds layer by layer by analyzing the correlation statistics of the last layer and clustering them into groups of units with high correlation [63]. The resulting clusters will form the units of the next layer and are connected to the units in the previous layer. This model is based on the assumption that each unit from the earlier layer will correspond to some region of the input and that these units will be grouped into filter banks. This process will result in the early layers which are closer to the input, to build up many clusters which are concentrated in a single region which can then be covered by a layer of 1×1 convolutions in the following layer. To avoid the overhead of expensive computations resulting from merging of the output of the pooling layer with that of the convolutional layer, a dimension reduction is applied to preserve the sparse representations in the network [63].

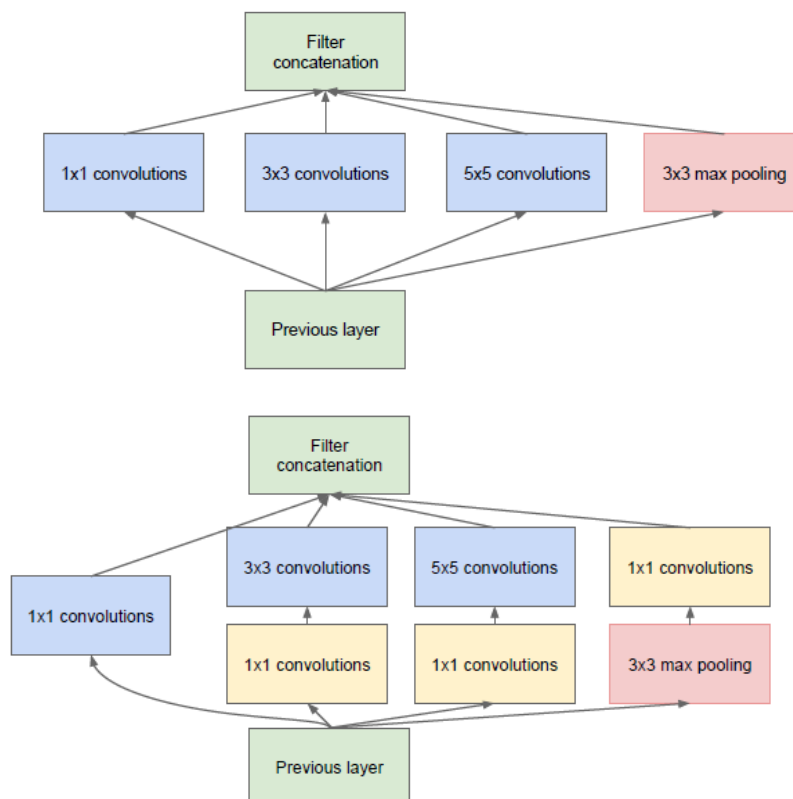


Figure 2.22 Inception Module Architectures [63]

2.8.6 (Google-Net) Going Deeper with Convolutions

Google-NET was able to outperform previous designs and win the ImageNet contest in 2014 ILSVCR14. The main advantage of this network architecture is the improvement in utilization of the computing resources inside the network. The implementation relied on designs which allowed increasing depth and width of the network while keeping the computational budget constant. The architecture decisions were based on the Hebbian principle and the use of multi-scale processing to optimize quality [63].

Google-NET has demonstrated that using dense building blocks for approximating the expected optimal sparse structure is a successful technique for improving the performance of Neural Networks. The experimental results of this network have shown that moving to sparser architectures is feasible and achieves comparable performance when compared to more expensive networks of similar depth and width [63].

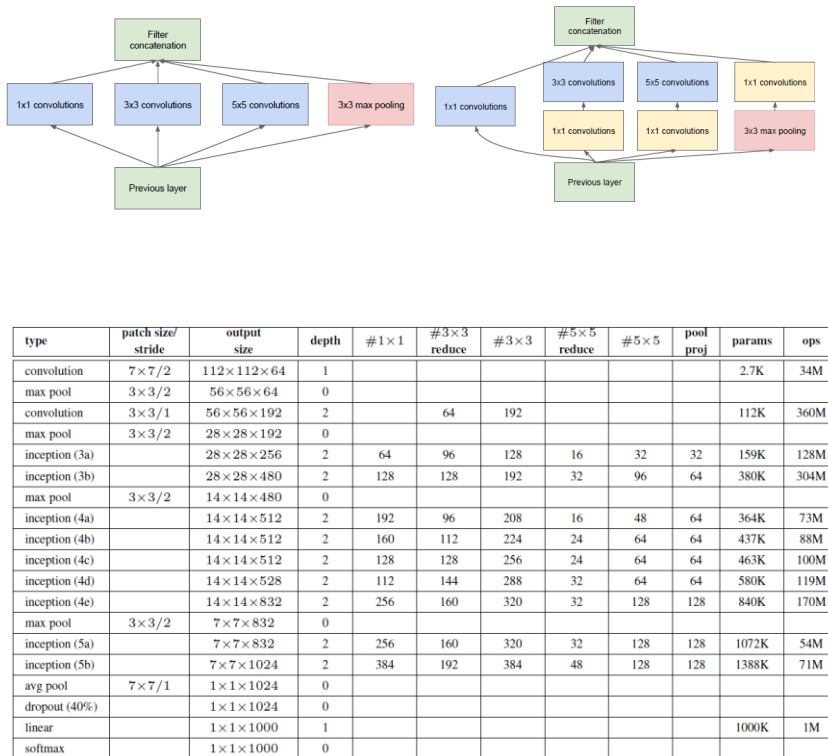


Figure 2.23 Google-NET Convolutional Neural Network Architecture [63]

2.8.7 Deep Residual Learning Framework

Challenge in Training Deeper Networks

State-of-the-art CNNs have shown that better classification performance can be achieved by architectures with substantially increased depth [64]. But at the same time, with depth being a significant factor, creating more deeper networks is not as simple as stacking more network layers into the network architecture [58].

One of the major problems that arises with training deeper networks is referred to as the Degradation problem [58]. Degradation occurs with the increase in network depth where the training accuracy gets saturated and then at a certain point it starts to rapidly degrade. Experiments have shown that the degradation is not caused by overfitting but rather due to the increase of network layers. The example below illustrates the training of the CIFAR-10 dataset where increase in number of layers has resulted in higher training and test error [58].

Residual learning was introduced to overcome degradation in learning performance with deep networks. If we assume that a series of stacked non-linear layers in a CNN can asymptotically approximate complex non-linear functions and that these layers can be represented by a mapping $H(x)$, where x is the input to the first layer, then we can equivalently assume that these stacked layers can also asymptotically approximate the residual function $H(x) - x$, given that both input and out have the same dimension.

The idea of residual learning is that instead of expecting the deep layers to approximate $H(x)$ we let them approximate a residual function $f(x) = H(x) - x$ so that the original function becomes $f(x) + x$. If the added layers can be constructed as an identity mapping then the training error of the deep network should not exceed the shallow model of the same network. Residual learning has proved that it is easier to optimize the residual mapping than to optimize the original mapping. In the case that the identity mapping was optimal then it would be easier to drive the residual to zero than to approximate the identity mapping by a stack of non-linear layers [58].

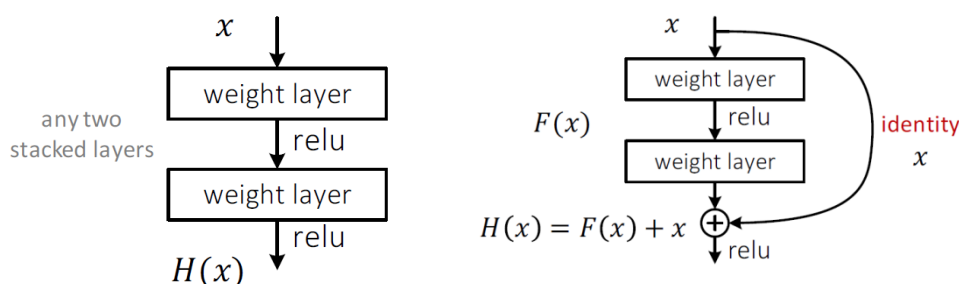


Figure 2.24 Deep Residual Learning Framework [58]

2.8.8 (ResNet) Deep Residual Learning Networks for Image Recognition

ResNet is a CNN architecture built on the residual learning framework [58]. The network was able to outperform previous designs and win the ImageNet contest in 2015 ILSVCR15. The architecture of this network provides the capability to train networks which are relatively deeper than previous network designs. The idea is based on reformulating the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. The experiments performed using this network demonstrated that Residual networks are easier to optimize and can achieve more accuracy when the depth of the network is increased [58].

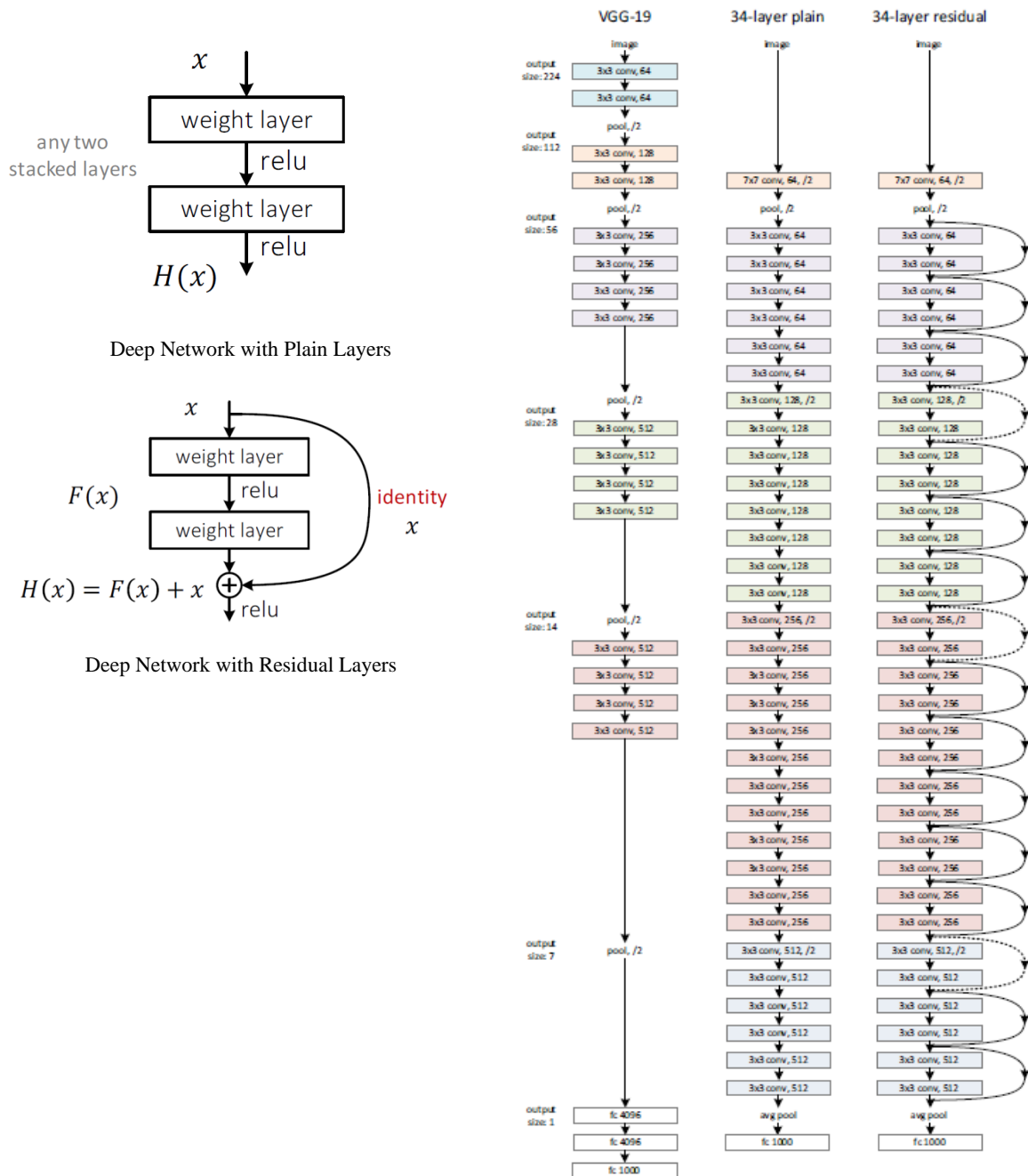


Figure 2.25 Comparison between Plain and Residual Convolutional Neural Network Architectures [58].

2.8.9 (Inception-ResNet) Impact of Residual Connections on Learning

The idea behind the architecture of the Inception-ResNet [57] Convolutional Neural Network was to build a very deep network by combining the successful models used for Inception architectures together with the learning technique of Residual connections. The motivation is that Inception architectures have been shown to achieve very good performance at relatively low computational cost and at the same time the use of residual connections have produced the best performance results in 2015 ImageNet challenge ILSVCR15 [57].

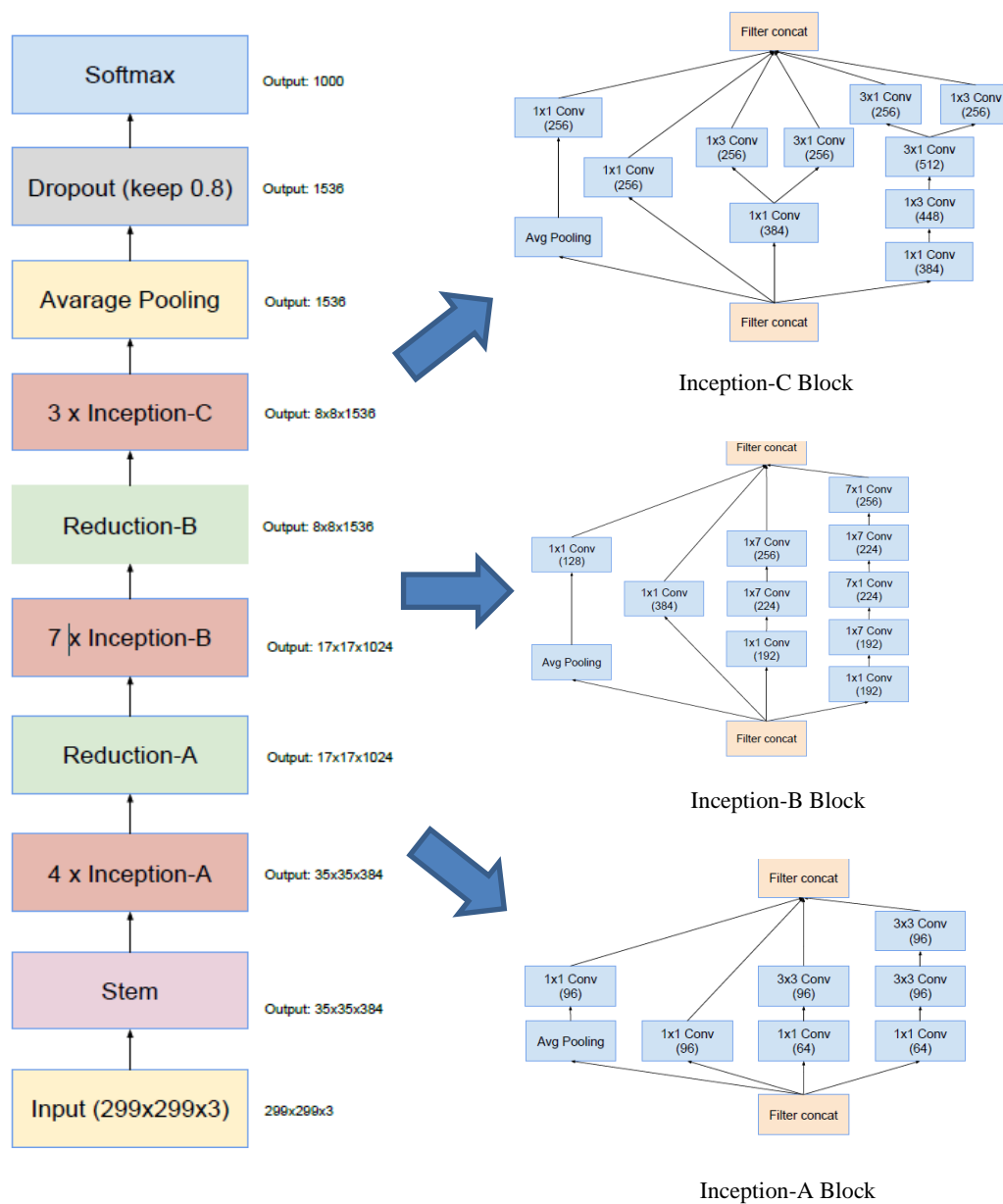
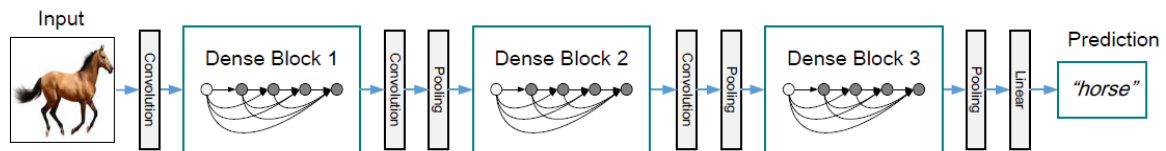


Figure 2.26 Inception-ResNet Convolutional Neural Network Architecture [57]

2.8.10 (DenseNet) Densely Connected Convolutional Networks

Dense Convolutional Networks attempt to ensure maximum information flow between layers in a deep network by connecting all layers, with matching feature-map sizes, directly with each other [41]. Recent studies have shown that convolutional networks can be substantially deeper, more accurate, and efficient to train if they contain shorter connections between layers close to the input and those close to the output. DenseNets build on this observation by connecting each layer to every other layer in a feed-forward fashion. Traditional convolutional networks with L layers have L connections, one between each layer and its subsequent layer. On the other hand, a DenseNet only has $L(L+1)/2$ direct connections [41]. For each layer, the feature-maps of all preceding layers are used as inputs and its own feature-maps are used as inputs into all subsequent layers. DenseNets have several advantages as they overcome the vanishing-gradient problem, strengthen feature propagation and substantially reduce the number of parameters. DenseNets obtained significant improvements over the state-of-the-art CNNs on four benchmark datasets used in object recognition (CIFAR-10, CIFAR-100, SVHN, and ImageNet) [41].



Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112×112	7×7 conv, stride 2			
Pooling	56×56	3×3 max pool, stride 2			
Dense Block (1)	56×56	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	56×56	1×1 conv			
	28×28	2×2 average pool, stride 2			
Dense Block (2)	28×28	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	28×28	1×1 conv			
	14×14	2×2 average pool, stride 2			
Dense Block (3)	14×14	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$
Transition Layer (3)	14×14	1×1 conv			
	7×7	2×2 average pool, stride 2			
Dense Block (4)	7×7	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Classification Layer	1×1	7×7 global average pool			
		1000D fully-connected, softmax			

Figure 2.27 DenseNet Convolutional Neural Network Architecture [41]

2.8.11 (NasNet) Learning Transferable Architectures for Image Recognition

The framework of learning transferable architectures is based on searching for an architectural building block on a small dataset and then transferring the block to a larger dataset [40]. The NasNet experiments search for the best convolutional layer or “cell” on a proxy dataset, such as the CIFAR-10 dataset, and then apply this cell to the ImageNet dataset by stacking together more copies of this cell, each with their own parameters to design the NasNet convolutional architecture. Searching for the best cell structure is much faster than searching for an entire network architecture and the cell itself is more likely to generalize to other problems. For experiments on ImageNet, a NASNet constructed from the best cell achieves accuracy of 82.7% top-1 and 96.2% top-5. The NasNet model is 1.2% better in top-1 accuracy than the best human-invented architectures while having 28% less FLOPS in computational demand from the previous state-of-the-art model. [40]

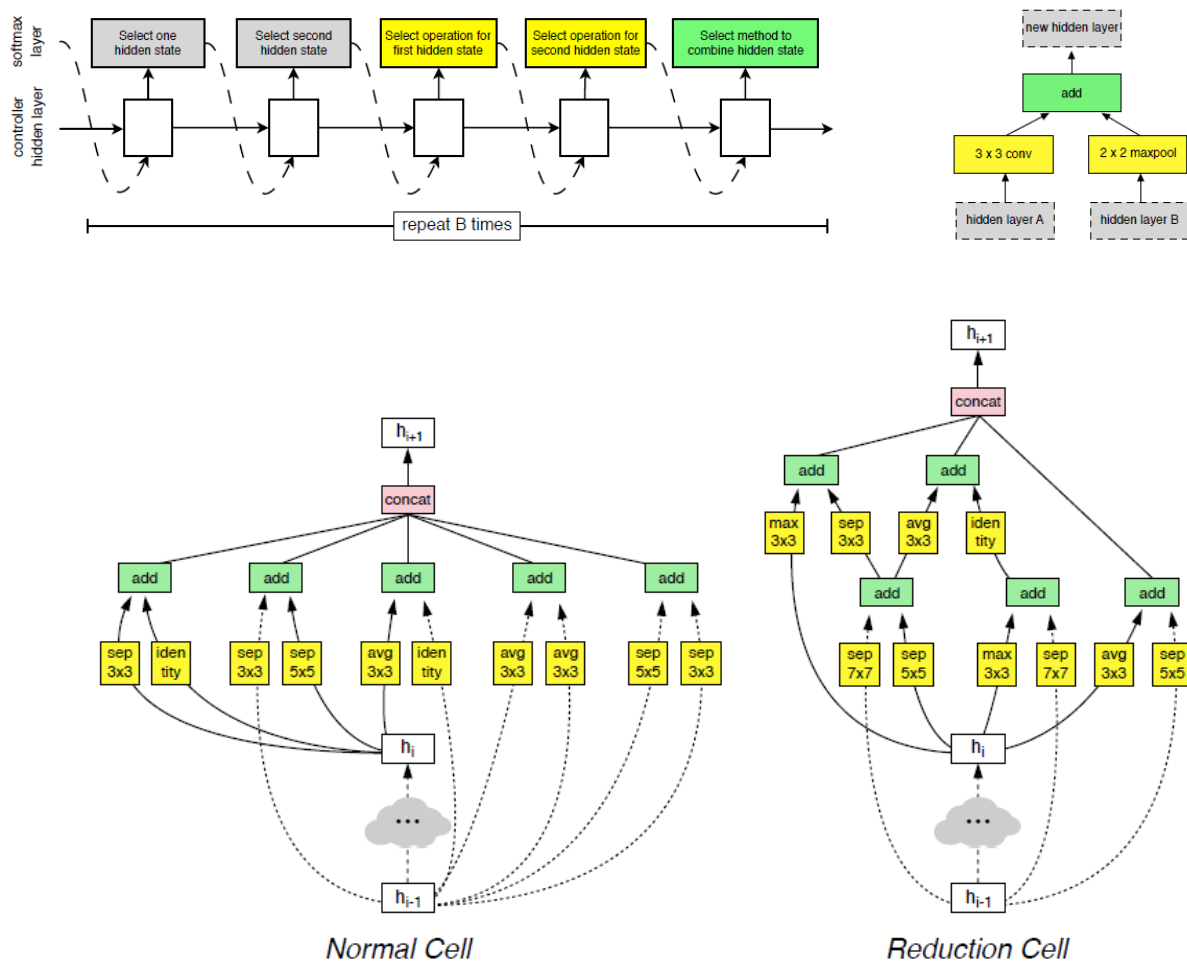


Figure 2.28 NasNet Convolutional Neural Network Architecture [40]

2.8.12 (MobileNet) Efficient CNNs for Mobile Vision Applications

MobileNet represents a class of efficient models designed for mobile and embedded vision applications. MobileNets are based on a streamlined architecture that uses depthwise separable convolutions to build light weight deep neural networks [39], [43]. The MobileNet architecture is based on depthwise separable convolutions which is a form of factorized convolutions which factorize a standard convolution into a depthwise convolution and a 1×1 convolution called a pointwise convolution. The depthwise convolution applies a single filter to each input channel while the pointwise convolution then applies a 1×1 convolution to combine the outputs of the depthwise convolution. This factorization has the effect of drastically reducing computation and model size [43]. Experiments have demonstrated the effectiveness of MobileNets across a wide range of applications and use cases including object detection, finegrain classification, face attributes and large scale geo-localization [43].

MobileNet V2 improve on the original design by introducing a novel layer called the inverted residual with linear bottleneck [39]. This module takes as an input a low-dimensional compressed representation which is first expanded to high dimension and filtered with a lightweight depthwise convolution and then features are subsequently projected back to a low-dimensional representation with a linear convolution. Experiments demonstrated that MobileNet V2 improves the state of the art performance of mobile models on multiple tasks and benchmarks as well as across a spectrum of different model sizes [39].

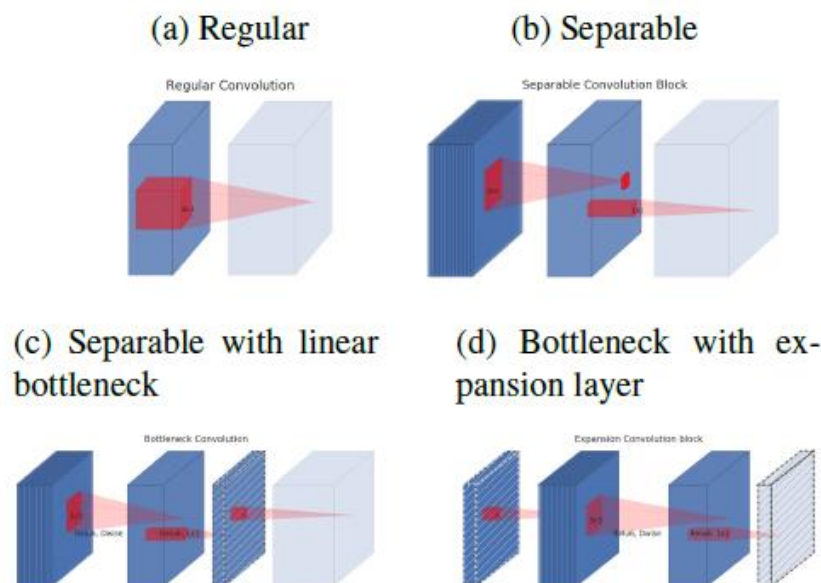


Figure 2.29 Architecture of Separable Convolution Blocks in MobileNet Convolutional Neural Network Architecture [39].

2.9 ROC Analysis for Evaluation of Classification Performance

The receiver operating characteristics (ROC) curves [68] will be used for evaluation of the classification performance of our proposed Convolutional Neural Network models as compared to the existing benchmarks for state-of-the-art classification methods. A confusion matrix is constructed by analysis of the four common possible outcomes which are defined for classification evaluation as shown in the figure.

		True class			
		p	n		
Hypothesized class	Y	True Positives	False Positives	fp rate = $\frac{FP}{N}$	tp rate = $\frac{TP}{P}$
	N	False Negatives	True Negatives	precision = $\frac{TP}{TP+FP}$	recall = $\frac{TP}{P}$
Column totals:		P	N	accuracy = $\frac{TP+TN}{P+N}$	
				F-measure = $\frac{2}{1/\text{precision}+1/\text{recall}}$	

Figure 2.30 Confusion Matrix Performance Metrics [66]

ROC Curve

The ROC curve is a 2-dimensional graph where the true positive rate (TP) is plotted on the Y axis and false positive rate (FP) is plotted on the X axis. The ROC describes relative tradeoffs between benefits (true positives) and costs (false positives). The lower left point (0, 0) represents a classifier which commits no false positive errors but also gains no true positives. The opposite of unconditionally issuing positive classifications is represented by the upper right point (1, 1). The point (0, 1) represents perfect classification. A point in ROC space is better than another if it is to the northwest, which means the true positive rate is higher or the false positive rate is lower or both [68].

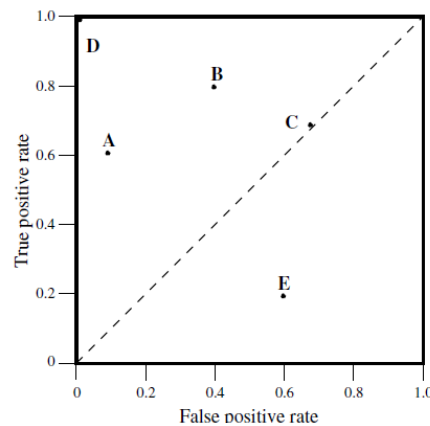


Figure 2.31 ROC Curve [68]

Applying a Threshold for ROC Curve

When using a layer such as Softmax for the Classification decision in a Convolutional Neural Network, the decision is a set of probabilities describing our confidence in each of the decisions. In this case a threshold can be used to produce a discrete classifier where each threshold value produces a different point in ROC graph. The figure shows an example of ROC curve where the instances have been sorted by their scores, and each point is labeled by the score threshold that produces it [68].

Avoiding Performance Evaluation Skews caused by Gene Expression Class Distribution

ROC curves have an advantage of insensitivity in changes to class distribution. If the proportion of positive to negative instances changes in a test set, the ROC curves will not change. The class distribution is the relationship of the positive column on the left to the negative column on the right. Any performance metric that uses values from both columns will be inherently sensitive to class skews. For example, Precision-Recall curves are sensitive to changes in class distribution as compared to ROC curves as demonstrated in the figure below [68].

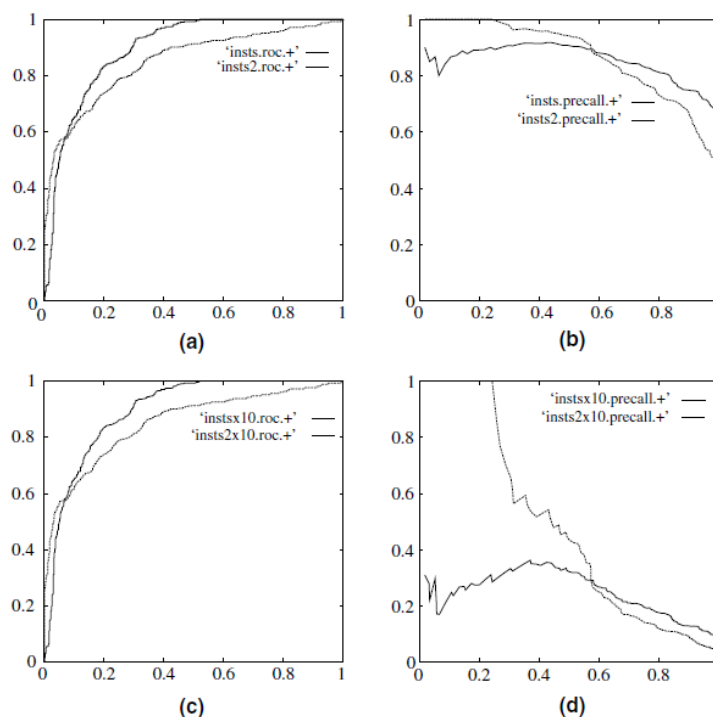


Figure 2.32 Comparison between ROC and Precision-Recall curves under skews in Class data Distribution [68]

- (a) ROC curves 1:1;
- (b) Precision-recall curves 1:1
- (c) ROC curves 1:10
- (d) Precision-recall curves 1:10

CHAPTER 3

3. RESEARCH METHODOLOGY

3.1 Problem Definition

The World Health Organization reports that cancer is a leading cause of death worldwide accounting for an estimated 9.6 million deaths in 2018 [1]. Despite this dramatic impact, between 30-50% of cancer death cases can be prevented through early detection and treatment [3]. Advancements in cancer classification and prediction play an important role in saving the lives of cancer patients, since a major challenge in cancer treatment is that patients are diagnosed at very late stages where appropriate interventions become less effective and full curative treatment is no longer achievable [4].

Machine learning for medical diagnosis using genomics is very difficult given the high dimensionality of the data and lack of sufficient patient samples for training [1], [4]. Technological advances in structural genomics have allowed studying the full set of DNAs in the human genome [25]. Next generation sequencing (NGS) methods such as whole-genome DNA sequencing and Total RNA sequencing are considered revolutionary technologies for studying genetic changes in Cancer [22], [27]. These technologies provide great potential for cancer classification and better understanding of tumor progression given their ability to sequence thousands of genes at one time and detect multiple types of genomic alterations [20], [21], [25]. They provide capabilities for comparing the sequence of DNA and RNA in cancer cells with that in normal cells to identify genetic changes that may be driving the growth of a tumor [26]. Gene expression analysis using total RNA sequencing provides a snapshot of the whole transcriptome rather than a predetermined subset of genes and can detect both coding plus multiple forms of noncoding RNA [22]. These methods have eliminated many limitations involved in microarray based experiments that were previously used for measuring gene expressions [22], [25], [27].

Cancer classification using gene expressions produced from Total RNA sequencing is extremely challenging given the complexity and massive amount of genetic data that is produced [20], [21], [25], [26], [38]. The magnitude of variants obtained from RNA-Sequencing is exponential which makes it difficult for traditional machine learning approaches to evaluate genetic variants for disease prediction [4], [22], [23]. Gene expression data is

characterized by being very high in dimensionality in terms of having a very large number of features representing the genes, and a very small number of training data representing the patient samples [9], [22], [33]. Complexity is also due to the fact that only a small subset of genes might be influencing the cancer tumor being diagnosed [4], [29].

Current cancer classification methods avoid processing the full set of genes to overcome these complexities and are mainly based on performing a process of gene feature selection as a prerequisite to the classifier learning process [28], [29], [30], [31]. Gene feature selection will allow the learning process to proceed, but the resulting classifier will not have the opportunity to learn the molecular signatures of genes which have been excluded and their influence on the underlying cancer tumor [34], [35]. Current classification methods which are based on gene feature selection are not optimal for early cancer diagnosis. This is because these methods will fall short in taking the full advantage of DNA and RNA sequencing technologies to discover the correlated patterns between genes across the full set of DNAs in the human genome and to detect multiple types of genetic alterations that may be driving the growth of a tumor across the whole transcriptome rather than a predetermined subset of genes [5], [6]. Another limitation of current methods is that they typically rely on gene expressions collected mainly from a single cancer tissue type based on the same anatomical site of origin. This approach does not utilize the full potential of the recent emerging whole-genome sequencing technologies and data produced by large-scale genomic projects which are producing detailed molecular characterizations of thousands of tumors using genome-wide platforms [38]. Recent studies which have performed an integrated multiplatform analysis across multiple cancer types have revealed molecular classification within and across tissues of origin [5], [7]. The results of these studies have recommended that the traditional approach of anatomic cancer classification should be supplemented by classification based on molecular alterations shared by tumors across different tissue types [5].

Deep Machine Learning continues to be an active research area [59] and therefore provides great potential for early disease detection and diagnosis. Among the great challenges in using deep learning for disease classification is the absence of a systematic approach to discover optimal model architectures. Deep learning is dependent on manually designing and configuring deep network architectures, where the optimal design configuration is achieved by training and experimentation on huge benchmark datasets [39], [41], [46], [57], [58], [42], [43]. Another challenge in using deep machine learning for disease diagnosis, is that deep networks are conceived as “black boxes” without much interpretation on how these complex models make their decisions [53]. Existing visualization techniques for deep networks used for

computer vision tasks [52], [53], [65] can be interpreted by non-experts when studied in conjunction with image or video datasets because they are visually comprehensible. These methods are not directly applicable to genomic datasets such as gene expressions, since they cannot be visually rendered in a human-friendly form that allows easy interpretations.

3.2 Research Objectives

To address the above problems, this has motivated our research for early cancer diagnosis by targeting the following research objectives:

Objective 1:

Leveraging the latest Deep Learning methods to design a comprehensive *Multi-Tissue cancer classifier* based on molecular signatures of whole-transcriptome wide gene expressions, that are collected from human samples representing multiple cancer tissue types and covering multiple organ sites.

Research Questions: Will the performance of disease prediction improve by learning the molecular signatures of whole-transcriptome wide gene expressions? Does a cancer classifier have to be limited to learning the molecular signatures of tumors from a single tissue type? Is there any value to learn the molecular signatures of tumors across multiple tissues and organ sites?

Method: Developing cancer classifiers with the capabilities of detecting more complex types of genetic alterations driving cancer progression, by learning the genomic signatures of whole-transcriptome gene expressions shared across multiple cancer tissue types and measuring the improvement in comparison to traditional single tissue classification.

Objective 2:

Design a Deep Learning framework for early cancer diagnosis by combining the process of gene feature selection and classification into one end-to-end learning system.

Research Questions: Can deep learning be used to overcome the limitations of traditional machine learning methods in processing complex high dimensional genomic data? Can we design a cancer classifier using genes across the full set of DNAs in the human genome without performing a prerequisite process of gene feature selection?

Method: Eliminating the dependency on the prerequisite process of gene feature selection which is performed by current state-of-the-art cancer classification methods for discovering a predefined subset of informative genes to be used in the learning process.

Objective 3:

Design a new *Deep Neural Network* architecture which is specifically designed to address the complex nature of whole-transcriptome gene expressions. The new model architecture should have the capabilities of learning the sequence of DNA and RNA in cancer cells and identifying genetic changes that alter cell behavior and cause uncontrollable growth and malignancy. The new architecture should also have the capabilities of learning the genomic signatures across multiple tissue types without requiring the prerequisite of gene feature selection.

Research Questions: Can we improve the performance of current cancer classifiers for early disease prediction by taking better advantage of Next Generation Sequencing methods such as whole-genome DNA sequencing and Total RNA sequencing that can provide a snapshot of the whole transcriptome? Can the existing state-of-the-art deep learning models that have been designed specifically for computer vision tasks, also be successfully applied for cancer classification using genomic data? Can we improve the performance of current cancer classifiers by designing a deep learning model architecture specifically designed for the complex nature of genomic data and whole-transcriptome gene expressions across multiple tissue types?

Method: Developing cancer classifiers with the capabilities of taking full advantage of genome-wide Next Generation Sequencing technologies to discover the correlated patterns of genes across the full set of DNAs in the human genome and across multiple cancer tissue types. To our knowledge, this is the first effort to develop a Multi-Tissue cancer classifier based on a full set of whole-transcriptome wide gene expressions collected from tumors across different tissue types without requiring a prerequisite process of gene feature selection.

Objective 4:

Design a *Deep Transfer Learning* model that can effectively function as a *generic* Multi-Tissue cancer classifier by learning genomic signatures collected from multiple cancer tissue types and using *Transfer Learning* to build classifiers for tumor types that are lacking sufficient patient samples to be trained independently.

Research Questions: Do we need a huge amount of human patient samples to train deep learning models with genomic data? Can we benefit from deep learning model architectures to efficiently build and train cancer classifiers despite the lack of huge amounts of cancer patient samples?

Method: Eliminating the dependency on huge amounts of patient data and contributing to solving one of the biggest challenges in cancer classification which is lack of patient samples. Comparing the classification performance between applying transfer learning using the genomic signatures of a pre-trained model versus performing a full training procedure using the available patient samples.

Objective 5:

Design an end-to-end *Deep Reinforcement Learning* framework that would automatically learn the optimal Deep Neural Network architecture together with the associated optimal hyperparameters that would maximize the performance of our multi-tissue cancer classifier.

Research Questions: Can we avoid the process of manually designing and handcrafting a deep model architecture and avoid the process of manually performing hyperparameter optimization to improve the performance of our cancer classifier?

Method: Developing a comprehensive multi-tissue cancer classifier that would eliminate the manual process of handcrafting the network architecture and eliminate the manual process of hyperparameter optimization and fine-tuning on the target dataset.

Objective 6:

Design visualization procedures to provide more biological insight on how the proposed network model is learning genomic signatures of whole-transcriptome gene expressions and accurately performing classification across multiple cancer tumors. Design the capability to visualize gene localization maps highlighting the important regions in the gene expressions influencing the tumor class prediction. Design the capability to visualize the molecular clusters formed by intermediate gene expression feature maps learned by the network which helps in revealing the genomic relationships of gene expressions that are influential in the tumor progression.

Research Questions: If we manage to successfully use deep learning models to improve the performance of cancer classifiers for early cancer diagnosis, can we provide medical professionals with any form of biological interpretation on how these complex models are making their predictions?

Method: Contribute to providing medical professionals with more confidence in using deep learning for medical diagnosis by providing interpretation on how these complex deep learning models are making their predictions.

3.3 Approach

The following sections outline our approach for achieving our research objectives and answering the research questions. We describe the motivation in using deep learning to design a multi-tissue cancer classifier and overcome the complexity in feature extraction in comparison to traditional machine learning methods. We outline our approach for using transfer learning to solve one of the biggest challenges in cancer classification which is lack of patient samples. We describe our approach to discover and learn the optimal deep network architecture that would maximize the performance of our classifier by designing an end-to-end Deep Reinforcement Learning framework. Finally, we introduce our approach using visualizations to provide more biological insight on how our deep learning framework is performing multi-tissue cancer classification. The detailed methods are presented in chapter 4.

3.3.1 Using Deep Learning to Design a Multi-Tissue Cancer Classifier

Deep Machine learning and Computational Intelligence are concerned with designing intelligent systems that can independently learn from data and make complex predictions and decisions in dynamically changing real world environments. Deep learning has had a major impact in many research and business applications such as Autonomous Self-driving Cars, Computer Vision, Medical Diagnosis, Biometric Identification, eCommerce, Banking and Cybersecurity. It has become a key element in many military defense applications and government intelligence and law enforcement agencies [40], [59], [60].

Traditional machine learning techniques have shown limitations in processing high dimensional data [66]. Recent neuroscience findings have provided additional insight into the principles governing information representation in the brain. The discovery motivated the emergence of deep machine learning which has since revolutionized the capabilities of processing high dimensional data [59]. Deep learning methods rely on building complex multi-layer network architectures capable of processing huge amounts of high dimensional data with minimal data preprocessing requirements, [42], [46], [58], Deep learning leverages spatial relationships among data to reduce the number of dimensions to be learned which dramatically improves the learning process in comparison to traditional machine learning techniques [67].

3.3.2 Overcoming Complexity in Feature Extraction of Gene Expression Data

Traditional machine learning methods are dependent on a prerequisite which requires domain experts to handcraft the relevant set of features to be used in the learning algorithm [59]. The design of a classification system required careful engineering and continuous fine-tuning to

design a feature extractor which would be capable of capturing the characteristics of the data being analyzed and transform it into a suitable feature vector to be fed as an input to the learning algorithm [67]. The performance of the learning system in terms of prediction and classification depended heavily on the successful identification of these features [59]. Deep learning on the other hand, is commonly referred to as Representation Learning since it is based on using raw data as input with no feature extraction as a prerequisite. Deep learning relies on building architectures with multiple levels of representation by combining non-linear building blocks, each level in the architecture transforms the input data into a representation at a higher more abstract level which provides the capability of learning complex non-linear functions [59]. For the problem of early cancer diagnosis using whole-transcriptome gene expression data, deep learning would provide the capabilities of automatically learning the molecular patterns of expressed genes which are influencing the cancer tumor being diagnosed and using that to amplify the discrimination score for classification. The major advantage is that the genetic features of the cancer tumors will not require to be pre-identified by medical professionals, but rather they will be automatically discovered through the deep learning process.

3.3.3 Deep Learning Architecture for Multi-Tissue Cancer Classification

Current methods for cancer classification are based on gene feature selection as a prerequisite to the classifier learning process. Our approach using deep learning provides an alternative solution to feature engineering and eliminates the dependency on huge amounts of training data and the prerequisite gene feature selection. This is achieved by combining the process of gene feature selection and classification into one end-to-end learning system using the whole set of transcriptome wide gene expressions collected from tumors across different tissue types. We propose a new Convolutional Neural Network (CNN) architecture called “Gene eXpression Network” (GeneXNet) that combines multiple layers of non-linear building blocks which transform the gene expression data into a representation at a higher more abstract level. This allows the network to automatically learn the molecular patterns of expressed genes which are influencing the tumors and use that to amplify the discrimination score for classification. The advantage is that the classifier will not be limited to learning the molecular characterization of a single tissue type but will have the capability of detecting more complex types of genomic alterations by learning the genetic signatures collected from multiple tumors and across multiple cancer tissue types. Another major advantage of our approach is that it allows performing very efficient transfer learning by reusing the molecular signatures learned by the trained networks. The weights of the pretrained networks can be used as feature extractors to

build and finetune classifiers for other different types of cancer tumors which might be lacking sufficient patient samples to be trained independently. This helps in solving one of the biggest challenges in building discriminative classifiers based on gene expressions which are characterized by having a very large number of genes versus a very small number of patient samples [1], [22].

3.3.4 Transfer Learning using Genomic Signatures of Multiple Tumors to Overcome Lack of Patient Samples

Our approach for building a comprehensive multi-tissue cancer classifier is by designing a new Convolutional Neural Network (CNN) architecture with the capability of learning the genomic signatures of whole-transcriptome wide gene expressions shared across multiple cancer tumor types. By training the model with samples from multiple tissue types collected from multiple organ sites, the classifier is able to learn and extract complex patterns from the gene expression data that represent genomic and transcriptomic alterations such as mutations, rearrangements, deletions, amplifications and the addition or removal of chemical marks. This allows the classifier to more accurately classify cancer tumors which are resulting from DNA or RNA changes that alter cell behavior across multiple tissues and cause uncontrollable growth and malignancy.

A major advantage is that we are able to reuse the genomic signatures learned by the trained model to perform very efficient transfer learning to solve one of the biggest challenges in cancer classification which is lack of patient samples. We demonstrate how transfer learning can be used to build and finetune classifiers for other different types of cancer tumors not included in the underlying dataset, which might be lacking sufficient patient samples to be trained independently. By reusing the weights of the pretrained network model, we demonstrate how the same network or an extended version of it can be used for feature extraction on a different cancer tumor type. The intuition behind transfer learning comes from recent studies which have performed an integrated multiplatform analysis across multiple cancer types that have revealed similar molecular classification within and across tissues of origin [5], [7]. This means that the discriminative molecular features for one cancer classifier will most likely be relevant for other cancer types. Our pretrained model will have already learned the complex types of genetic alterations and genomic signatures collected from multiple cancer tissue types originating from different organs and can effectively function as a generic model for cancer classification.

3.3.5 Deep Reinforcement Learning Framework to Discover the Optimal Deep Network Architecture

Our objective is to design an end-to-end learning framework that would enable us to automatically learn the optimal deep network architecture together with the associated optimal hyperparameters that would maximize the multi-tissue classification performance on our cancer tumor dataset.

The development of deep neural network architectures to improve accuracy and performance continues to be an active research area [39], [41], [46], [57], [58], [42], [43]. The drawback in using similar design approaches for building a comprehensive multi-tissue cancer classifier is that they rely on manually designing and configuring the network architecture [86], where the optimal design configuration is achieved by experimentation on benchmark datasets such as ImageNet [48]. One of the great challenges in using deep networks is the absence of a systematic approach to search within the huge network architecture space which is exponential in size to discover the optimal architecture [87]. Since our objective is to build a comprehensive multi-tissue cancer classifier based on molecular signatures of whole-transcriptome gene expressions, we would like to design our end-to-end deep learning framework without manually configuring the optimal network architecture. We would like to eliminate the manual process of handcrafting the network architecture which typically depends on carefully engineering and fine-tuning the design to achieve optimal performance on the target dataset.

To solve this problem, we propose a different approach by designing an end-to-end Deep Reinforcement Learning (DRL) framework. The objective of the DRL framework is to discover and learn the optimal Deep Network architecture that would maximize the performance of our multi-tissue cancer classifier on any potential gene expression dataset. In our proposed DRL framework, we use a Recurrent Neural Network (RNN) to generate different network architectures and we train the RNN using Reinforcement Learning to find an optimal architecture that would maximize the expected classification performance on our underlying multi-tissue cancer dataset. Our methods are motivated from the work done in the areas of Robotics and Optimal Control of Autonomous Vehicles using Deep Reinforcement Learning [89], [90], [91], [92], [93] and also the work done in Adversarial Game Playing using Reinforcement Learning and Deep Neural Networks [100], [101]. We build on the Policy Gradient optimization methods which use Reinforcement Learning and Trajectory optimization to learn complex nonlinear policies used in controlling high dimensional robotics systems using deep neural networks [94], [95], [96]. We also build on the gradient based

optimization methods using Recurrent Neural Networks which have been used for image classification [40], [81].

3.3.6 Visualizing Genomic Relationships of Gene Expressions Across Multiple Tumors

One of the challenges in using deep learning for disease diagnosis, is that deep networks are conceived as “black boxes” without much interpretation on how these complex models make their decisions [53]. Extensive work has been done to introduce novel visualization techniques for deep networks to help understand and interpret their record breaking performance in computer vision tasks [52], [53], [65]. The output from these techniques can be interpreted by non-experts when studied in conjunction with image or video datasets because they are visually comprehensible. Unfortunately, these methods are not directly applicable to genomic datasets such as gene expressions, since they cannot be visually rendered in a human-friendly form that allows easy interpretations. Our approach is to design a learning system architecture that can contribute in solving this problem by taking full potential of next generation sequencing technologies that produce datasets with detailed molecular characterizations of thousands of tumors using genome-wide platforms.

We introduce visualization procedures to provide more biological insight on how our model is performing cancer classification across multiple tumor types. We visualize gene localization maps highlighting the important regions in the gene expressions influencing the tumor class prediction. We also visualize the molecular clusters formed by intermediate gene expression feature maps learned by the network which helps in revealing the genomic relationships of gene expressions that are influential in the tumor progression.

CHAPTER 4

4. METHODS

The following sections present our detailed methods for achieving our research objectives. We describe our Deep Learning framework and formulate the details of our Gene eXpression Network architecture. We present the details of using transfer learning using genomic signatures across multiple cancer tumors. We formulate our network training and optimization using stochastic gradient descent and adaptive learning optimization. We describe and formulate the details of our end-to-end Deep Reinforcement Learning framework to discover and learn the optimal deep network architecture that maximizes the performance of our cancer classifier. Finally, we describe the details of our visualization procedures to provide more biological insight on how our framework is performing multi-tissue cancer classification.

4.1 Deep Learning System Architecture

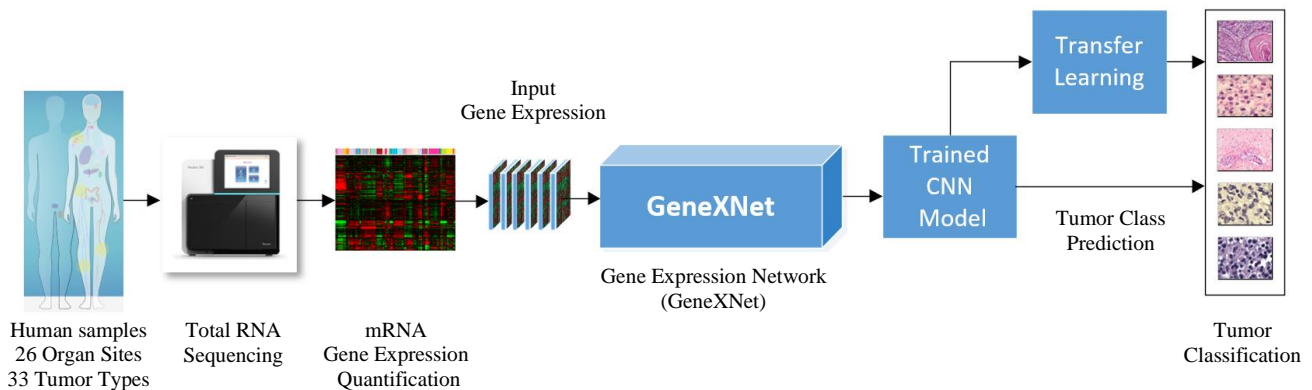


Figure 4.1 Deep Learning System Architecture

A schematic diagram of our end-to-end deep learning system architecture is shown in Figure 4.1. The first section represents the data collection and preparation process. It depends on collecting human samples representing multiple types of cancer tumors collected from multiple tissues spanning different organs across the body. The next step performs the gene expression quantification using a Next Generation Sequencing procedure. Total RNA sequencing is performed for measuring gene expression quantification across the whole-transcriptome and extracting both coding mRNA and noncoding miRNA. The gene expression data is normalized and then converted into a representation which makes it suitable for feeding it as input data to our deep learning model. Details about the cancer tumors used in our experiments is explained in the datasets section of the experiments chapter.

The second section of our learning system represents building and training a deep Convolutional Neural Network (CNN) to automatically learn the molecular signatures of the full set of whole-transcriptome gene expression data and produce a trained model which can be used for classification of cancer tumors. Our model, which we refer to as “Gene eXpression Network” (GeneXNet), relies on building an architecture with multiple layers of non-linear functions which transform the gene expression data into feature maps to increase the level of accuracy and invariance of the selected gene features [67]. As the model increases in depth, it becomes capable of representing complex genetic alterations shared by tumors across different tissue types, which are very sensitive to the slightest details in the input samples. The genetic signatures learned by the feature maps in the deep layers, eliminate the need for the traditional prerequisite process of gene feature selection. This is because the feature maps are insensitive to any insignificant genes or irrelevant variations in the gene expression data [59], [67].

We train the model using supervised learning by feeding the collected human samples as input and producing an output probability score for each labelled category of cancer tumors. We define a cross-entropy loss function suitable for gene expression data that measures the error between the network input and the desired output, then we use stochastic gradient descent optimization and backpropagation [62] to adjust the network weights and reduce the classification error to the optimal levels. Full training details are explained in the experiments.

4.2 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) have contributed to many record breaking achievements especially in the areas of computer vision and image recognition [39], [40], [41], [42], [46], [60]. The development of new CNN architectures to improve accuracy and performance continues to be an active research area such as AlexNet [66], VGGNet [64], GoogLeNet [63], InceptionNet [57], ResNet [46], [58], DenseNet [41], MobileNet [39], [43], SENet [42] and NasNet [40]. CNNs are made of multiple layers where each layer is arranged in the form of a 3D volume of neurons that has a specific width, height and depth. Each layer transforms the input volume to an output volume using a non-linear transformation function. CNNs differ in that the neurons in a particular layer will only be connected to a small region in the previous layer instead of the traditional fully connected networks [59].

The motivation in using CNNs for classification of cancer tumors using gene expressions is that the convolution operation is very suitable for the high dimensional and sparse nature of the data. Since the input data has a very high dimensionality, it is not practical to use traditional kernel learning methods and fully connected networks since the resulting models will have a huge number of parameters to be learned which makes the learning process infeasible [59].

4.3 Gene Expression Data Representation for CNNs

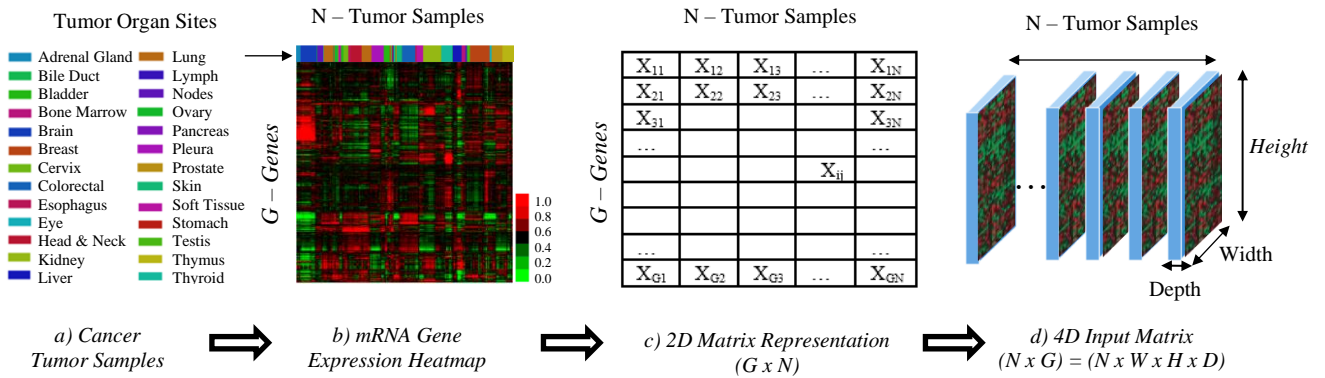


Figure 4.2 Gene Expression Data Representation for CNNs

In order to train our Convolutional Neural Network model using the cancer tumor samples, we first have to represent the gene expression data in a format suitable for the network input. Figure 4.2 shows an outline for the gene expression data representation we implement for one of the datasets used in our experiments for patient samples representing 26 different cancer organ sites [38]. Full details about the cancer tumors used in our experiments is explained in the experiments chapter. Figure 4.2(b) represents a clustering heatmap of the mRNA gene expressions. The column clusters represent the cancer tumor types grouped by organ site, where each column represents a patient sample, each row represents a single gene and the cell color legends reflect the mRNA expression level of genes. If we have a total of N cancer tumor samples, each sample will have a total of G features representing the full set of genes produced by the whole-transcriptome sequencing procedure. We then represent the gene expression data in an equivalent 2D matrix of real numbers with dimensions (G, N) as in Figure 4.2(c). The matrix stores real values of the normalized gene expressions such that the value in cell X_{ij} represents the expression level measured for gene (i) in the patient sample (j). Each tumor sample can be represented by a $(G, 1)$ dimensional vector of gene expressions which we convert into the equivalent 3D volume with dimensions (Width, Height, Depth) to make it suitable as an input vector to our CNN model. The volume dimensions can be reshaped with any arbitrary length which matches the correct number of total features. The depth dimension is taken from the CNN terminology used in image classification where the depth is usually set to 3 representing the number of RGB color channels. For images, values represent the pixel intensity, while for our cancer tumor dataset the values represent gene expression quantification. The training data for all the N samples can then be represented by the 4D input matrix with dimensions (No. of Samples, Width, Height, Depth) as shown in Figure 4.2(d).

4.4 Learning Genomic Signatures using Convolutions

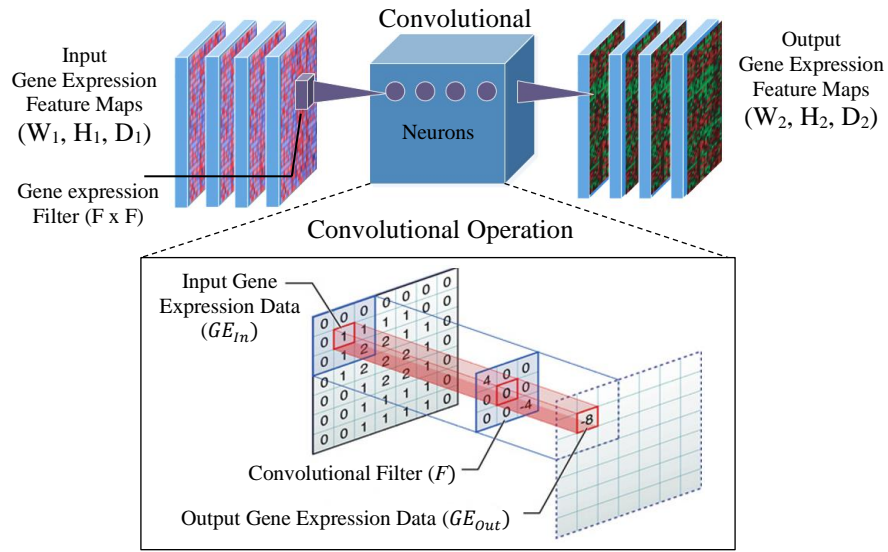


Figure 4.3 Convolutional Layer for Gene Expression Data

The intuition in using the convolution operation for learning genomic signatures of cancer tumors is to leverage several architectural characteristics which distinguish Convolutional Neural Networks from traditional machine learning methods. These characteristics include: Sparse Connectivity, Parameter Sharing, Pooling and Equivariant representations [59], [67]. Since the gene expression data is very high in dimensionality, it is not practical to use the traditional fully connected neural networks since the resulting network will have a huge number of parameters to be learned which makes the learning process infeasible. To overcome this problem, we make use of convolutional layers which implement small convolutional filters to represent the weight parameters of the model. Figure 4.3 shows a schematic diagram for the convolutional layer we implement for gene expression data. The objective of the convolution layers is to learn filters that will be activated when matched with specific patterns or features in the gene expressions. The convolution layers perform a convolution operation which is a dot product between a sliding filter and the input across the full depth of the input gene expression volume to produce an activation map. We implement the convolution operation as in [75] by defining the convolution for each 2D layer of the gene expression volume as:

$$GE_{out}(i, j) = (F * GE_{in})(i, j) = \sum_m \sum_n GE_{in}(i - m, j - n) F(m, n) \quad (1)$$

where GE_{in} , GE_{out} represents the input and output gene expression feature maps and F is the sliding convolutional filter. The output volume of each layer is created by stacking the activation maps for all filters.

Each neuron in the convolutional layer is only connected to a local region in the input volume covering the full depth which is the receptive field of the neuron. The neurons still perform the standard operation of a neural network by calculating the dot product between the input and the weights then applying a non-linear function. The main difference is that the neuron is connected only with its receptive field in the input and at the same time shares the same weight parameters as other neurons in the same feature activation map.

Since the gene expression feature maps are represented by a 3D volume, we then define the convolution across the full depth of feature maps as:

$$GE_{out}(k, i, j) = \sum_{l, m, n} GE_{In}(l, i + m - 1, j + n - 1) F(k, l, m, n) \quad (2)$$

where GE_{In} , $GE_{out}(k, i, j)$ represents the input and output gene expression values for the feature map at depth k , row i and column j and $F(k, l, m, n)$ represents a 4D convolutional filter between the output feature map at depth k , and the input feature map at depth l with an offset of m rows and n columns.

In order to tackle the complexity and high dimensional nature of the gene expression data we also make use of *downsampling* as in [75] by defining a stride parameter S to skip over some positions of the gene expression feature maps to reduce the computational cost:

$$GE_{out}(k, i, j) = \sum_{l, m, n} [GE_{In}(l, (i - 1) * S + m, (j - 1) * S + n) F(k, l, m, n)] \quad (3)$$

Table 4.1 shows the formulas we use to calculate the dimensions of the output volume representing the gene expression feature maps after applying the convolution operation.

Table 4.1 Calculating Volume of Gene Expression Feature Maps after Convolution

Parameter	Description
W_1, H_1, D_1	Input volume width, height and depth
G	No. of genes = $(W_1 \times H_1 \times D_1)$
K	No. of Filters = No. of Hidden Neurons = No. of Activation Maps = D_2
F	Filter size = $(F \times F)$
S	Stride applied when moving the filter across the input volume
P	Zero padding applied to input volume
W_2	Output Volume Width = $[(W_1 - F + 2P)/S] + 1$
H_2	Output Volume Height = $(H_1 - F + 2P)/S + 1$
D_2	Output Volume Depth = No. of Filters = K

By implementing local receptive fields using small sized filters, the network can learn and extract small meaningful relationships between the molecular signatures of the genes which in turn can describe the characteristic influencing the cancer tumor. The sparse network connectivity, parameter sharing and convolutions using small kernels have helped in tackling the very high dimensional nature of the gene expression data since it dramatically reduced the number of parameters that the network needs to learn which means the learning process is much more efficient in terms of computation and storage requirements. This also helped in overcoming problems related to lack of sufficient cancer patient samples since the learning process became less prone to overfitting. Figure 4.4 illustrates the reduction in complexity by using sparsely connected networks for gene expression data in comparison to fully connected networks.

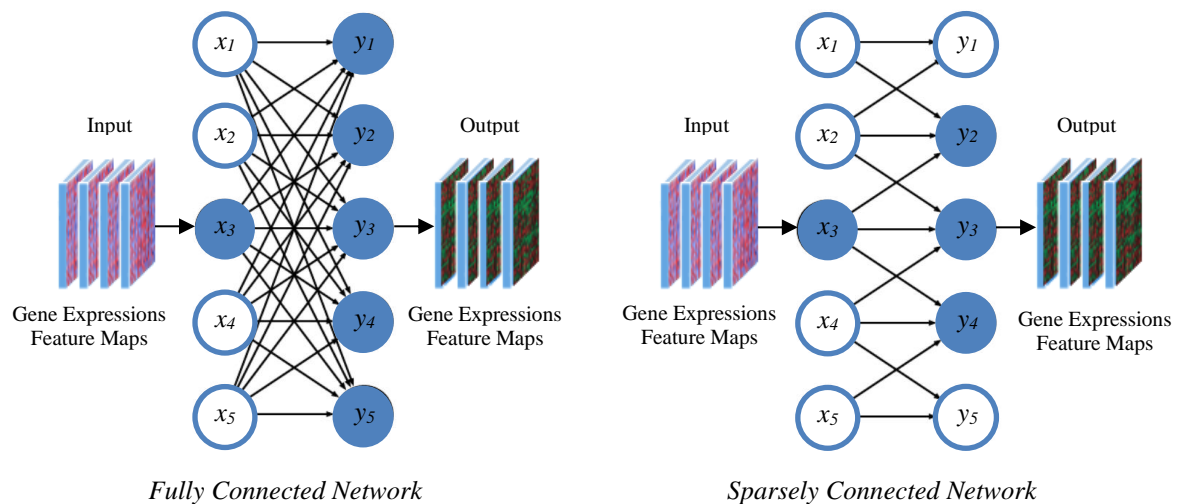


Figure 4.4 Sparsely Connected Networks for Learning Gene Expressions

We also make use of a *Pooling* layer after the convolution layer which provides a very important characteristic for learning the genomic signatures of cancer tumors by performing subsampling of the gene expression data. Our network design incorporates a max pooling function which is based on replacing the output of the feature maps at certain locations with a summary statistic of the nearby output values [75]. This allows our network model to generate gene expression feature maps which are invariant to local translations in the molecular signatures of the cancer tumor. This is a very important feature which enables building a classifier that can make predictions across multiple tumor types with the capability of learning the complex types of genomic signatures collected from multiple cancer tissue types originating from different organs. The intuition in extracting features which are invariant to local translations is adapted from using convolutional neural networks for image recognition.

4.5 Gene eXpression Network (GeneXNet) Architecture

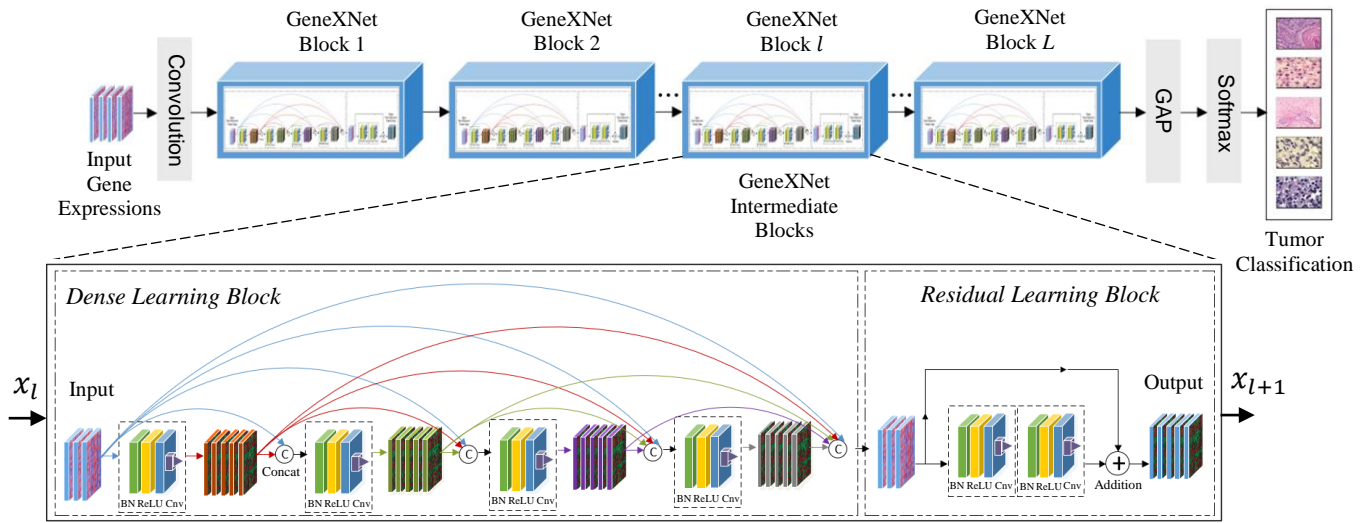


Figure 4.5 Gene eXpression Network (GeneXNet) Architecture

In this section we describe the detailed architecture of our proposed CNN model shown in Figure 4.5. Recent benchmark results obtained by deep CNNs for image recognition tasks have demonstrated that network depth is of great importance for feature extraction and have managed to achieve outstanding results by designing networks with deeper and more complex architectures [46], [58]. These models were able to exploit deep architectures because of the availability of large training datasets such as ImageNet which contains over 1 million training images [48]. Training deep models requires large amounts of training data to avoid common problems such as overfitting, vanishing gradients and degradation of accuracy [46], [58].

Applying the same deep CNN architectures for classification of gene expression data is not an evident task since it faces two conflicting problems. On one hand, we need to benefit from deep network architectures to efficiently extract the molecular signatures of the large number of genes so that our classifier can accurately generalize when presented with tumor data from multiple tissue types. But on the other hand, the lack of sufficient human training samples, which could be in the range of only a few hundred samples, implies great challenges for training deep networks and results in overfitting during training which implies using smaller more compact networks.

We attempted to build an end-to-end learning system for cancer classification without performing the prerequisite process of gene feature selection by using some of the available state-of-the-art CNN models which have been specifically designed for computer vision tasks. Our experimental results have shown that training these deep models suffered from severe

overfitting when presented with the underlying dataset that includes the full set of transcriptome gene expressions collected from tumors across different tissue types. The dataset did not have sufficient training samples to train these deep models and achieve the required accuracy. Many regularization methods have been proposed to overcome overfitting by adding constraints on the learning model or including additional terms in the error function which can potentially help to decrease overfitting and improve performance [75]. Examples of regularization methods include L1, L2 regularization, early stopping, noise injection, data augmentation, bagging and dropout [51], [61], [75]. Our experiments have shown that these regularization methods could slightly help in reducing overfitting but are not sufficient to build a general multi-tissue cancer classifier given the large number of features in the whole-transcriptome gene expressions and lack of sufficient cancer patient training samples.

To solve these conflicting problems, we propose a new CNN architecture which we refer to as Gene eXpression Network (GeneXNet) shown in Figure 4.5. Our network is designed to specifically target the complex nature of gene expression data and also addresses the lack of training samples by incorporating multiple layers of building blocks which we refer to as GeneXNet blocks shown in Figure 4.6. These blocks are motivated from both deep residual learning networks [46], [58] and also densely connected convolutional networks [41] and are formed by merging together two different types of learning sub-blocks.

4.6 GeneXNet Building Block Formulation

Our proposed Gene eXpression Network (GeneXNet) architecture combines multiple layers of non-linear building blocks which transform the gene expression data into a representation at a higher more abstract level allowing the network to automatically learn the molecular signatures influencing the cancer tumors. We refer to these blocks as GeneXNet blocks which are shown in Figure 4.6 and are formed by merging together two different types of learning sub-blocks.

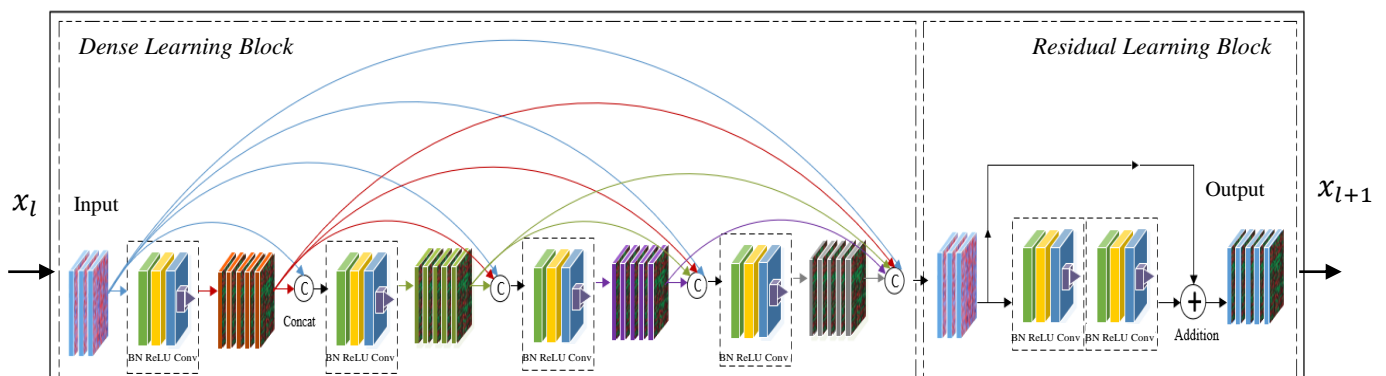


Figure 4.6 Gene eXpression Network (GeneXNet) Building Blocks

To formulate our building block, we define our network to have L layers of blocks where the non-linear transformation of Gene expressions can be denoted by G_l and can be defined as:

$$x_{l+1} = G_l(x_l, W_l) \quad (4)$$

$$W_l = \{w_{l,i} | 1 \leq i \leq K_l\} \quad (5)$$

where l is the index of the block, W_l represents the set of weights and biases of the l^{th} block, $w_{l,i}$ represents the weights of the i^{th} convolutional layer in the l^{th} block, K_l represents the number of convolution layers in the l^{th} block and x_l, x_{l+1} represent the input and output features of the l^{th} block. We apply “pre-activation” of weight layers as in [46] by defining the transformation at each layer as a sequence of multiple operations which are Batch Normalization (BN) [55], Rectified Linear Unit (ReLU) [49] and Convolution.

If gene expression data flows through the network using only the transformation in (4), that would be following the traditional approach for CNN layers. Deep residual learning provides a framework for more efficiently training deep networks by reformulating the layers as learning residual functions with reference to the layer inputs [58]. Empirical results have shown that residual learning helps to avoid degradation in performance accuracy as the depth of the network increases [58]. Residual networks have achieved excellent performance in many image recognition and object detection tasks where networks with over 150 layers have been trained on ImageNet [66] and managed to achieve substantial accuracy gains in comparison to normal networks which simply stack consecutive layers [46]. To make use of residual learning we reformulate our building block by implementing the non-linear transformation of gene expressions G_l as a residual function defined as:

$$x_{l+1} = f_l[G_l(x_l, W_l) + M(x_l)] \quad (6)$$

where G_l is a residual function for the l^{th} block, $M(x_l)$ is a mapping which bypasses the non-linear transformation and f_l represents a mapping function of the input and output features of the l^{th} block. The simplest form of residual learning can be realized by choosing f_l to be a Rectified Linear Unit (ReLU) [49] and also introducing identity skip connections which are equivalent to choosing $M(x_l)$ as an identity mapping so that $M(x_l) = x_l$. Another formulation can be realized by implementing both $M(x_l)$ and f_l as identity mappings. We apply the later formulation which has shown to improve accuracy by creating a more direct path for information propagation and allowing the signal to propagate more directly from one unit to

any other unit in the forward and backward passes [46]. The resulting non-linear transformation of gene expressions and the gradient of the loss function can then be expressed recursively as:

$$x_L = x_l + \sum_{i=l}^{L-1} G_i(x_i, W_i) \quad (7)$$

$$\frac{\partial \varepsilon}{\partial x_l} = \frac{\partial \varepsilon}{\partial x_L} \frac{\partial x_L}{\partial x_l} = \frac{\partial \varepsilon}{\partial x_L} \left[1 + \frac{\partial}{\partial x_l} \sum_{i=l}^{L-1} G_i(x_i, W_i) \right] \quad (8)$$

where x_L represents the output features of the network with L layers of blocks, ε is the loss function and $\partial \varepsilon / \partial x_l$ is the gradient obtained by applying the chain rule and backpropagation [46]. The residual function G_i is implemented as in (4) by applying two or more weight layers each using pre-activation and the sequence of multiple operations BN, ReLU and convolution.

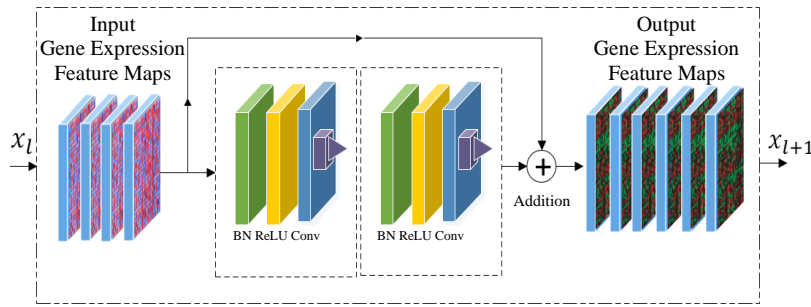


Figure 4.7 Residual Learning Block of Gene eXpression Network.

The resulting block is shown in Figure 4.7 which we refer to as the *Residual Learning block*. We also experiment with applying a bottleneck architecture [46], [58], by modifying the design of this block to have three layers instead of two in the form of (1x1), (3x3) and (1x1) convolutions. Since we are using the full set of whole-transcriptome genes, the role of the (1x1) convolution is to enhance computational efficiency by reducing the large dimensions of the intermediate feature maps before applying the convolution and then restore them back again.

Despite the strong advantages of residual learning networks in allowing the gradient to flow directly through the skip connections, there have been other proposed approaches to use stochastic depth to improve the training of deep residual networks by dropping layers randomly during training [47]. This has led to different intuitions that there might be a great amount of redundancy in deep residual networks and that not all the layers are required [41]. Densely connected convolutional networks (DenseNets) [41] exploit the potential of the network through feature reuse as an alternative to deep or wide architectures by connecting all layers

with matching feature-map sizes directly with each other. This design consideration is very important for our task, since one of the biggest challenges in our work is to build a multi-tissue cancer classifier that can benefit from deep network architectures to efficiently extract the molecular signatures of large number of genes, without facing severe overfitting or degradation in performance due to the lack of sufficient human training samples. This has inspired us to further reformulate the design of our GeneXNet building block and augment its learning capability by introducing additional dense layers that precede the residual learning layers. The dense layers follow a similar approach as in DenseNets [41]. The design of our dense layers is implemented by passing additional inputs into each layer from all preceding layers and passing the feature maps of each layer to all subsequent layers. Our aim from this design is to provide each layer with direct access to the gradients from the loss functions and the original input signal which can potentially improve flow of information throughout the network. Our additional dense layers are formulated as follows:

$$x_{l+1} = G_l(Concat[x_1, x_2, x_3, \dots, x_l]) \quad (9)$$

where x_{l+1} represents the output of the l^{th} block, $Concat[x_1, x_2, \dots, x_l]$ represents the concatenation of the gene expression feature maps resulting from all preceding layers and G_l represents the same transformation as in (4) which applies pre-activation of weights and the sequence of multiple operations BN, ReLU and convolution. The resulting block is shown in Figure 4.8 which we refer to as the *Dense Learning block*.

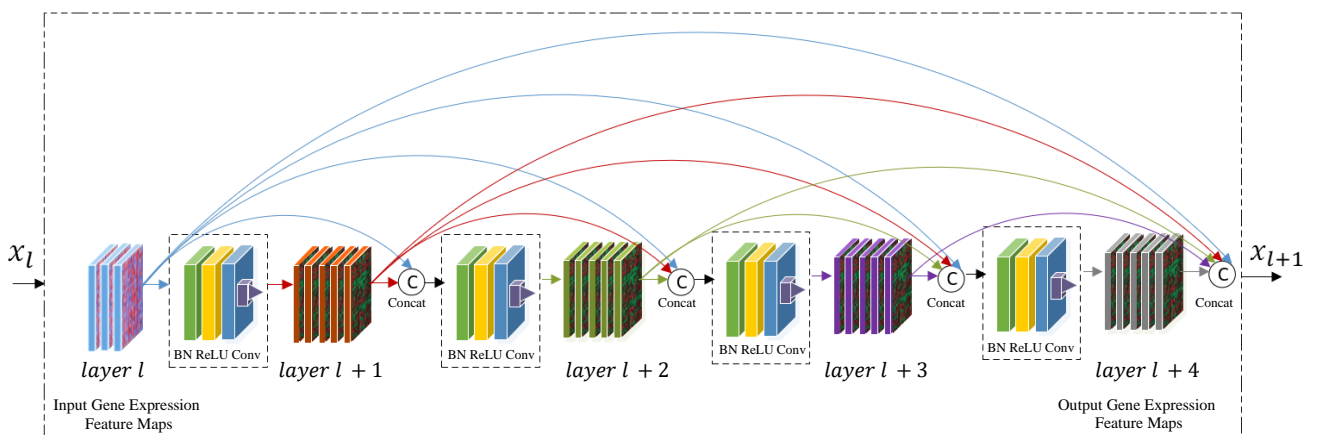


Figure 4.8 Dense Learning Block of Gene eXpression Network.

Our proposed GeneXNet block is finally formed by merging together these two sub-blocks as shown in Figure 4.6, which represents a combination of dense learning and residual learning layers. We define several parameters in order to control the variation of the network design and

size across different gene expression data sets. The parameter θ_k controls the number of filters used in the convolution layers. The two parameters θ_D and θ_R define the percentage of dense and residual sub-blocks in the network, where $0 \leq \theta_D \leq 1$ and $0 \leq \theta_R \leq 1$. For example, when both θ_D and θ_R are set to 1, then all the network blocks include both a dense and a residual sub-block. When θ_D is set to 1 and θ_R to 0.5, then all the network blocks include dense sub-blocks while only half of the blocks include residual sub-blocks.

The full Gene eXpression Network (GeneXNet) architecture is shown in Figure 4.5. It is implemented by feeding the gene expression input matrix to multiple layers of GeneXNet blocks each containing a combination of dense and residual learning layers as described above. The network ends with a global average pooling [54] after the last GeneXNet block and a fully connected softmax layer for classification. We experiment with different network sizes having two to four GeneXNet blocks and with different θ_k , θ_D , θ_R configurations. A detailed architecture is shown in table 2 implementing a network with four GeneXNet blocks, $\theta_k=32$ and both θ_D , θ_R set to 1.

Table 4.2 Gene eXpression Network detailed architecture.
(Implementing a network with 4 blocks, $\theta_k=32$, $\theta_D=1$, $\theta_R=1$)

GeneXNet Block (l)	Output Size	Dense Sub-block		Residual Sub-block	
		Layer operations $\theta_D = 1$	Filters $\theta_k = 32$	Layer operations $\theta_R = 1$	Filters $\theta_k = 32$
Input	(142,142,3)				
Pre-layers	(71,71,64)	<i>Conv(7x7, 64)</i>			
GeneXNet Block 1	(36,36,256)	$\left[\begin{array}{c} \text{Conv}(1 \times 1, 128) \\ \text{Conv}(3 \times 3, 32) \end{array} \right] * 6$	$4\theta_k$ θ_k	$\left[\begin{array}{c} \text{Conv}(1 \times 1, 64) \\ \text{Conv}(3 \times 3, 64) \\ \text{Conv}(1 \times 1, 256) \end{array} \right] * 2$	$2^l \theta_k$ $2^l \theta_k$ $2^{l+2} \theta_k$
GeneXNet Block 2	(18,18,512)	$\left[\begin{array}{c} \text{Conv}(1 \times 1, 128) \\ \text{Conv}(3 \times 3, 32) \end{array} \right] * 12$	$4\theta_k$ θ_k	$\left[\begin{array}{c} \text{Conv}(1 \times 1, 128) \\ \text{Conv}(3 \times 3, 128) \\ \text{Conv}(1 \times 1, 512) \end{array} \right] * 2$	$2^l \theta_k$ $2^l \theta_k$ $2^{l+2} \theta_k$
GeneXNet Block 3	(9,9,1024)	$\left[\begin{array}{c} \text{Conv}(1 \times 1, 128) \\ \text{Conv}(3 \times 3, 32) \end{array} \right] * 24$	$4\theta_k$ θ_k	$\left[\begin{array}{c} \text{Conv}(1 \times 1, 256) \\ \text{Conv}(3 \times 3, 256) \\ \text{Conv}(1 \times 1, 1024) \end{array} \right] * 2$	$2^l \theta_k$ $2^l \theta_k$ $2^{l+2} \theta_k$
GeneXNet Block 4	(5,5,2048)	$\left[\begin{array}{c} \text{Conv}(1 \times 1, 128) \\ \text{Conv}(3 \times 3, 32) \end{array} \right] * 16$	$4\theta_k$ θ_k	$\left[\begin{array}{c} \text{Conv}(1 \times 1, 512) \\ \text{Conv}(3 \times 3, 512) \\ \text{Conv}(1 \times 1, 2048) \end{array} \right] * 2$	$2^l \theta_k$ $2^l \theta_k$ $2^{l+2} \theta_k$
Classification	(1,1,2048)	<i>Global Average Pooling</i>			
	(C-Classes)	<i>Fully connected (C-Tumor Types) – Softmax</i>			

Our results have demonstrated that our proposed network which combines both dense and residual learning layers, has allowed training deeper network architectures with complex data such as gene expressions, despite the large number of genes. The dense layers allow the network to efficiently extract the genetic signatures from multiple tumors and across multiple cancer types. This is achieved by means of re-using the gene expression feature maps learned by different layers, which increases the variation of input signals fed to subsequent layers since it represents the collective knowledge of the network [41]. The residual layers with identity mappings contribute to providing a direct path for information propagation in the forward and backward passes [58] while the connectivity of the dense layers provide each layer with more direct access to the gradients from the loss function and the original input signal [41]. Our results have also shown that the combination of dense connections augmented with residual layers performs a regularizing effect which allows the network to achieve high accuracy in tumor classification while avoiding problems related to overfitting due to lack of human samples.

4.7 Transfer Learning using Genomic Signatures Across Multiple Cancer Tumor Types

Our approach for building a comprehensive multi-tissue cancer classifier is by designing the Gene eXpression Network (GeneXNet) with the capability of learning the genomic signatures of whole-transcriptome wide gene expressions shared across multiple cancer tumor types. By training the model with samples from multiple tissue types collected from multiple sites of origin, the classifier is able to learn and extract complex patterns from the gene expression data that represent genomic and transcriptomic alterations such as mutations, rearrangements, deletions, amplifications and the addition or removal of chemical marks. This allows the classifier to more accurately classify cancer tumors which are resulting from DNA or RNA changes that alter cell behavior across multiple tissues and cause uncontrollable growth and malignancy.

A major advantage is that we are able to reuse the genomic signatures learned by the trained model to perform very efficient transfer learning to solve one of the biggest challenges in cancer classification which is lack of patient samples. We demonstrate how transfer learning can be used to build and finetune classifiers for other different types of cancer tumors not included in the underlying dataset, which might be lacking sufficient patient samples to be trained independently. By reusing the weights of the pretrained GeneXNet model, we

demonstrate how the same network or an extended version of it can be used for feature extraction on a different cancer tumor type.

The intuition behind transfer learning comes from recent studies which have performed an integrated multiplatform analysis across multiple cancer types that have revealed similar molecular classification within and across tissues of origin [5], [7]. This means that the discriminative molecular features for one cancer classifier will most likely be relevant for other cancer types. Our pretrained model will have already learned the complex types of genetic alterations and genomic signatures collected from multiple cancer tissue types originating from different organs and can effectively function as a generic model for cancer classification.

Transfer learning using our GeneXNet model provides the capability to learn abstract feature representations from gene expressions of a specific multi-tumor cancer dataset and then transfer these representations to classify another type of cancer tumor. Our work is motivated from One-shot learning and Zero-shot learning methods used in Computer Vision which attempt to learn visual models of object categories using very little training data or even no training data at all in the case of unseen object categories [105], [106]. This is achieved by using deep learning models to learn abstract feature representations and then transferring the knowledge from previously learned categories and using it for detection of new categories without the need to learn the representations of new object categories from scratch [107].

Our proposed approach for performing transfer learning can be summarized as follows:

- 1) We build a multi-tissue multi-class classifier by training our GeneXNet model using ALL the underlying cancer tumor dataset which includes multiple organ sites covering multiple tumor types.

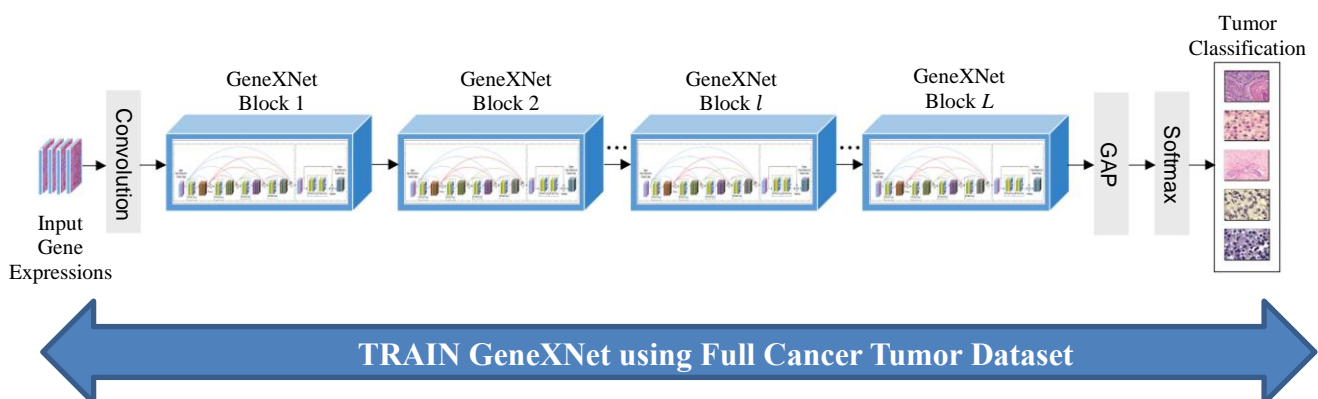


Figure 4.9 Transfer Learning – Training GeneXNet Model

- 2) We create a GeneXNet Base block by freezing the weights of the trained GeneXNet model and then removing the classification layers at the end of the network. The GeneXNet Base block will function as a feature extractor inside a new extended network model.

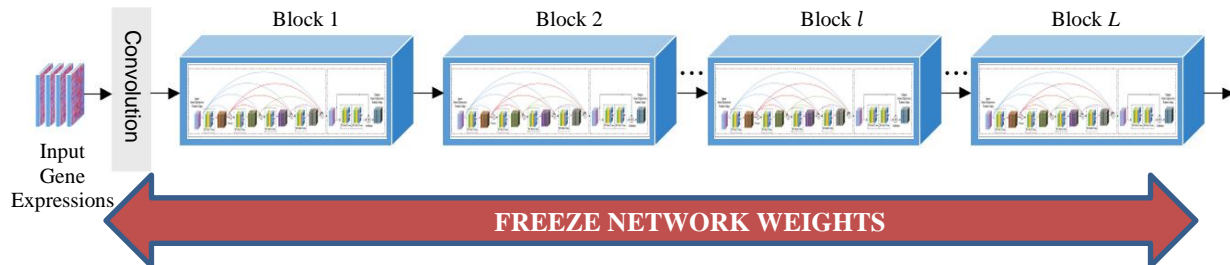


Figure 4.10 Transfer Learning – Freezing GeneXNet Weights

- 3) We create a new extended GeneXNet model by stacking the pre-trained GeneXNet Base block and adding a new randomly initialized classification layer.
- 4) We Re-train and Finetune the new extended network using a new cancer tumor dataset which might be lacking sufficient patient samples to be trained independently. The training is performed while freezing the original network layers that have already been pre-trained. We perform Finetuning by Un-Freezing some of the last layers in the GeneXNet Base Block and Re-training these layers again together with the new classification layers.

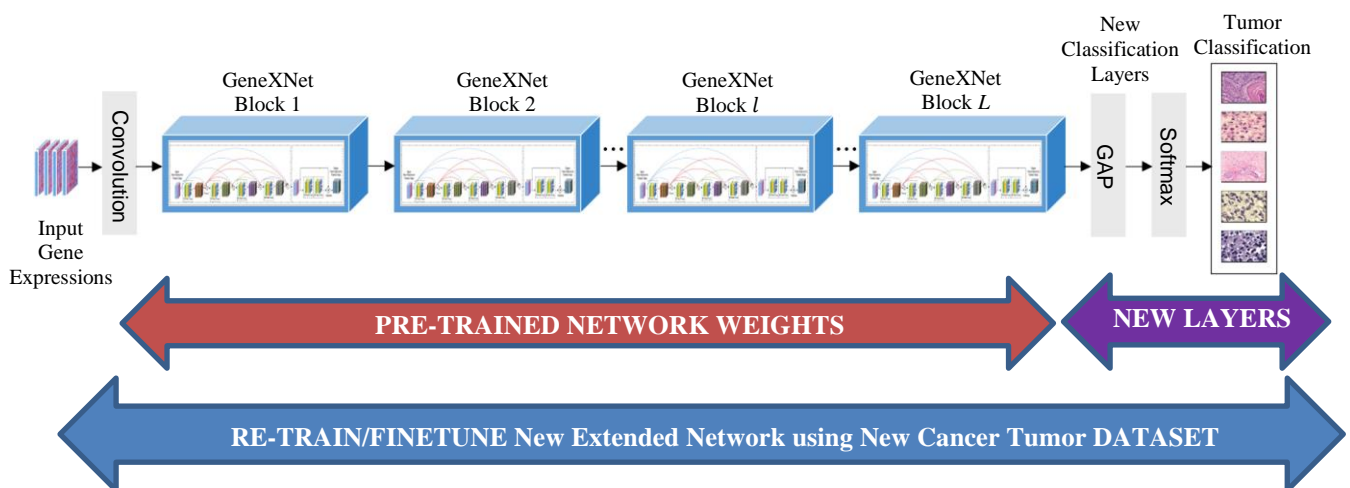


Figure 4.11 Transfer Learning – ReTraining and Finetuning Extended GeneXNet Model

4.8 Gene eXpression Network Training and Optimization

Training a deep multi-layer Convolutional Neural Network architecture like our Gene eXpression Network is a very complex optimization problem as it involves non-convex loss functions [67]. Adjusting the weights of the network to reduce the classification error requires an optimization algorithm capable of adapting the learning rate and leveraging information in the Hessian matrix of the loss function [62]. Among the challenges we faced in model optimization is the very high dimensional landscape of the network weight space resulting from training the network with whole-transcriptome gene expressions for every tumor sample. To overcome these problems, we define a multi-class cross-entropy loss function suitable for gene expression data and we train our model using mini-batch Stochastic Gradient Descent (SDG) with an adaptive learning rate optimization algorithm [62]. We implement several different optimization algorithms including Momentum [77], [78], AdaGrad [51], RMSprop [62] and Adam [50]. The following sections describe details of our network training and optimization for building a multi-tissue cancer classifier using whole-transcriptome gene expressions.

4.8.1 Optimization Objective and Loss Function

The objective of training our Gene eXpression Network is to find an optimal mapping function $y = f^*(x, W)$ by learning the network parameters W that would correctly classify our input gene expression data x , which represents the cancer tumor sample, to the correct output y , which represents the class of the cancer tumor type. Our network architecture as described in the previous sections, represents this complex mapping function and we need to train and optimize the network to learn the network parameters W that would result in the optimal classification performance. To learn the network parameters we follow the approach of learning conditional probability distributions using maximum likelihood [75]. In this approach, our network model represents a probability distribution $P(y | x, W)$ which is the conditional probability of predicting the correct tumor class given the tumor sample and the network parameters. We define the loss function as the Negative Log Likelihood (NLL) or Cross Entropy Loss between the training data and the network's class predictions since it represents the conditional probability of the tumor classes given the gene expression input. We define an overall optimization objective by defining an Error function $E(W)$ and then use gradient descent optimization to learn the parameters W that would reduce the error E to the optimal level when presented with the entire training data. Since the training data only represents a limited sample of the real cancer tumor distribution, our optimization objective is to minimize the expected loss on the training data.

We define the Error objective function $E(W)$ as the average over the training data given by:

$$E(W) = \frac{1}{N} \sum_{i=1}^N E_i[f(x_i, W), y_i] \quad (10)$$

where x_i is the i^{th} training sample, W represents the network's parameters to be learned, y_i is the target output, $f(x_i, W)$ represents the predicted output by the network and E_i represents the loss function for the i^{th} training sample.

Since our target is to build a Multi-Tissue classifier then we need to choose both the prediction function f and the loss function E_i which are suitable for a multiclass classification problem. We therefore define the output prediction function f_k using the softmax function as:

$$f_k = \text{softmax}(z)_k = P(\text{Class}_k | x_i, W) = \frac{e^{z_k}}{\sum_j^C e^{z_j}} \quad (11)$$

$$z_k = \log P(y = k | x_i, W) \quad (12)$$

where f_k represents the output prediction for the k^{th} tumor class, C represents the number of tumor classes, z_k is the k^{th} element in the output vector which represents the unnormalized log probability of the k^{th} class. The output of the softmax is a vector with each element having the normalized class probability. The advantage of the softmax is that it is a form of multiclass logistic regression and produces the output predictions in the form of a valid probability distribution over the number of classes.

We then define loss function E_i using the cross-entropy loss as:

$$E_i = - \sum_{j=1}^C t_k \cdot \log(f_k) \quad (13)$$

where t_k represents the k^{th} element of the target output vector for class k using a 1-of- C coding scheme such that all elements of the vector are zeros except for the k^{th} element which equals one. Since only a single term equals one, then the cross-entropy loss can be written as:

$$E_i = -\log \left(\frac{e^{z_k}}{\sum_j^C e^{z_j}} \right) \quad (14)$$

where E_i represents the cross-entropy loss for the i^{th} training sample and z_k is the k^{th} element in the output vector which represents the unnormalized log probability of the k^{th} class.

We can then define the gradient g of the Error E with respect to the network parameters W as:

$$g = \nabla_W E(W) = \frac{1}{N} \nabla_W \sum_{i=1}^N E[f(x_i, W), y_i] \quad (15)$$

4.8.2 Optimization Algorithms with Accelerated Gradient & Adaptive Learning

We use mini-batch Stochastic Gradient Descent (SDG) optimization for training our Gene eXpression Network. The motivation in using SDG is that it is very well suited for training our deep network given the high dimensionality of the gene expression data which includes a very large number of features representing the genes across the whole transcriptome. Performing gradient descent optimization using a mini batch of samples instead of the traditional batch training using the entire training data is more computationally efficient for large data sets. It has also been shown to perform a regularizing effect due to the noise it adds to the learning process [76]. We can obtain an unbiased estimate of the gradient by sampling a mini batch of tumor samples drawn i.i.d from the training data and calculating the average gradient on the mini batch. We then update the network parameters W in the direction of the gradient g to optimize the generalization error using the following update:

$$W = W - \eta_t \nabla_W E(W) \quad (16)$$

where η_t is the learning rate at iteration t which is also a parameter that can change across training iterations. The challenge with this update is that choosing the right learning rate is very difficult. If we choose a very small learning rate, then training will be very slow and if it is too large it will not guarantee convergence where the error function can fluctuate around the local minimum. Learning rate scheduling is one common approach to solve this problem by updating the learning rate at certain intervals based on specific criteria. But at the same time these updates have to be defined before the training and are not adaptive based on the tumor samples. Another big challenge that we faced in optimization for our network is the non-convex nature of the error function and the very high dimensional landscape of the network weight space which causes the optimization algorithm to suffer the presence of Local Min, Plateaus, Saddle points and other flat regions [75]. To overcome all these challenges, we train our network by adopting and experimenting with a variety of different optimization algorithms which adopt accelerated gradient methods and adaptive learning rate methods which we describe in the following sections.

GeneXNet Optimization with Momentum

We experiment with applying the Momentum update [77] to accelerate the learning process by accumulating an exponentially decaying moving average of past gradients and continually moving in that direction. We implement Momentum by introducing a new velocity term v which controls the direction and speed of the parameter updates. The velocity is calculated as an exponentially decaying average of the negative gradient as:

$$v = \alpha v - \eta \nabla_w \left[\frac{1}{N} \sum_{i=1}^N E[f(x_i, W), y_i] \right] \quad (17)$$

$$W = W + v \quad (18)$$

where $\alpha \in [0,1)$ is a parameter which determines how fast the contributions of the previous updates for the gradient will decay exponentially. Momentum helps in solving the poor conditioning of the Hessian matrix and variance in applying the standard SDG by accelerating in the correct direction of the local minimum. Figure 4.12 illustrates the acceleration effect of momentum on SDG optimization [75], where the contour lines represent a poorly conditioned Hessian matrix. The red path represents the direction followed by momentum, while the black path represents the standard SDG which has a slower learning since it oscillates heavily before finally converging.

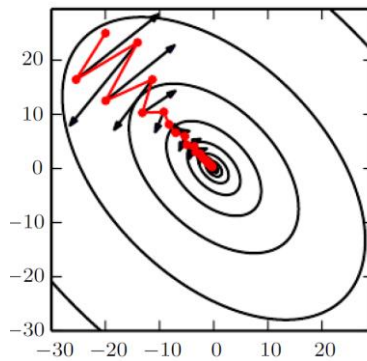


Figure 4.12 Accelerating Stochastic Gradient Descent Optimization with Momentum [72]

We also experiment with applying the Nesterov Momentum update [78] which is motivated by the accelerated optimization methods of Nesterov [79]. We apply a correction factor to the gradient calculation by performing it after the velocity update which is now defined as:

$$v = \alpha v - \eta \nabla_w \left[\frac{1}{N} \sum_{i=1}^N E[f(x_i, W + \alpha v), y_i] \right] \quad (19)$$

GeneXNet Optimization with AdaGrad

Since choosing the learning rate is one of the most complex hyperparameters, we also experiment with optimization algorithms with an adaptive learning rate. We apply the AdaGrad algorithm [51] which adapts the learning rate of all model parameters individually by performing large updates for infrequent parameters and smaller updates for the frequent ones. This adaptive learning characteristic is very important for training our network since it has been shown to perform well on sparse data [80] which is one of the big challenges in our underlying gene expression dataset.

We apply this adaptive learning to our Gene eXpression Network by updating the learning rate for each parameter and scaling it at a rate which is inversely proportional to the historical values of the gradient as follows:

$$W_{t+1} = W_t - \frac{\eta}{\sqrt{G_t + \varepsilon}} \odot g_t \quad (20)$$

where η is the learning rate for each parameter at each iteration t , ε is a constant that avoids division by zero, G_t is a matrix having each diagonal element as the sum of squares for the gradients with respect to each parameter for all previous iterations and \odot represents element wise matrix vector multiplication. The outline of the algorithm we implement is as follows:

Algorithm 4.1 GeneXNet Optimization with AdaGrad	
1	Input: W_0, η, ε initial network parameters, learning rate, const
2	Output: W GeneXNet optimized network parameters
3	Init $G_0 = 0$
4	while (Convergence criteria is <i>false</i>):
5	Read minibatch of N cancer tumor samples (x_i, y_i)
6	Calculate gradient $g_t = \frac{1}{N} \nabla_W \sum_{i=1}^N E[f(x_i, W), y_i]$
7	Calculate accumulated gradient $G_t = G_t + g_t \odot g_t$
8	Update network parameters $W_{t+1} = W_t - \frac{\eta}{\sqrt{G_t + \varepsilon}} \odot g_t$
9	end while

GeneXNet Optimization with RMSProp

We also experiment with the RMSProp algorithm [62] which is a modified version of AdaGrad since it has been shown to perform better on non-convex loss functions. This is achieved by calculating the gradient across iterations as an exponentially decaying average of all past gradients which allows discarding the historical values and converge more rapidly. We add a new parameter δ to configure the moving average and calculate the accumulated gradient as:

$$G_t = \delta \cdot G_t + (1 - \delta) g_t \odot g_t \quad (21)$$

We also combine the update with the Nesterov momentum as before to accelerate the learning process by adding an additional parameter α and calculating the velocity as:

$$v = \alpha v - \frac{\eta}{\sqrt{G_t}} \odot g_t \quad (22)$$

The outline of the algorithm we implement is as follows:

Algorithm 4.2 GeneXNet Optimization with RMSProp	
1	Input: $W_0, v_0, \eta, \delta, \alpha$ initial network parameters, initial velocity, learning rate, rate of decay, momentum
2	Output: W GeneXNet optimized network parameters
3	Init $G_0 = 0$
4	while (Convergence criteria is <i>false</i>):
5	Read minibatch of N cancer tumor samples (x_i, y_i)
6	Calculate gradient $g_t = \frac{1}{N} \nabla_W \sum_{i=1}^N E[f(x_i, W), y_i]$
7	Calculate accumulated gradient $G_t = \delta \cdot G_t + (1 - \delta) g_t \odot g_t$
8	Calculate velocity $v = \alpha v - \frac{\eta}{\sqrt{G_t}} \odot g_t$
9	Update network parameters $W_{t+1} = W_t + v$
10	end while

GeneXNet Optimization with Adam

The final optimization algorithm we experiment with is the Adaptive Moment Estimation (Adam) [50]. For this optimization method we combine the updates from both the previous optimizations methods of RMSProp and momentum by calculating two different averages of past gradients to be used in the update term. The first is calculating an estimate of the first order moment by calculating an exponentially decaying average of past gradients as in momentum. The second is calculating an estimate of the second order moment by calculating an exponentially decaying average of past squared gradients as in RMSProp:

$$m_t = \delta_1 \cdot m_{t-1} + (1 - \delta_1)g_t \quad (23)$$

$$v_t = \delta_2 \cdot v_{t-1} + (1 - \delta_2)g_t^2 \quad (24)$$

where m_t , v_t are the estimation of the first moment representing the mean and the second moment representing the uncentered variance of the gradients. In addition we also calculate bias corrections to the estimates of the first order moment and the uncentered second order moment as in [50] to remove the bias of these values towards zero. The outline of the algorithm we implement is as follows:

Algorithm 4.3 GeneXNet Optimization with Adam

1	Input: $W_0, \eta, \epsilon, \delta_1, \delta_2$ initial network parameters, learning rate, constant, 1 st moment rate of decay, 2 nd moment rate of decay
2	Output: W GeneXNet optimized network parameters
3	Init 1 st and 2 nd moment variables $m_t = 0, v_t = 0$
4	while (Convergence criteria is <i>false</i>):
5	Read minibatch of N cancer tumor samples (x_i, y_i)
6	Calculate gradient $g_t = \frac{1}{N} \nabla_W \sum_{i=1}^N E[f(x_i, W), y_i]$
7	Calculate 1st moment $m_t = \delta_1 \cdot m_{t-1} + (1 - \delta_1)g_t$
8	Calculate 2nd moment $v_t = \delta_2 \cdot v_{t-1} + (1 - \delta_2)g_t^2$
9	Correct bias of 1 st moment $\tilde{m}_t = \frac{m_t}{1 - \delta_1^t}$
10	Correct bias of 2 nd moment $\tilde{v}_t = \frac{v_t}{1 - \delta_2^t}$
11	Update network parameters $W_{t+1} = W_t - \eta \frac{\tilde{m}_t}{\sqrt{\tilde{v}_t + \epsilon}}$
12	end while

4.9 Visualizing Genomic Relationships of Gene Expressions Across Multiple Cancer Tumors

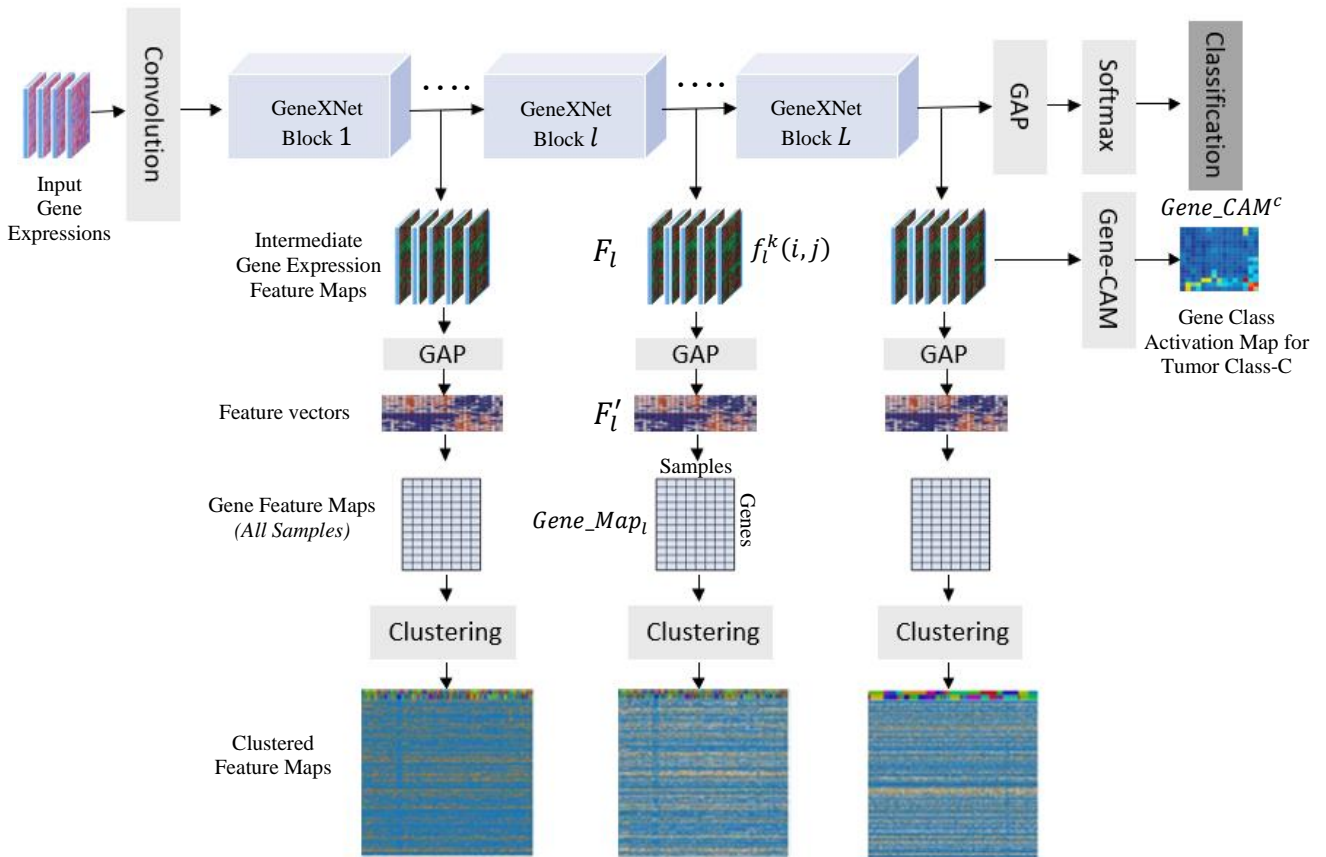


Figure 4.13 Visualizing genomic relationships of gene expressions

One of the challenges in using deep learning for disease diagnosis, is that deep networks are conceived as “black boxes” without much interpretation on how these complex models make their decisions [53]. Extensive work has been done to introduce novel visualization techniques for deep networks to help understand and interpret their record breaking performance in computer vision tasks [52], [53], [65]. The output from these techniques can be interpreted by non-experts when studied in conjunction with image or video datasets because they are visually comprehensible. Unfortunately, these methods are not directly applicable to genomic datasets such as gene expressions, since they cannot be visually rendered in a human-friendly form that allows easy interpretations. Our learning system architecture can contribute in solving this problem, since it is designed to take full potential of next generation sequencing technologies that produce datasets with detailed molecular characterizations of thousands of tumors using genome-wide platforms.

We introduce two visualization procedures for better understanding how our proposed deep network is performing cancer classification across multiple tumor types. Our methods are inspired from the work used to visualize intermediate feature activations for CNNs used in image classification [65]. We also build on the methods for Class Activation Maps (CAM) [52], [53] which visualize heatmaps of class activations for deep networks used in image classification and captioning. The outline of the procedures is shown in Figure 4.13 and described in the next sections. The results of applying these procedures to the underlying dataset is described in the experiments.

4.9.1 Visualizing Class-Discriminative Localization Maps of Gene Expressions

We introduce a visualization method which uses the gradient information flowing into the last convolutional layer of the GeneXNet model to produce gene localization maps highlighting the important regions in the gene expressions which influenced the resulting tumor class prediction. The gene expression data used to train the network is sparse and very high in dimensionality since it represents a snapshot of the whole transcriptome rather than a predetermined subset of genes. By identifying a class-discriminative localization map in the gene expressions, we can identify the subset of genes driving cancer progression and resulted in the model's tumor class prediction. We refer to this localization map as a Gene-Class-Activation-Map (Gene-CAM). For each tumor type, the Gene-CAM is a representation of the discriminative genes used by the network to correctly classify the tumor. The procedure can be summarized as follows:

We build a multi-tissue classifier by training our GeneXNet model with the genomic signatures of multiple tumor types across multiple sites using the underlying dataset. We group the data by tumor type and feed the trained network with each of the samples one by one to produce a prediction. For a GeneXNet with L blocks, the network will produce a set of intermediate activation feature maps as the output of each block. Let F_l represent the output feature maps of the l^{th} block with dimensions (width: X_l , height: Y_l , depth: D_l). This volume represents the molecular features learned by the network that will be activated when matched with similar patterns in the input gene expressions of a given tumor sample.

Let $f_L^k(i, j)$ represent the k^{th} feature map for the last block at special location (i, j) . Since the network uses Global Average Pooling (GAP) [54] before the final Softmax layer to calculate the spatial average of the feature maps, then the classification score s^c for tumor type c which is used as input to the softmax can be written as:

$$s^c = \sum_k^{D_L} w_k^c \sum_i^{X_L} \sum_j^{Y_L} f_L^k(i, j) \quad (25)$$

where c is the tumor class, w_k^c represents the weights for class c with respect to feature map k and D_L is the number of feature maps in the last block before the GAP layer each with width X_L and height Y_L .

To generate the Gene-Class Activation Map, we redefine the weights of each feature map with respect to a class as α_k^c by computing the gradient of the score of each class with respect to each feature map as follows:

$$\alpha_k^c = \frac{1}{X_L \cdot Y_L} \sum_i^{X_L} \sum_j^{Y_L} \frac{\partial s^c}{\partial f_{ij}^k} \quad (26)$$

where the new weights α_k^c represent the importance of each feature map for class discrimination. The Gene-Class-Activation-Map (Gene-CAM) is then calculated as:

$$Gene_CAM^c(i, j) = ReLU \left[\sum_k^{D_L} \alpha_k^c \cdot f^k(i, j) \right] \quad (27)$$

The resulting map with dimensions (X_L, Y_L) represents a gene localization for the given tumor sample that captures the discriminative regions in the gene expression input matrix which influenced the prediction of the tumor class. The ReLU [49] is applied to obtain only the features that have a positive contribution to the correct class [53].

Finally, to visualize the Gene-CAM we resize it using up-sampling and overlay it against the input gene expression matrix. The resulting heatmap highlights the important regions in the gene expression input matrix which in turn helps identify the subset of genes that are possibly influencing the Cancer tumor and resulted in the model's prediction.

4.9.2 Visualizing Molecular Clusters of Intermediate Feature Maps

We introduce a visualization procedure for observing the evolution of molecular clusters formed by intermediate gene expression feature maps learned by the network. The genetic signatures learned by the feature maps in the deep layers make the network capable of representing complex genetic alterations shared by tumors across different tissue types.

Visualizing the molecular clusters of gene expressions provides more insight on how the network is learning small meaningful relationships between the genes which in turn describe the characteristic influencing the Cancer tumor. We demonstrate how this visualization provides the opportunity to study the genomic relationships of gene expressions across multiple cancer tissue types. This is motivated by recent studies which have performed an integrated multiplatform analysis across multiple cancer types that have revealed molecular classification within and across tissues of origin [5], [7]. The procedure can be summarized as follows:

As in the previous section, for a GeneXNet with L blocks, let F_l represent the output feature maps of the l^{th} block. Let $f_l^k(i, j)$ represent the k^{th} feature map for the l^{th} block at special location (i, j) . We apply Global Average Pooling (GAP) [54] to each of the intermediate feature maps after each block to convert the volume F_l into a 1-dimensional feature vector F'_l with dimensions $(1, 1, D_l)$ as follows:

$$f_l'^k(i, j) = \frac{1}{X_l \cdot Y_l} \sum_i^{X_l} \sum_j^{Y_l} f_l^k(i, j) \quad (28)$$

$$F'_l = [f_l'^k(i, j)] \quad \forall k \in \{1, \dots, D_l\} \quad (29)$$

where D_l is the number of feature maps in the l^{th} block each with width X_l and height Y_l . The feature vector F'_l represents the spatial average of the feature maps produced by each filter in the convolutional layer. The intuition behind using GAP is due to its ability to produce a generic localizable deep representation of the features which can be used for class discrimination [53].

We stack together all the feature vectors at the l^{th} block across all N tumor samples to produce what we refer to as a *Gene Feature Map* ($Gene_Map_l$) of dimensions (D_l, N) :

$$Gene_Map_l = [F'_l(n)^T] \quad \forall n \in \{1, \dots, N\} \quad (30)$$

The resulting matrix stores the collective class-discriminative localization maps for the gene expressions at the l^{th} block across all the tumor types. It also represents the collective genetic signatures learned by the feature maps shared by tumors across different organ sites.

Finally, we perform a consensus hierarchical clustering [70] of the gene feature map $Gene_Map_l$ to generate a $Gene_Cluster_Map_l$ which is a molecular clustering that groups

each of the tumor types together based on the class discriminative gene localizations extracted from the gene expressions. Consensus clustering is specifically tailored for gene expression data and is based on resampling to reach a consensus across multiple runs of a clustering algorithm and assess the stability of the discovered clusters [69].

By visualizing a heatmap of the resulting clusters, we can observe the evolution of molecular clusters formed by intermediate gene expression feature maps learned by the network. Visualizing the molecular clustering helps in revealing the genomic relationships and high-level structures of gene expressions across multiple cancer tumor types that appeared influential in the cancer tumor progression beyond the standard grouping by anatomical organ site. The results of applying the visualization procedures to the underlying dataset are described in the experiments.

CHAPTER 5

5. EXPERIMENTS

5.1 Datasets

Our objective was to design a comprehensive multi-tissue Cancer classifier capable of detecting complex types of genetic alterations by learning the genomic signatures of whole-transcriptome wide gene expressions across multiple cancer tissue types. To achieve this objective, the datasets we selected for our experiments included a total of 11,093 human samples for mRNA gene expression quantification, which were collected from 26 different human anatomical sites of origin and covering 33 different Cancer tumor types. The datasets were obtained from “The Cancer Genome Atlas” (TCGA) [38] and generated by means of Total RNA sequencing using Illumina based systems [5]. Each individual human sample represents the whole transcriptome and includes a total of 60,483 genes annotated against a reference genome. The patients included both males and females and the biospecimens were collected from tumor tissue, adjacent normal tissue and normal whole blood samples. Table 5.1 shows a listing of the 33 cancer tumor types we used in our experiments together with the associated human sites of origin and the number of human samples available for each tumor type. One of the biggest challenges in using this dataset is the very small number of human samples in each of the tumor types, compared to the very large number of genes. Most of the tumor types only have several hundred samples and some even have less than a hundred samples while we have a total of 60,483 genes for each sample.

TCGA is a comprehensive atlas of cancer genomic profiles which includes the molecular characterization of over 20,000 primary cancer and matched normal samples [38]. TCGA uses next generation sequencing (NGS) methods such as DNA and RNA sequencing to generate cancer profiles in multiple various genome-wide platforms including DNA (DNA methylation, exome sequencing and copy number), RNA (mRNA and microRNA sequencing) and other forms of relevant cancer sets of proteins [5]. The RNA-Sequencing experiment consists of isolating RNA, converting it to complementary DNA (cDNA), then preparing the sequencing library and sequencing it on a NGS platform [22]. The expression of genes are quantified by generating the FASTQ-format files which contain reads sequenced from the NGS platform and then aligning these reads to an annotated reference genome [38].

Table 5.1 Multi-tissue Cancer Tumor Dataset used in our experiments. The dataset includes 33 different cancer tumor types across 26 different anatomical organ sites.

Organ Site	Cancer Tumor Type(s)	Total Samples	Organ Site	Cancer Tumor Type(s)	Total Samples
Adrenal Gland	Adrenocortical carcinoma (ACC), Pheochromocytoma and Paranganglioma (PCPG)	265	Liver	Liver hepatocellular carcinoma (LIHC)	424
Bile Duct	Cholangiocarcinoma (CHOL)	45	Lung	Lung adenocarcinoma (LUAD), Lung squamous cell carcinoma (LUSC)	1145
Bladder	Bladder Urothelial Carcinoma (BLCA)	433	Lymph Nodes	Lymphoid Neoplasm Diffuse Large B-cell Lymphoma (DLBC)	48
Bone Marrow	Acute Myeloid Leukemia (LAML)	151	Ovary	Ovarian serous cystadenocarcinoma (OV)	379
Brain	Glioblastoma multiforme (GBM), Brain Lower Grade Glioma (LGG)	703	Pancreas	Pancreatic adenocarcinoma (PAAD)	182
Breast	Breast invasive carcinoma (BRCA)	1222	Pleura	Mesothelioma (MESO)	86
Cervix	Cervical squamous cell carcinoma and endocervical adenocarcinoma (CESC)	309	Prostate	Prostate adenocarcinoma (PRAD)	551
Colorectal	Colon adenocarcinoma (COAD), Rectum adenocarcinoma (READ)	698	Skin	Skin Cutaneous Melanoma (SKCM)	472
Esophagus	Esophageal carcinoma (ESCA)	173	Soft Tissue	Sarcoma (SARC)	265
Eye	Uveal Melanoma (UVM)	80	Stomach	Stomach adenocarcinoma (STAD)	407
Head and Neck	Head and Neck squamous cell carcinoma (HNSC)	546	Testis	Testicular Germ Cell Tumors (TGCT)	156
Kidney	Kidney Chromophobe (KICH), Kidney renal clear cell carcinoma (KIRC), Kidney renal papillary cell carcinoma (KIRP),	1021	Thymus	Thymoma (THYM)	121
			Thyroid	Thyroid carcinoma (THCA)	568
			Uterus	Uterine Corpus Endometrial Carcinoma (UCEC), Uterine Carcinosarcoma (UCS)	643
			(ALL Sites)	(All Tumors)	11,093

The gene expression values we used to generate our datasets are based on the read counts measured on a gene level and then normalized using the Fragments Per Kilobase of transcript per Million mapped reads (FPKM) [38]. The formula for FPKM normalization is defined in (13) where RC_g are the number of genes mapped to the gene, RC_{pc} are the number of reads mapped to all protein coding genes and L is the length of the gene in base pairs.

$$Gene\ Expression\ Quantification = FPKM = \frac{RC_g * 10^9}{RC_{pc} * L} \quad (31)$$

We transform the gene expression data in a format that makes it suitable as input to our model. We represent the gene expressions in an equivalent 2D matrix of dimensions (60,483, 11,093) as in Figure 4.2. Each column represents a human sample and each row represents a single gene. We convert each human sample into an equivalent 3D volume of genes with dimensions (142, 142, 3). The full dataset for all the 11,093 samples is represented by the 4D input matrix with dimensions (11,903, 142, 142, 3) to make it suitable as input to our GeneXNet model.

5.2 Classification Experiments

Our experiments demonstrate how the design of our Gene eXpression Network (GeneXNet) can be used as a general end-to-end learning system for classification across multiple cancer tissue types without performing the prerequisite process of gene feature selection. We also demonstrate how our model can specifically target the complex nature of the whole-transcriptome gene expression data and addresses the lack of training samples, without suffering from severe overfitting in comparison to using the current state-of-the-art deep CNN models which have been designed specifically for computer vision tasks.

We perform several different multi-class and binary classification tasks. For binary classification we predict whether the given sample represents a tumor versus a normal tissue. For multi-class classification we predict for a given sample the type of Cancer tumor within each anatomical site of origin. The following are details of the classification experiments:

5.2.1 Experiment 1 - Multi-tissue Multi-class classification

We build a multi-tissue multi-class classifier by training our model using ALL the data which includes the genomic signatures from 26 organ sites covering 33 tumor types.

5.2.2 Experiment 2 - Multi-Tumor Binary classification

We build a multi-tumor binary classifier for each individual organ site. We group the data by each individual site and train each model separately. For this task we selected the organ sites that relatively had the greatest number of samples compared to the other sites (at least 400 samples per site). These included 11 sites as shown in Table 5.2.

5.2.3 Experiment 3 - Comparison between Transfer Learning and Full Training

We repeat the second experiment, but this time we perform performing *transfer learning* using the weights of the pre-trained model from the first experiment. The objective was to compare the performance between transfer learning using a pre-trained model and full training. We evaluate whether finetuning the pre-trained model was able to achieve a comparable performance in comparison to models which were fully trained.

5.2.4 Experiment 4 – Transfer Learning for Tumors Lacking Sufficient Training Data

We use transfer learning to build binary classifiers for the organ sites with the least number of samples which did not have sufficient data to be trained independently. We start with the pre-trained model from the first task and use the data from each site separately to finetune the pre-trained model. These included Bile Duct and Esophagus which only had 45 and 147 samples respectively.

Table 5.2 Multi-Tumor Binary Classification Dataset used in our experiments. The dataset includes 11 Individual Organ Sites that relatively had the greatest number of samples

Organ Site	Cancer Tumor Type(s)	Total Samples
Bladder	Bladder Urothelial Carcinoma (BLCA)	433
Breast	Breast invasive carcinoma (BRCA)	1222
Colorectal	Colon adenocarcinoma (COAD), Rectum adenocarcinoma (READ)	698
Head & Neck	Head and Neck squamous cell carcinoma (HNSC)	546
Kidney	Kidney Chromophobe (KICH), Kidney renal clear cell carcinoma (KIRC), Kidney renal papillary cell carcinoma (KIRP)	1021
Liver	Liver hepatocellular carcinoma (LIHC)	424
Lung	Lung adenocarcinoma (LUAD), Lung squamous cell carcinoma (LUSC)	1145
Prostate	Prostate adenocarcinoma (PRAD)	551
Stomach	Stomach adenocarcinoma (STAD)	407
Thyroid	Thyroid carcinoma (THCA)	568
Uterus	Uterine Corpus Endometrial Carcinoma (UCEC), Uterine Carcinosarcoma (UCS)	643

5.2.5 Experiment 5 – Comparison between GeneXNet & State-of-the-art models

We evaluate the multi-tissue classification performance of our GeneXNet model in comparison with some of the current state-of-the-art deep CNN models specifically designed for computer vision tasks. We perform the same multi-class classification in experiment 1 using all the data but replacing our model with the publicly available implementations of ResNet [46],[58], DenseNet [41], NasNet [40] and MobileNet [39],[43].

5.2.6 Experiment 6 – Comparison between different GeneXNet Architectures

We evaluate the multi-tissue classification performance of our GeneXNet model with different architectures and sizes by tuning the parameters θ_D , θ_R with values (0, 0.25, 0.5 and 1) and θ_k with values (32, 64). These parameters define the percentage of dense and residual sub-blocks in the network and the number of filters used in the convolution layers as described in section 4.5.

5.3 Training, Optimization and Evaluation

We treat our dataset as a very scarce and valuable resource since the biggest challenge in using the underlying dataset to train our deep models is the very small number of available tumor samples compared to the very large number of genes. We use stratified random sampling to divide our dataset into 85% for training/validation and 15% for final testing. We train all our models using stratified k-fold cross-validation experimenting with different fold sizes. We use the validation data to optimize the hyperparameters of our models while the test data is strictly used only once to evaluate the final performance of each model.

Training a deep multi-layer CNN architecture like GeneXNet is a very complex optimization problem as it involves non-convex loss functions [67]. Adjusting the weights of the network to reduce the classification error requires an optimization algorithm capable of adapting the learning rate and leveraging information in the Hessian matrix of the loss function [62]. Among the challenges we faced in model optimization is the very high dimensional landscape of the network weight space resulting from training the network with the whole-transcriptome wide gene expressions for every tumor sample. To overcome these problems, we train our model using mini-batch Stochastic Gradient Descent (SDG) with an adaptive learning rate optimization algorithm [62]. We experiment with Adam [50], AdaGrad [51] and RMSprop [62]. We start with a learning rate of $1e^{-4}$ and divide it by half when the validation loss plateaus for more than 50 epochs.

We evaluate the classification performance of our GeneXNet models using the receiver operating characteristics (ROC) curves [68]. For all our experiments across each of the cancer tumor types, we report the average classification accuracy and ROC Area Under the Curve (AUC) on the Test dataset. The ROC AUC has an advantage of being less sensitive to changes in class distribution as it summarizes the performance over a range of tradeoffs between the true positive and false positive rates [68]. To overcome any potential impact on the classification performance due to class imbalance, we experimented with two different methods for addressing class imbalance. We used Synthetic Minority Over-sampling [109] and Adaptive Synthetic Sampling [110].

ALL DATA (26 ORGAN SITES, 33 TUMOR TYPES)					
TRAINING & VALIDATION (85 %)					TEST (15 %)
TRAINING (68%)				VAL (17%)	
SPLIT 1	TRAIN	TRAIN	TRAIN	TRAIN	
SPLIT 2	TRAIN	TRAIN	TRAIN	VAL	
SPLIT 3	TRAIN	TRAIN	VAL	TRAIN	
SPLIT 4	TRAIN	VAL	TRAIN	TRAIN	
SPLIT 5	VAL	TRAIN	TRAIN	TRAIN	

Figure 5.1 Training with K-Fold Cross Validation

5.4 Results

Experiment 1 - Multi-tissue Multi-class classification

The results of the first experiment which performed multi-class classification using ALL the data including 26 organ sites covering 33 tumor types are shown in Table 5.4. Our GeneXNet model was able to achieve excellent results with an overall classification accuracy of 98.93% and a ROC AUC of 0.99 on the test dataset. The results show that our model achieved 100% accuracy on 14 different tumor types, even for some tumor types which had very little human samples such as: Bile Duct Cholangiocarcinoma (CHOL), Eye Uveal Melanoma (UVM) and Pleura Mesothelioma (MESO) which only had 45, 80 and 86 samples respectively.

Table 5.4 Results Of Multi-Tissue Classification Using 26 Organ Sites Covering 33 Tumor Types

Anatomical Site of Origin	Cancer Tumor Type(s)	Total Samples	Classification Accuracy (%)
Adrenal Gland	Adrenocortical carcinoma (ACC), Pheochromocytoma and Paraganglioma (PCPG)	265	100
Bile Duct	Cholangiocarcinoma (CHOL)	45	100
Bladder	Bladder Urothelial Carcinoma (BLCA)	433	98.46
Bone Marrow	Acute Myeloid Leukemia (LAML)	151	91.3
Brain	Glioblastoma multiforme (GBM), Brain Lower Grade Glioma (LGG)	703	100
Breast	Breast invasive carcinoma (BRCA)	1222	99.46
Cervix	Cervical squamous cell carcinoma and endocervical adenocarcinoma (CESC)	309	100
Colorectal	Colon adenocarcinoma (COAD), Rectum adenocarcinoma (READ)	698	99.05
Esophagus	Esophageal carcinoma (ESCA)	173	96.15
Eye	Uveal Melanoma (UVM)	80	100
Head and Neck	Head and Neck squamous cell carcinoma (HNSC)	546	100
Kidney	Kidney Chromophobe (KICH), Kidney renal clear cell carcinoma (KIRC), Kidney renal papillary cell carcinoma (KIRP),	1021	99.35
Liver	Liver hepatocellular carcinoma (LIHC)	424	98.44
Lung	Lung adenocarcinoma (LUAD), Lung squamous cell carcinoma (LUSC)	1145	99.42
Lymph Nodes	Lymphoid Neoplasm Diffuse Large B-cell Lymphoma (DLBC)	48	87.5
Ovary	Ovarian serous cystadenocarcinoma (OV)	379	98.25
Pancreas	Pancreatic adenocarcinoma (PAAD)	182	96.43
Pleura	Mesothelioma (MESO)	86	100
Prostate	Prostate adenocarcinoma (PRAD)	551	97.59
Skin	Skin Cutaneous Melanoma (SKCM)	472	98.59
Soft Tissue	Sarcoma (SARC)	265	100
Stomach	Stomach adenocarcinoma (STAD)	407	98.39
Testis	Testicular Germ Cell Tumors (TGCT)	156	100
Thymus	Thymoma (THYM)	121	100
Thyroid	Thyroid carcinoma (THCA)	568	97.67
Uterus	Uterine Corpus Endometrial Carcinoma (UCEC), Uterine Carcinosarcoma (UCS)	643	100
(ALL Sites)	(All Tumors)	11,093	98.93

Experiment 2 - Multi-Tumor Binary classification

The results of the second experiment which performed binary classification for the 11 selected individual organ sites are shown in Table 5.5. Our GeneXNet model was able to achieve 100% accuracy for 8 different tumor types and between 95.35% to 99.42% accuracy for the remaining tumors.

Table 5.5 Results Of Multi-Tumor Binary Classification For 11 Individual Organ Sites

Anatomical Site of Origin (Human Organ)	Total Samples	Full Training		Transfer Learning & Finetuning	
		Accuracy (%)	ROC AUC	Accuracy (%)	ROC AUC
Bladder	433	96.92	1.0	95.38	0.99
Breast	1222	98.37	0.998	98.37	1.0
Colorectal	698	100	1.0	100	1.0
Head and Neck	546	98.78	0.985	92.68	1.0
Kidney	1021	100	1.0	100	0.97
Liver	424	100	1.0	98.44	1.0
Lung	1145	99.42	1.0	99.42	0.94
Prostate	551	97.59	0.961	97.59	0.94
Stomach	407	96.77	0.979	96.77	0.88
Thyroid	568	95.35	0.981	93.02	1.0
Uterus	643	100	1.0	100	0.89
Bile Duct*	45	-	-	85.71	0.89
Esophagus*	173	-	-	92.31	0.99

Experiment 3 - Comparison between Transfer Learning and Full Training

The results of the third experiment which performed transfer learning are shown in Table 5.5. The results show that transfer learning managed to achieve excellent results which are comparable to the results achieved using full training.

Experiment 4 – Transfer Learning for Tumors with very little data

The results of the fourth experiment which performed transfer learning to build binary classifiers for organ sites which did not have sufficient data to be trained independently are shown in the last two rows of Table 5.5. Transfer learning was able to solve the problem for tumor sites such as Bile Duct and Esophagus which did not have sufficient data to be trained independently. By finetuning the pre-trained model, we were able to achieve 92.31% accuracy for Esophageal carcinoma (ESCA) and 85.71% accuracy for Bile Duct Cholangiocarcinoma (CHOL) despite that these sites only had 147 and 45 samples respectively.

Experiment 5 – Comparison between GeneXNet and State-of-the-art CNN models

The results of the fifth experiment for evaluating the performance of our GeneXNet model in comparison with state-of-the-art CNN models is shown in Table 5.6. A comparison between the ROC curves for the different models is shown in Figure 5.2. These results demonstrate that our GeneXNet model consistently outperformed other CNN models by a large margin. The classification accuracy achieved by our model is 98.93% which is significantly higher than the other models which achieve an accuracy below 37%. Figure 5.2 shows that our model produced a much higher ROC curve in comparison to the other models.

Table 5.6 Classification Performance Of GeneXNet In Comparison With State-Of-The-Art CNN Models

Network Model	Accuracy (%)	ROC AUC	Cross Entropy Loss
GeneXNet	98.93	0.99	0.06
ResNet-50 v2 [46]	36.96	0.86	4.9
DenseNet-121 [41]	22.33	0.79	6.09
NasNetMobile [40]	21.61	0.84	2.58
MobileNet v2 [39]	24.96	0.8	5.99

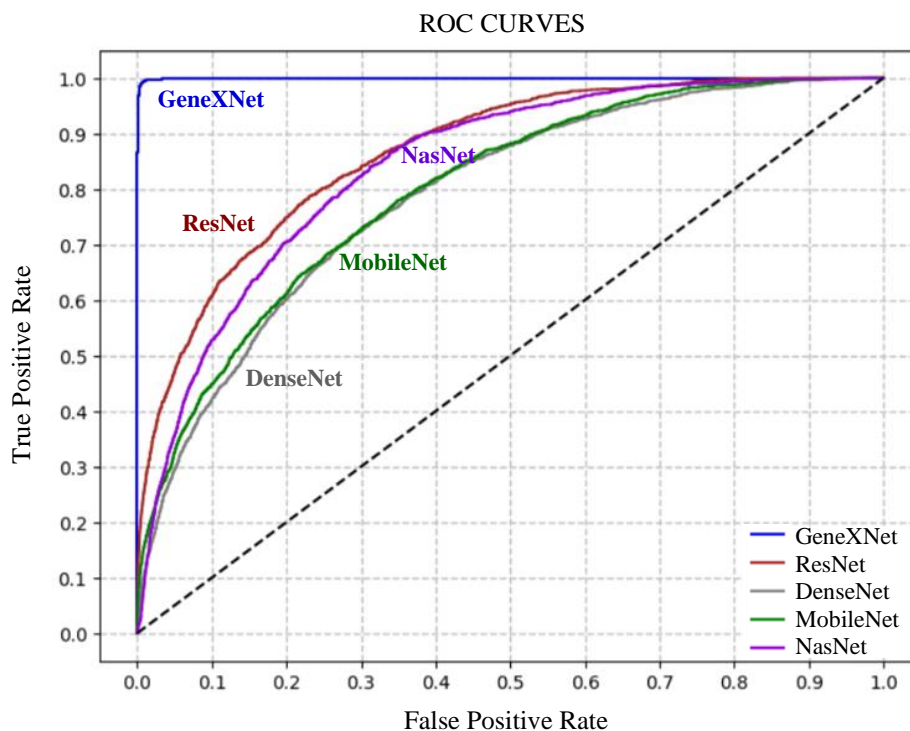


Figure 5.2 Comparison of ROC curves for Multi-Tissue classification between our GeneXNet model and state-of-the-art CNN models. Our model produced a much higher ROC curve and outperformed other models by a large margin.

To provide more insight on this degradation in performance for state-of-the-art models, Figure 5.3 shows a comparison between the training and validation curves for each model by plotting the cross-entropy loss across the training epochs. Figure 5.3 demonstrates that training these state-of-the-art models which were specifically designed for computer vision tasks, suffered from severe overfitting when presented with the underlying dataset that includes whole transcriptome gene expressions from multiple tumors types.

On the other hand, our GeneXNet model was able to achieve high accuracy in multi-tumor classification while avoiding overfitting. This ability is attributed to the architecture of our model that is designed specifically to target the complex nature of gene expressions and which incorporates both dense and residual learning layers that perform a regularizing effect which allows the network to overcome overfitting.

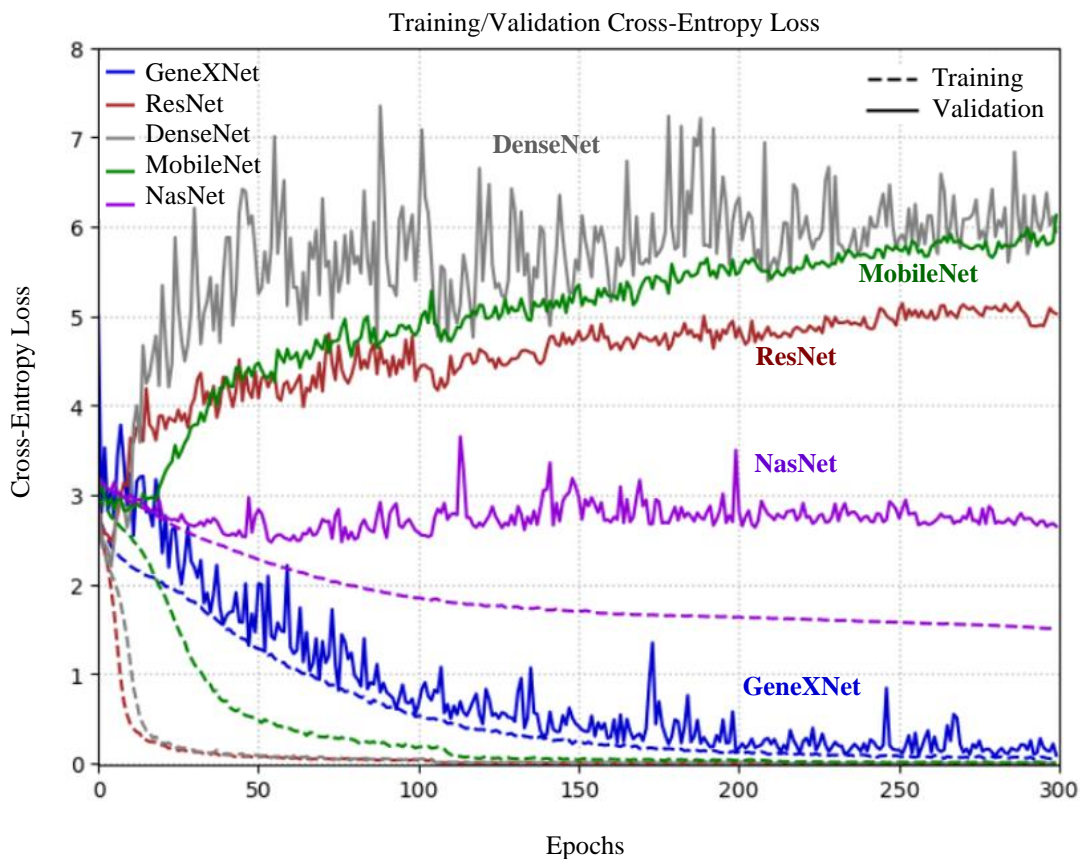


Figure 5.3 Comparison of training and validation Cross-Entropy Loss for Multi-Tissue classification, between GeneXNet and other models. Our model achieved minimum loss while other models suffered severe overfitting. Dashed curves are training and solid are validation.

5.5 Analysis of Classification Results

The results of our classification experiments have demonstrated how the design of our proposed Gene eXpression Network (GeneXNet) can be used as a general end-to-end learning system for classification across multiple cancer tissue types without performing the prerequisite process of gene feature selection. We demonstrated how our model can specifically target the complex nature of the whole-transcriptome gene expression data and addresses the lack of training samples, without suffering from severe overfitting in comparison to using the current state-of-the-art deep CNN models which have been designed specifically for computer vision tasks. Our model has allowed training deeper network architectures with complex data like whole-transcriptome gene expressions, despite the large number of genes. The experiments demonstrated that our model design which combines both dense and residual learning layers, performs a regularizing effect which helps avoid overfitting and degradation in performance as the network depth increases. This is achieved by means of re-using the gene expression feature maps learned by different layers, which increases the variation of input signals fed to subsequent layers since it represents the collective knowledge of the network. The connectivity of the dense layers provide each layer with more direct access to the gradients from the loss function and the original input signal, while the residual layers with identity mappings provide a direct path for information propagation in the forward and backward passes.

The results of our Transfer Learning experiments have demonstrated that the comprehensive genomic signatures learned by training our model using all the data allowed us to perform efficient transfer learning by using the pre-trained model as a generic feature extractor to build additional classifiers for any of the individual tumor sites, especially for the organ sites which were lacking sufficient patient samples to be trained independently. These results have demonstrated how transfer learning was able to solve one of the biggest challenges in cancer classification which is lack of patient samples. The experiment demonstrated that by reusing the weights of the pretrained GeneXNet model, we were able to use the same network for feature extraction on a different cancer tumor type. The experiments have also demonstrated that the discriminative molecular features for one cancer classifier were also relevant for other cancer types. The results demonstrated that our pretrained model was able to learn the complex types of genomic signatures collected from multiple cancer tissue types and that it was able to effectively function as a generic model for cancer classification.

5.6 Visualizing Class-Discriminative Localization Maps

We introduced a visualization method in section 4.10 to identify a class discriminative Gene-Class Activation Map (Gene-CAM) which is a localization map extracted from the gene expression input samples. We apply the visualization procedure to the underlying dataset to produce a Gene-CAM for each of the 33 individual tumor types and visualize them using heatmaps. Figure 5.4 shows the resulting heatmaps of four selected tumor types (Breast, Liver, Stomach and Uterus). We used a GeneXNet with 4 blocks which produces feature activation maps of dimensions (5, 5, 2048) after the 4th block. By mapping the resulting Gene-CAM to each input sample, the network was able to identify a subset of 75 discriminative genes. For visualization, we apply a threshold where each heatmap shows the top 20 genes influencing the underlying tumor across 20 randomly selected samples. The rows represent the genes and the columns represent the samples and the values are the normalized gene expression levels. The gene symbols are displayed on the right side of each row together with the percentage of samples which have also identified this gene in their Gene-CAM. Each map is a visual representation of the discriminative genes used by the network to correctly classify the tumor.

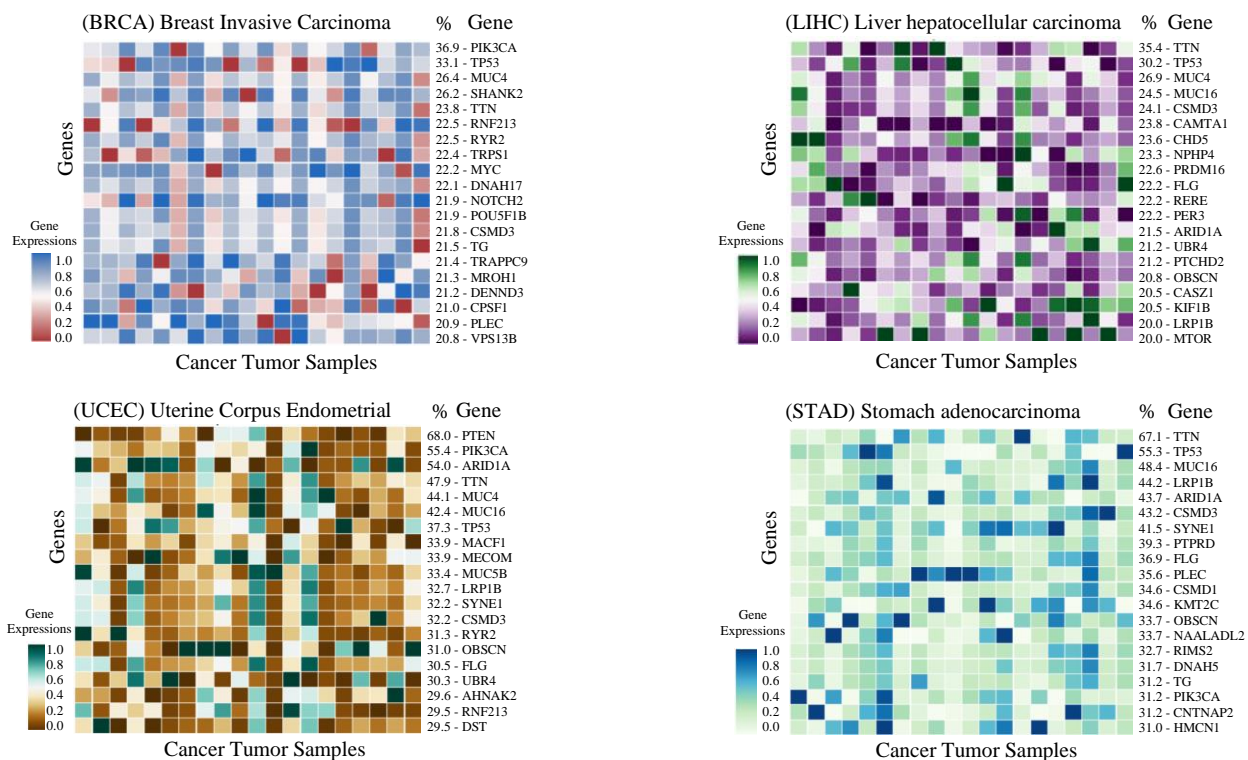


Figure 5.4 Visualizing class-discriminative localization maps highlighting the important regions in the gene expressions which influenced the tumor class prediction. Each map shows the top 20 genes across 20 random samples and is a visual representation of the discriminative genes used by our network to correctly classify the tumor. The rows represent genes, columns represent samples and the values are the gene expression levels.

5.7 Biological Significance of Visualizing Class-Discriminative Maps

The strength of our method is that the network was able to automatically identify a small subset of class-discriminative genes out of the total 60,483 genes originally included in each individual sample. What was also very interesting about these results is that the network automatically identified the TP53 gene as one of the top features common across all tumor types. This result implicitly validates our procedure since the TP53 is considered the most commonly mutated gene in all cancers which produces a protein that suppresses the growth of tumors [4].

We also observed from our experiments that some of the identified discriminative genes were also common in at least 30% of samples across different tumor types even though the tissues belonged to different organ sites. This subset includes: TP53, TTN, MUC16, LRP1B, CSMD3, PIK3CA, MUC4, RYR2, USH2A, FLG, PTPRD, CSMD1. These discriminative genes identified by the network have great biological significance for early cancer diagnosis. For example, the mutations of PIK3CA gene are one of the most common in Breast cancer and are reported in over one third of cases [112]. Mutations in TTN gene are associated with one of the most common inherited cardiac disorders known as Hypertrophic Cardiomyopathy (HCM) [111]. MUC16 has a biological role in the progression of Ovarian tumors and there has been substantial work to develop therapeutic approaches to eradicate Ovarian tumors by targeting MUC16 [113]. LRP1B is frequently mutated in Melanoma, Non-small Cell Lung cancer (NSCLC) and other types of tumors. LRP1B is also a potential contributor to the emergence of chemotherapy resistance while treating cancer patients [114]. CSMD3 was identified as the second most frequently mutated gene in Lung cancer after TP53 [4]. MUC4 is a membrane bound mucin gene responsible for progression of several cancers due to its anti-adhesive properties including Bile Duct, Breast, Colon, Esophagus, Ovary, Lung, Prostate, Stomach and Pancreas [111]. Mutations of RYR2 gene are a common cause of abnormal heart failures such as Catecholaminergic Polymorphic Ventricular Tachycardia (CPVT) [111]. PTPRD is frequently mutated in various types of cancer, including Glioblastoma, Melanoma, Breast and Colon [4]. CSMD1 has been found as a tumor suppressor in the development of Breast cancer [111]. To validate our visualization results, we compared the subset of discriminative genes identified by our network with the top mutated genes reported in the underlying dataset based on percentage of cases with simple somatic mutations [38]. We observed from this comparison that the set of discriminative genes identified by our network were also identified among the top mutated genes in 92% of the samples.

5.8 Visualizing Molecular Clusters of Intermediate Feature Maps

We introduced a visualization method in section 4.10 for observing the molecular clusters formed by intermediate gene expression feature maps learned by the network. We apply the method using all the underlying dataset which includes 11,093 samples for 26 organ sites across 33 tumor types. We used a GeneXNet with four blocks to produce a molecular clustering of the gene feature maps (*Gene_Cluster_Maps*) after each block. Each individual *Gene_Cluster_Map* represents a molecular clustering that groups the tumors by organ site based on the class discriminative gene localizations extracted from the gene expressions and learned by the network after each block. As outlined in Table 4.2, the output depth after each block is 256, 512, 1024 and 2048 respectively. Figure 5.5 shows a heatmap of the *Gene_Cluster_Map* for the last block filtered for clusters with at least 200 samples per cluster, which resulted in a total of 17 cluster groups comprising the 26 organ sites. The rows represent the gene localization feature maps, the columns represent the samples and the values are the normalized activations from the feature maps. The heatmap visually illustrates the genomic relationships and high-level structures of the cancer tumor types across the different sites of origin.

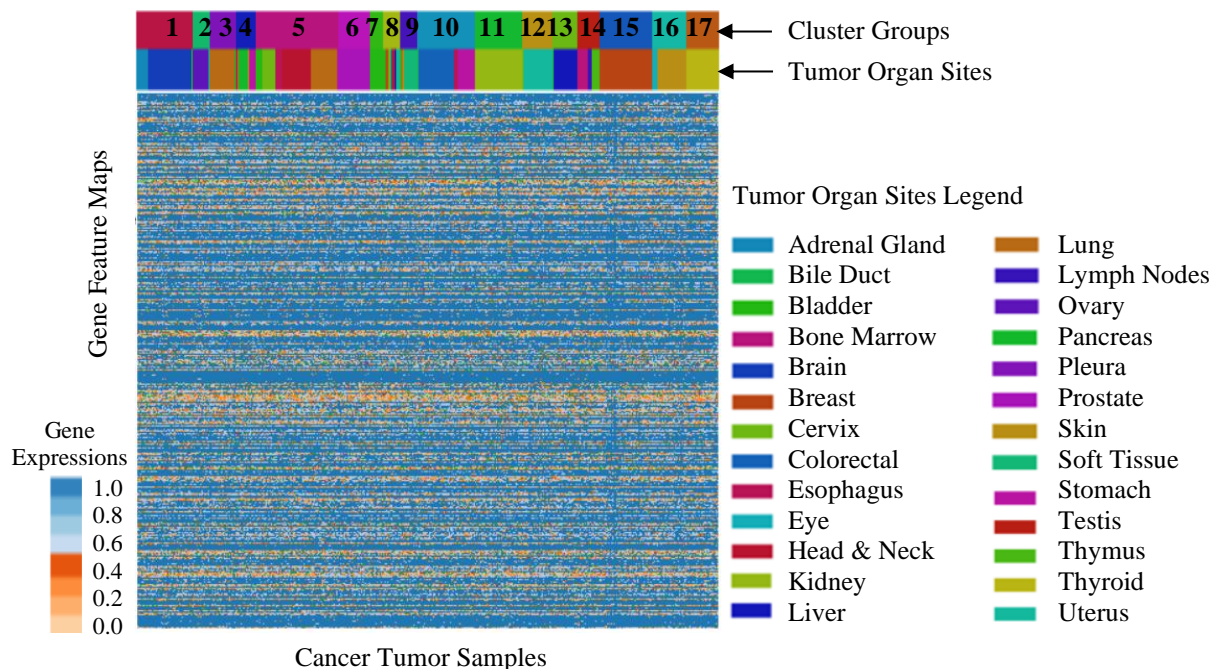


Figure 5.5 Visualizing molecular clusters of intermediate feature maps to reveal genomic relationships across multiple tumors that appeared influential in cancer progression. The heatmap shows a Gene Cluster Map of 17 cluster groups comprising 33 tumors across 26 organ sites. Rows represent gene feature maps, columns represent samples and the values are activations of feature maps. The heatmap visually illustrates the genomic relationships and high-level structures of the cancer tumor types across the different organ sites.

5.9 Biological Significance of Visualizing Molecular Clusters of Intermediate Feature Maps

We observed from our experiments that the number of cluster groups learned by the network in the *Gene_Cluster_Maps* decreases as we move towards the deep layers in the network. The feature maps generated after the first block seem to have little in common across the different tumor types which is evident by the very large number of resulting cluster groups. As we reach the final network block, we observed that the *Gene_Cluster_Map* has less number of clusters where more clusters have merged together to finally reach only 17 cluster groups. These results have great significance since they demonstrate that as we go deeper in the network, the gene feature maps become more abstract in the sense that they are less representative of the individual tumor samples and more representative of the tumor classes.

We further analyzed the resulting cluster groups in-terms of membership of tumor organ sites among the groups. We observed that although tissue site of origin was mostly a dominant factor for cluster formation, but some clusters also included tumor types across multiple different organs. We also observed that clusters were formed for tumor types which appeared to have similar organs or tissue characteristics. For example, Bile Duct and Liver tumors clustered together including Cholangiocarcinoma (CHOL) and Liver hepatocellular Carcinoma (LIHC). Brain and Nervous system tumors clustered together including Brain Lower Grade Glioma (LGG) and Glioblastoma multiforme (GBM). Kidney and Adrenal Gland tumors formed multiple clusters including Chromophobe (KICH), Renal Clear Cell (KIRC), Renal Papillary cell (KIRP) and Adrenocortical Carcinoma (ACC). Lymph Nodes and Bone Marrow tumors clustered together including Lymphoid Neoplasm Diffuse Large B-cell Lymphoma (DLBC) and Acute Myeloid Leukemia (LAML). Many small overlapping clusters formed together for Stomach, Colorectal, Esophagus and Pancreas tumors including STAD, COAD, READ, ESCA and PAAD. Finally, the remaining clusters were dominated by mostly tumors of a single organ site but also included less than 5% of other tumors types.

Visualizing the evolution of molecular clusters formed by intermediate gene feature maps, has demonstrated how our proposed GeneXNet is functioning as a comprehensive multi-tumor Cancer classifier. The network was capable of learning the complex molecular signatures and genetic alterations shared by tumors across different tissue types and organ sites. This also demonstrates how the network was able to perform efficient transfer learning by using the pre-trained models as a generic multi-tumor feature extractor to build additional classifiers for any individual tumor types especially for organ sites which were lacking sufficient patient samples to be trained independently.

CHAPTER 6

6. CONCLUSIONS

6.1 Motivation

The objective of our research has been to contribute in saving the lives of cancer patients through early cancer diagnosis and detection. Our work in cancer classification helps in directly solving one of the major challenges in cancer treatment, since patients are diagnosed at very late stages when appropriate medical interventions become less effective and full curative treatment is no longer achievable. To our knowledge, this is the first effort to develop a Multi-Tissue cancer classifier based on a full set of whole-transcriptome wide gene expressions collected from tumors across different tissue types without requiring a prerequisite process of gene feature selection. We have contributed in providing medical professionals with more confidence in using deep learning for medical diagnosis by providing some biological insights on how complex deep learning models are performing cancer classification and making predictions across multiple cancer tissues using gene expressions.

6.2 Contributions

Our work has contributed to developing cancer classifiers with the capabilities of detecting more complex types of genetic alterations driving cancer progression, by learning the genomic signatures shared across multiple cancer tissue types. This was achieved by introducing a Deep Learning framework for early cancer diagnosis and designing a comprehensive *Multi-Tissue cancer classifier* based on molecular signatures of whole-transcriptome wide gene expressions. Our cancer classifier is based gene expressions collected from human samples representing multiple cancer tissue types and covering multiple organ sites.

We have contributed to eliminating the dependency on the prerequisite process of gene feature selection which is performed by current state-of-the-art cancer classification methods for discovering a predefined subset of informative genes to be used in the learning process. This was achieved by designing our Deep Learning framework as an end-to-end learning system for early cancer diagnosis which combines the process of gene feature selection and classification into one integrated learning system.

We have contributed in developing cancer classifiers with the capabilities of taking full advantage of genome-wide Next Generation Sequencing technologies to discover the

correlated patterns of genes across the full set of DNAs in the human genome and across multiple cancer tissue types. This was achieved by designing a new *Deep Neural Network* architecture called Gene eXpression Network (GeneXNet), which is specifically designed to address the complex nature of whole-transcriptome gene expressions. We demonstrated how our model architecture can learn the sequence of DNA and RNA in cancer cells and identify genetic changes that alter cell behavior and cause uncontrollable growth and malignancy. We also demonstrated how our new model architecture has the capabilities for learning the genomic signatures across multiple tissue types without requiring the prerequisite of gene feature selection.

We have contributed to eliminating the dependency on huge amounts of patient data and helped in solving one of the biggest challenges in cancer classification which is lack of patient samples. This was achieved by designing a *Deep Transfer Learning* model that effectively functions as a *generic* Multi-Tissue cancer classifier by learning genomic signatures collected from multiple cancer tissue types. We demonstrated how our model can be used for *Transfer Learning* to build classifiers for tumor types that are lacking sufficient patient samples to be trained independently.

We have contributed to eliminating the manual process of handcrafting the design of deep network architectures and contributed to eliminating the manual process of hyperparameter optimization and fine-tuning on the target dataset. This was achieved by designing an end-to-end *Deep Reinforcement Learning* framework that automatically learns the optimal Deep Neural Network architecture together with the associated optimal hyperparameters that maximizes the performance of our multi-tissue cancer classifier.

We have contributed in providing medical professionals with more confidence in using deep learning for medical diagnosis by providing some biological interpretation on how complex deep learning models are performing cancer classification and making predictions on cancer tumors. This was achieved by designing visualization procedures to provide more biological insight on how the proposed network model is learning genomic signatures of whole-transcriptome gene expressions and accurately performing classification across multiple cancer tumors. We have demonstrated how our network design provides the capability to visualize gene localization maps highlighting the important regions in the gene expressions influencing the tumor class prediction. We have also demonstrated how our network design provides the capability to visualize the molecular clusters formed by intermediate gene expression feature maps learned by the network which helps in revealing the genomic relationships of gene expressions that are influential in the tumor progression.

6.3 Analysis

Our classification experiments have demonstrated how the design of our proposed Gene eXpression Network (GeneXNet) can be used as a general end-to-end learning system for classification across multiple cancer tissue types without performing the prerequisite process of gene feature selection. We demonstrated how our model can specifically target the complex nature of the whole-transcriptome gene expression data and addresses the lack of training samples, without suffering from severe overfitting in comparison to using the current state-of-the-art deep CNN models which have been designed specifically for computer vision tasks. Our model has allowed training deeper network architectures with complex data like whole-transcriptome gene expressions, despite the large number of genes. The experiments demonstrated that our model design which combines both dense and residual learning layers, performs a regularizing effect which helps avoid overfitting and degradation in performance as the network depth increases. This is achieved by means of re-using the gene expression feature maps learned by different layers, which increases the variation of input signals fed to subsequent layers since it represents the collective knowledge of the network. The connectivity of the dense layers provide each layer with more direct access to the gradients from the loss function and the original input signal, while the residual layers with identity mappings provide a direct path for information propagation in the forward and backward passes.

Our Transfer Learning experiments have demonstrated that the comprehensive genomic signatures learned by training our model using all the data allowed us to perform efficient transfer learning by using the pre-trained model as a generic feature extractor to build additional classifiers for any of the individual tumor sites, especially for the organ sites which were lacking sufficient patient samples to be trained independently. These results have demonstrated how transfer learning was able to solve one of the biggest challenges in cancer classification which is lack of patient samples. The experiment demonstrated that by reusing the weights of the pretrained GeneXNet model, we were able to use the same network for feature extraction on a different cancer tumor type. The experiments have also demonstrated that the discriminative molecular features for one cancer classifier were also relevant for other cancer types. The results demonstrated that our pretrained model was able to learn the complex types of genomic signatures collected from multiple cancer tissue types and that it was able to effectively function as a generic model for cancer classification.

6.4 Biological Significance

Our work in cancer classification helps in directly solving one of the major challenges in cancer treatment, since patients are diagnosed at very late stages when appropriate medical interventions become less effective and full curative treatment is no longer achievable. To our knowledge, this is the first effort to develop a Multi-Tissue cancer classifier based on a full set of whole-transcriptome wide gene expressions collected from tumors across different tissue types without requiring a prerequisite process of gene feature selection. We have contributed in providing medical professionals with more confidence in using deep learning for medical diagnosis by providing some biological insights on how complex deep learning models are performing cancer classification and making predictions across cancer tumors.

We introduced a visualization method which uses the gradient information flowing in our proposed Gene eXpression Network (GeneXNet) model to produce gene localization maps highlighting the important regions in the gene expressions which influenced the resulting tumor class prediction. The gene expression data is sparse and very high in dimensionality since it represents a snapshot of the whole transcriptome rather than a predetermined subset of genes. By identifying class-discriminative localization map in the gene expressions, we were able to identify the subset of genes driving cancer progression and resulted in the model's tumor class prediction. Our experiments have demonstrated the strength of our method as our GeneXNet model was able to automatically identify a small subset of class-discriminative genes out of the total 60,483 genes originally included in each individual sample of our cancer tumor dataset. The network automatically identified the TP53 gene as one of the top features common across all tumor types which implicitly validates our procedure since the TP53 is considered the most commonly mutated gene in all cancers. Our experiments also demonstrated that some of the identified discriminative genes were also common in other samples across different tumor types even though the tissues belonged to different organ sites. This subset includes: TP53, TTN, MUC16, LRP1B, CSMD3, PIK3CA, MUC4, RYR2, USH2A, FLG, PTPRD, CSMD1. These discriminative genes identified by the network have great biological significance for early cancer diagnosis. For example, the mutations of PIK3CA gene are one of the most common in Breast cancer and are reported in over one third of cases [112]. Mutations in TTN gene are associated with one of the most common inherited cardiac disorders known as Hypertrophic Cardiomyopathy (HCM) [111]. MUC16 has a biological role in the progression of Ovarian tumors and there has been substantial work to develop therapeutic approaches to eradicate Ovarian tumors by targeting MUC16 [113]. LRP1B is frequently mutated in

Melanoma, Non-small Cell Lung cancer (NSCLC) and other types of tumors. LRP1B is also a potential contributor to the emergence of chemotherapy resistance while treating cancer patients [114]. CSMD3 was identified as the second most frequently mutated gene in Lung cancer after TP53 [4]. MUC4 is a membrane bound mucin gene responsible for progression of several cancers due to its anti-adhesive properties including Bile Duct, Breast, Colon, Esophagus, Ovary, Lung, Prostate, Stomach and Pancreas [111]. Mutations of RYR2 gene are a common cause of abnormal heart failures such as Catecholaminergic Polymorphic Ventricular Tachycardia (CPVT) [111]. PTPRD is frequently mutated in various types of cancer, including Glioblastoma, Melanoma, Breast and Colon [4].

We introduced a visualization procedure for observing the evolution of molecular clusters formed by intermediate gene expression feature maps learned by our GeneXNet model. The genetic signatures learned by the feature maps in the deep layers make the network capable of representing complex genetic alterations shared by tumors across different tissue types. Visualizing the molecular clusters of gene expressions provides more insight on how the network is learning small meaningful relationships between the genes which in turn describe the characteristic influencing the Cancer tumor. Our experiments have demonstrated how this visualization provides the opportunity to study the genomic relationships of gene expressions across multiple cancer tissue types. We observed from our experiments that the number of cluster groups learned by the network decreases as we move towards the deep layers. We observed that the final network block has less number of clusters where more clusters have merged together. These results have great significance since they demonstrate that as we go deeper in the network, the gene feature maps become more abstract in the sense that they are less representative of the individual tumor samples and more representative of the tumor classes. We also observed from our experiments that although tissue site of origin was mostly a dominant factor for cluster formation, but some clusters also included tumor types across multiple different organs. We observed that clusters were formed for tumor types which appeared to have similar organs or tissue characteristics. For example, Bile Duct and Liver tumors clustered together. Brain and Nervous system tumors also clustered together. Kidney and Adrenal Gland tumors formed multiple clusters, Lymph Nodes and Bone Marrow tumors clustered together. Many small overlapping clusters formed together for Stomach, Colorectal, Esophagus and Pancreas tumors. Visualizing the evolution of molecular clusters formed by intermediate gene feature maps, has demonstrated how our proposed GeneXNet is functioning as a comprehensive multi-tumor Cancer classifier.

6.5 Future work

We believe there is great potential for further research to expand on our work for cancer diagnosis. Our work focused on designing a multi-tissue cancer classifier based on Total RNA Sequencing using gene expressions from coding mRNA. Future work can explore learning more complex genomic signatures by including Omics data using other multiple forms of NGS platforms and experimental strategies such as DNA hypermethylation, aneuploidy, non-coding microRNA, DNA Copy Number Variants (CNV) and Reverse Phase Protein Arrays (RPPA). This will provide the opportunity to create a more comprehensive repository of pretrained models readily available for cancer classification using transfer learning.

One of the common approaches in classification is to use an Ensemble of multiple classifiers (mixture of experts) to improve the classification accuracy. Future work can target cancer diagnosis and improving classifier performance by designing *Ensemble Models* which could integrate multiple genome-wide platforms by learning molecular signatures across multiple forms of Omics data. Future work can also target using different Gene eXpression Network models in addition to other network architectures and combine their classification decisions.

Future work can further expand on our visualization methods to provide more in-depth biological insights to medical professionals and provide them with more confidence in using deep learning for medical diagnosis. Our experiments have demonstrated how our proposed network was able to automatically identify discriminative genes that were common across the cancer tumor types even though the tissues belonged to different organ sites. We have provided some biological significance for these identified genes for early cancer diagnosis. Future work needs to expand further on these results and provide more in-depth biological interpretation on the discriminative genes and their influence on early cancer diagnosis. Our experiments have demonstrated how our proposed network is functioning as a comprehensive multi-tumor cancer classifier by visualizing the evolution of molecular clusters formed by intermediate gene feature maps. Future work could target to perform a more in-depth analysis and biological evaluation of the clusters formed for different tumor types. This requires the research collaboration with medical experts as it requires more in-depth knowledge of the medical characteristics of the underlying human organs and their tissue characteristics.

6.6 COVID-19

COVID-19 has shown a dramatic and devastating impact across the world. Predicting virus diseases such as COVID-19 is extremely challenging, but there is great potential for the application of deep machine learning for early detection and diagnosis. Although our work was focused on cancer classification, but we believe that our proposed methods are applicable to other diseases and Omics data in particular for COVID-19. Next generation sequencing provides a great opportunity to investigate the mechanisms that underpin COVID-19 infections and transmission. Future work should target the use of deep machine learning and Omics data in the development of novel screening methods, drug molecules, vaccines, and potential antibiotics. Future work should also explore the use of genomics and transcriptomics data to predict the effects of new vaccines and drugs on patients.

REFERENCES

- [1] International Agency for Research on Cancer and World Health Organization, “World Cancer Report 2020,” 2020.
- [2] F. Bray, J. Ferlay, I. Soerjomataram, R. L. Siegel, L. A. Torre, and A. Jemal, “Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries,” *CA: A Cancer Journal for Clinicians*, vol. 68, no. 6, pp. 394–424, 2018, doi: 10.3322/caac.21492.
- [3] World Health Organization, “Cancer Prevention.” <https://www.who.int/health-topics/cancer>.
- [4] US National Cancer Institute (NCI), “Cancer Research.” <https://www.cancer.gov/research>.
- [5] K. A. Hoadley *et al.*, “Cell-of-Origin Patterns Dominate the Molecular Classification of 10,000 Tumors from 33 Types of Cancer,” *Cell*, vol. 173, no. 2, pp. 291–304.e6, Apr. 2018, doi: 10.1016/j.cell.2018.03.022.
- [6] T. M. Malta *et al.*, “Machine Learning Identifies Stemness Features Associated with Oncogenic Dedifferentiation,” *Cell*, vol. 173, no. 2, pp. 338–354.e15, Apr. 2018, doi: 10.1016/j.cell.2018.03.034.
- [7] K. A. Hoadley *et al.*, “Multiplatform Analysis of 12 Cancer Types Reveals Molecular Classification within and across Tissues of Origin,” *Cell*, vol. 158, no. 4, pp. 929–944, Aug. 2014, doi: 10.1016/j.cell.2014.06.049.
- [8] P. Wu and D. Wang, “Classification of a DNA Microarray for Diagnosing Cancer Using a Complex Network Based Method,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 16, no. 3, pp. 801–808, May 2019, doi: 10.1109/TCBB.2018.2868341.
- [9] C. Peng, X. Wu, W. Yuan, X. Zhang, and Y. Li, “MGRFE: multilayer recursive feature elimination based on an embedded genetic algorithm for cancer classification,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pp. 1–1, 2019, doi: 10.1109/TCBB.2019.2921961.
- [10] F. Hu, Y. Zhou, Q. Wang, Z. Yang, Y. Shi, and Q. Chi, “Gene expression classification of lung adenocarcinoma into molecular subtypes,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pp. 1–1, 2019, doi: 10.1109/TCBB.2019.2905553.
- [11] H. Lu, H. Gao, M. Ye, and X. Wang, “A Hybrid Ensemble Algorithm Combining AdaBoost and Genetic Algorithm for Cancer Classification with Gene Expression Data,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pp. 1–1, 2019, doi: 10.1109/TCBB.2019.2952102.
- [12] J. Xu, P. Wu, Y. Chen, Q. Meng, H. Dawood, and M. M. Khan, “A Novel Deep Flexible Neural Forest Model for Classification of Cancer Subtypes Based on Gene Expression Data,” *IEEE Access*, vol. 7, pp. 22086–22095, 2019, doi: 10.1109/ACCESS.2019.2898723.
- [13] C.-Q. Xia, K. Han, Y. Qi, Y. Zhang, and D.-J. Yu, “A Self-Training Subspace Clustering Algorithm under Low-Rank Representation for Cancer Classification on Gene Expression Data,” *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 15, no. 4, pp. 1315–1324, Aug. 2018, doi: 10.1109/TCBB.2017.2712607.
- [14] E. Razak, F. Yusof, and R. A. Raus, “Classification of miRNA Expression Data Using Random Forests for Cancer Diagnosis,” in *2016 International Conference on Computer and Communication Engineering (ICCCE)*, Jul. 2016, pp. 187–190, doi: 10.1109/ICCCE.2016.49.
- [15] S. H. Bouazza, N. Hamdi, A. Zeroual, and K. Auhmani, “Gene-expression-based cancer classification through feature selection with KNN and SVM classifiers,” in *2015 Intelligent Systems and Computer Vision (ISCV)*, Mar. 2015, pp. 1–6, doi: 10.1109/ISACV.2015.7106168.
- [16] S. A. Ludwig, D. Jakobovic, and S. Picsek, “Analyzing gene expression data: Fuzzy decision tree algorithm applied to the classification of cancer data,” in *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Aug. 2015, pp. 1–8, doi: 10.1109/FUZZ-IEEE.2015.7337854.
- [17] S.-Y. Hsieh and Y.-C. Chou, “A Faster cDNA Microarray Gene Expression Data Classifier for Diagnosing Diseases,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 13, no. 1, pp. 43–54, Jan. 2016, doi: 10.1109/TCBB.2015.2474389.

- [18] J.-X. Liu, Y. Xu, C.-H. Zheng, H. Kong, and Z.-H. Lai, "RPCA-Based Tumor Classification Using Gene Expression Data," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 12, no. 4, pp. 964–970, Jul. 2015, doi: 10.1109/TCBB.2014.2383375.
- [19] "Classification Analysis of DNA Microarrays | Wiley," *Wiley.com*. .
- [20] K. Liu, J. Ye, Y. Yang, L. Shen, and H. Jiang, "A Unified Model for Joint Normalization and Differential Gene Expression Detection in RNA-Seq Data," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 16, no. 2, pp. 442–454, Mar. 2019, doi: 10.1109/TCBB.2018.2790918.
- [21] J. M. Knight, I. Ivanov, K. Triff, R. S. Chapkin, and E. R. Dougherty, "Detecting Multivariate Gene Interactions in RNA-Seq Data Using Optimal Bayesian Classification," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 2, pp. 484–493, Mar. 2018, doi: 10.1109/TCBB.2015.2485223.
- [22] K. R. Kukurba and S. B. Montgomery, "RNA Sequencing and Analysis," *Cold Spring Harb Protoc*, vol. 2015, no. 11, p. pdb.top084970, Nov. 2015, doi: 10.1101/pdb.top084970.
- [23] J. E. Dancey, P. L. Bedard, N. Onetto, and T. J. Hudson, "The Genetic Basis for Cancer Treatment Decisions," *Cell*, vol. 148, no. 3, pp. 409–420, Feb. 2012, doi: 10.1016/j.cell.2012.01.014.
- [24] S. Sleijfer, J. Bogaerts, and L. L. Siu, "Designing Transformative Clinical Trials in the Cancer Genome Era," *JCO*, vol. 31, no. 15, pp. 1834–1841, Apr. 2013, doi: 10.1200/JCO.2012.45.3639.
- [25] L. E. MacConaill, "Existing and Emerging Technologies for Tumor Genomic Profiling," *JCO*, vol. 31, no. 15, pp. 1815–1824, Apr. 2013, doi: 10.1200/JCO.2012.46.5948.
- [26] J. M. Rizzo and M. J. Buck, "Key Principles and Clinical Applications of 'Next-Generation' DNA Sequencing," *Cancer Prev Res*, vol. 5, no. 7, pp. 887–900, Jul. 2012, doi: 10.1158/1940-6207.CAPR-11-0432.
- [27] Z. Wang, M. Gerstein, and M. Snyder, "RNA-Seq: a revolutionary tool for transcriptomics," *Nature Reviews Genetics*, vol. 10, no. 1, pp. 57–63, Jan. 2009, doi: 10.1038/nrg2484.
- [28] C. Liu and H. S. Wong, "Structured Penalized Logistic Regression for Gene Selection in Gene Expression Data Analysis," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 16, no. 1, pp. 312–321, Jan. 2019, doi: 10.1109/TCBB.2017.2767589.
- [29] X. Lin *et al.*, "The Robust Classification Model Based on Combinatorial Features," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 16, no. 2, pp. 650–657, Mar. 2019, doi: 10.1109/TCBB.2017.2779512.
- [30] K. R. Kavitha, A. V. Ram, S. Anandu, S. Karthik, S. Kailas, and N. M. Arjun, "PCA-based gene selection for cancer classification," in *2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, Dec. 2018, pp. 1–4, doi: 10.1109/ICCIC.2018.8782337.
- [31] S. An, J. Wang, and J. Wei, "Local-Nearest-Neighbors-Based Feature Weighting for Gene Selection," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 5, pp. 1538–1548, Sep. 2018, doi: 10.1109/TCBB.2017.2712775.
- [32] D. Pavithra and B. Lakshmanan, "Feature selection and classification in gene expression cancer data," in *2017 International Conference on Computational Intelligence in Data Science (ICCIDS)*, Jun. 2017, pp. 1–6, doi: 10.1109/ICCIDS.2017.8272668.
- [33] J. C. Ang, A. Mirzal, H. Haron, and H. N. A. Hamed, "Supervised, Unsupervised, and Semi-Supervised Feature Selection: A Review on Gene Selection," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 13, no. 5, pp. 971–989, Sep. 2016, doi: 10.1109/TCBB.2015.2478454.
- [34] J. Tang and S. Zhou, "A New Approach for Feature Selection from Microarray Data Based on Mutual Information," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 13, no. 6, pp. 1004–1015, Nov. 2016, doi: 10.1109/TCBB.2016.2515582.
- [35] J.-X. Liu, Y. Xu, Y.-L. Gao, C.-H. Zheng, D. Wang, and Q. Zhu, "A Class-Information-Based Sparse Component Analysis Method to Identify Differentially Expressed Genes on RNA-Seq Data," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 13, no. 2, pp. 392–398, Mar. 2016, doi: 10.1109/TCBB.2015.2440265.

- [36] W. H. Chan *et al.*, “Identification of informative genes and pathways using an improved penalized support vector machine with a weighting scheme,” *Computers in Biology and Medicine*, vol. 77, pp. 102–115, Oct. 2016, doi: 10.1016/j.compbio.2016.08.004.
- [37] C. Lazar *et al.*, “A Survey on Filter Techniques for Feature Selection in Gene Expression Microarray Analysis,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 9, no. 4, pp. 1106–1119, Jul. 2012, doi: 10.1109/TCBB.2012.33.
- [38] “The Cancer Genome Atlas (TCGA) Research Network.” <https://www.cancer.gov/tcga>.
- [39] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.
- [40] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, “Learning Transferable Architectures for Scalable Image Recognition,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 8697–8710, doi: 10.1109/CVPR.2018.00907.
- [41] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely Connected Convolutional Networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 2261–2269, doi: 10.1109/CVPR.2017.243.
- [42] J. Hu, L. Shen, and G. Sun, “Squeeze-and-Excitation Networks,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 7132–7141, doi: 10.1109/CVPR.2018.00745.
- [43] A. G. Howard *et al.*, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” *CoRR*, vol. abs/1704.04861, 2017, Accessed: Nov. 17, 2019. [Online]. Available: <http://arxiv.org/abs/1704.04861>.
- [44] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated Residual Transformations for Deep Neural Networks,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 5987–5995, doi: 10.1109/CVPR.2017.634.
- [45] F. Chollet, “Xception: Deep Learning with Depthwise Separable Convolutions,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 1800–1807, doi: 10.1109/CVPR.2017.195.
- [46] K. He, X. Zhang, S. Ren, and J. Sun, “Identity Mappings in Deep Residual Networks,” in *Computer Vision – ECCV 2016*, Cham, 2016, pp. 630–645, doi: 10.1007/978-3-319-46493-0_38.
- [47] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, “Deep Networks with Stochastic Depth,” in *Computer Vision – ECCV 2016*, Cham, 2016, pp. 646–661, doi: 10.1007/978-3-319-46493-0_39.
- [48] O. Russakovsky *et al.*, “ImageNet Large Scale Visual Recognition Challenge,” *Int J Comput Vis*, vol. 115, no. 3, pp. 211–252, Dec. 2015, doi: 10.1007/s11263-015-0816-y.
- [49] V. Nair and G. E. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines,” in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, USA, 2010, pp. 807–814, Accessed: Nov. 17, 2019. [Online]. Available: <http://dl.acm.org/citation.cfm?id=3104322.3104425>.
- [50] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015, Accessed: Nov. 17, 2019. [Online]. Available: <http://arxiv.org/abs/1412.6980>.
- [51] J. Duchi, E. Hazan, and Y. Singer, “Adaptive Subgradient Methods for Online Learning and Stochastic Optimization,” *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.
- [52] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning Deep Features for Discriminative Localization,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 2921–2929, doi: 10.1109/CVPR.2016.319.
- [53] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization,” in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 618–626, doi: 10.1109/ICCV.2017.74.
- [54] M. Lin, Q. Chen, and S. Yan, “Network In Network,” in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014, Accessed: Nov. 17, 2019. [Online]. Available: <http://arxiv.org/abs/1312.4400>.

- [55] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, Lille, France, 2015, pp. 448–456, Accessed: Nov. 24, 2019. [Online]. Available: <http://dl.acm.org/citation.cfm?id=3045118.3045167>.
- [56] M. Abadi *et al.*, "TensorFlow: A System for Large-scale Machine Learning," in *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation*, Berkeley, CA, USA, 2016, pp. 265–283, Accessed: Nov. 17, 2019. [Online]. Available: <http://dl.acm.org/citation.cfm?id=3026877.3026899>.
- [57] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the Impact of Residual Connections on Learning," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, California, USA, 2017, pp. 4278–4284, Accessed: Nov. 17, 2019. [Online]. Available: <http://dl.acm.org/citation.cfm?id=3298023.3298188>.
- [58] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
- [59] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.
- [60] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.
- [61] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.
- [62] S. Ruder, "An overview of gradient descent optimization algorithms," *CoRR*, vol. abs/1609.04747, 2016, Accessed: Nov. 17, 2019. [Online]. Available: <http://arxiv.org/abs/1609.04747>.
- [63] C. Szegedy *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.
- [64] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *CoRR*, vol. abs/1409.1556, 2014, Accessed: Nov. 17, 2019. [Online]. Available: <http://arxiv.org/abs/1409.1556>.
- [65] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," in *Computer Vision – ECCV 2014*, Cham, 2014, pp. 818–833, doi: 10.1007/978-3-319-10590-1_53.
- [66] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, p. 2012.
- [67] Y. LeCun, "Learning Invariant Feature Hierarchies," in *Proceedings of the 12th International Conference on Computer Vision - Volume Part I*, Berlin, Heidelberg, 2012, pp. 496–505, doi: 10.1007/978-3-642-33863-2_51.
- [68] T. Fawcett, "An Introduction to ROC Analysis," *Pattern Recogn. Lett.*, vol. 27, no. 8, pp. 861–874, Jun. 2006, doi: 10.1016/j.patrec.2005.10.010.
- [69] M. D. Wilkerson and D. N. Hayes, "ConsensusClusterPlus: a class discovery tool with confidence assessments and item tracking," *Bioinformatics*, vol. 26, no. 12, pp. 1572–1573, Jun. 2010, doi: 10.1093/bioinformatics/btq170.
- [70] S. Monti, P. Tamayo, J. Mesirov, and T. Golub, "Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data," *Mach. Learn.*, vol. 52, no. 1–2, pp. 91–118, Jul. 2003, doi: 10.1023/A:1023949509487.
- [71] T. Haferlach *et al.*, "Clinical Utility of Microarray-Based Gene Expression Profiling in the Diagnosis and Subclassification of Leukemia: Report From the International Microarray Innovations in Leukemia Study Group," *Journal of Clinical Oncology*, vol. 28, no. 15, pp. 2529–2537, Apr. 2010, doi: 10.1200/JCO.2009.23.4732.
- [72] L. J. van 't Veer *et al.*, "Gene expression profiling predicts clinical outcome of breast cancer," *Nature*, vol. 415, no. 6871, pp. 530–536, Jan. 2002, doi: 10.1038/415530a.

- [73] T. R. Golub *et al.*, “Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring,” *Science*, vol. 286, no. 5439, pp. 531–537, Oct. 1999, doi: 10.1126/science.286.5439.531.
- [74] T. Sørlie *et al.*, “Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications,” *PNAS*, vol. 98, no. 19, pp. 10869–10874, Sep. 2001, doi: 10.1073/pnas.191367098.
- [75] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.
- [76] D. R. Wilson and T. R. Martinez, “The general inefficiency of batch training for gradient descent learning,” *Neural Netw.*, vol. 16, no. 10, pp. 1429–1451, Dec. 2003, doi: 10.1016/S0893-6080(03)00138-2.
- [77] N. Qian, “On the momentum term in gradient descent learning algorithms,” *Neural Networks*, vol. 12, no. 1, pp. 145–151, Jan. 1999, doi: 10.1016/S0893-6080(98)00116-6.
- [78] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, “On the importance of initialization and momentum in deep learning,” in *International Conference on Machine Learning*, Feb. 2013, pp. 1139–1147, Accessed: Mar. 04, 2020. [Online]. Available: <http://proceedings.mlr.press/v28/sutskever13.html>.
- [79] Y. Nesterov, *Lectures on Convex Optimization*, 2nd ed. Springer International Publishing, 2018.
- [80] J. Dean *et al.*, “Large Scale Distributed Deep Networks,” in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1223–1231.
- [81] B. Zoph and Q. V. Le, “Neural Architecture Search with Reinforcement Learning,” in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017, [Online]. Available: <https://openreview.net/forum?id=r1Ue8Hcxg>.
- [82] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding deep learning requires rethinking generalization,” in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017, [Online]. Available: <https://openreview.net/forum?id=Sy8gdb9xx>.
- [83] B. Baker, O. Gupta, N. Naik, and R. Raskar, “Designing Neural Network Architectures using Reinforcement Learning,” in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017, [Online]. Available: <https://openreview.net/forum?id=S1c2cvqee>.
- [84] M. Andrychowicz *et al.*, “Learning to learn by gradient descent by gradient descent,” in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 3981–3989.
- [85] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *J. Mach. Learn. Res.*, vol. 13, no. null, pp. 281–305, Feb. 2012.
- [86] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, “Algorithms for hyper-parameter optimization,” in *Proceedings of the 24th International Conference on Neural Information Processing Systems*, Granada, Spain, Dec. 2011, pp. 2546–2554, Accessed: Mar. 08, 2020. [Online].
- [87] S. Saxena and J. Verbeek, “Convolutional Neural Fabrics,” in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 4053–4061.
- [88] J. Snoek *et al.*, “Scalable Bayesian optimization using deep neural networks,” in *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, Lille, France, Jul. 2015, pp. 2171–2180, Accessed: Mar. 08, 2020. [Online].
- [89] A. Nagabandi, G. Kahn, R. S. Fearing, and S. Levine, “Neural Network Dynamics for Model-Based Deep Reinforcement Learning with Model-Free Fine-Tuning,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 7559–7566, doi: 10.1109/ICRA.2018.8463189.
- [90] S. Gu, E. Holly, T. Lillicrap, and S. Levine, “Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates,” in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 3389–3396, doi: 10.1109/ICRA.2017.7989385.

- [91] G. Kahn, T. Zhang, S. Levine, and P. Abbeel, "PLATO: Policy learning using adaptive trajectory optimization," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 3342–3349, doi: 10.1109/ICRA.2017.7989379.
- [92] S. Gu, T. Lillicrap, Z. Ghahramani, R. E. Turner, B. Schölkopf, and S. Levine, "Interpolated policy gradient: merging on-policy and off-policy gradient estimation for deep reinforcement learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, Long Beach, California, USA, Dec. 2017, pp. 3849–3858, Accessed: Mar. 08, 2020. [Online].
- [93] M. Zhang, Z. McCarthy, C. Finn, S. Levine, and P. Abbeel, "Learning deep neural network policies with continuous memory states," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 520–527, doi: 10.1109/ICRA.2016.7487174.
- [94] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust Region Policy Optimization," in *International Conference on Machine Learning*, Jun. 2015, pp. 1889–1897, Accessed: Mar. 08, 2020. [Online]. Available: <http://proceedings.mlr.press/v37/schulman15.html>.
- [95] S. Levine and V. Koltun, "Guided Policy Search," in *International Conference on Machine Learning*, Feb. 2013, pp. 1–9, Accessed: Mar. 08, 2020. [Online]. Available: <http://proceedings.mlr.press/v28/levine13.html>.
- [96] J. Peters and S. Schaal, "Reinforcement learning of motor skills with policy gradients," *Neural Networks*, vol. 21, no. 4, pp. 682–697, May 2008, doi: 10.1016/j.neunet.2008.02.003.
- [97] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.
- [98] J. Baxter and P. L. Bartlett, "Infinite-Horizon Policy-Gradient Estimation," *Journal of Artificial Intelligence Research*, vol. 15, pp. 319–350, Nov. 2001, doi: 10.1613/jair.806.
- [99] R. J. Williams, "Simple statistical gradient-following algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, no. 3, pp. 229–256, May 1992, doi: 10.1007/BF00992696.
- [100] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, Jan. 2016, doi: 10.1038/nature16961.
- [101] D. Silver *et al.*, "Mastering the game of Go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, Oct. 2017, doi: 10.1038/nature24270.
- [102] R. Pascanu, Ç. Gülçehre, K. Cho, and Y. Bengio, "How to Construct Deep Recurrent Neural Networks," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.
- [103] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, Mar. 1994, doi: 10.1109/72.279181.
- [104] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.
- [105] Li Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 594–611, Apr. 2006, doi: 10.1109/TPAMI.2006.79.
- [106] B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum, "One shot learning of simple visual concepts," *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 33, no. 33, 2011, Accessed: Mar. 04, 2020. [Online]. Available: <https://escholarship.org/uc/item/4ht821jx>.
- [107] R. Socher, M. Ganjoo, C. D. Manning, and A. Y. Ng, "Zero-shot learning through cross-modal transfer," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 1*, Lake Tahoe, Nevada, Dec. 2013, pp. 935–943, Accessed: Mar. 04, 2020. [Online].
- [108] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, Oct. 2018, doi: 10.1016/j.neunet.2018.07.011.
- [109] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, Jun. 2002, doi: 10.1613/jair.953.

- [110] Haibo He, Yang Bai, E. A. Garcia, and Shutao Li, “ADASYN: Adaptive synthetic sampling approach for imbalanced learning,” in *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, Jun. 2008, pp. 1322–1328, doi: 10.1109/IJCNN.2008.4633969.
- [111] US National Library of Medicine, Bethesda (MD), “Gene: National Center for Biotechnology Information,” 2004. <https://www.ncbi.nlm.nih.gov/gene/>.
- [112] D. Zardavas, W. A. Phillips, and S. Loi, “PIK3CA mutations in breast cancer: reconciling findings from preclinical and clinical data,” *Breast Cancer Research*, vol. 16, no. 1, p. 201, Jan. 2014, doi: 10.1186/bcr3605.
- [113] M. Felder *et al.*, “MUC16 (CA125): tumor biomarker to cancer therapy, a work in progress,” *Molecular Cancer*, vol. 13, no. 1, p. 129, May 2014, doi: 10.1186/1476-4598-13-129.
- [114] P. A. Cowin *et al.*, “LRP1B Deletion in High-Grade Serous Ovarian Cancers Is Associated with Acquired Chemotherapy Resistance to Liposomal Doxorubicin,” *Cancer Res*, vol. 72, no. 16, pp. 4060–4073, Aug. 2012, doi: 10.1158/0008-5472.CAN-12-0203.