T.R.

GEBZE TECHNICAL UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

IMAGE CREATION FROM NUMERICAL DATA

ÖMER FARUK AKDUMAN

SUPERVISOR
DR. BURCU YILMAZ

GEBZE
2023

T.R.
GEBZE TECHNICAL UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

IMAGE CREATION FROM NUMERICAL DATA

ÖMER FARUK AKDUMAN

SUPERVISOR
DR. BURCU YILMAZ

2023
GEBZE

GEBZE TECHNICAL UNIVERSITY	GRADUATION PROJECT JURY APPROVAL FORM

This study has been accepted as an Undergraduate Graduation Project in the Department of Computer Engineering on 21/01/2023 by the following jury.

JURY

Member
(Supervisor) : Dr. Burcu YILMAZ

Member : Prof. Habil Kalkan

ABSTRACT

Today, with the increase in the processing power of computers, CNN architectures and deep learning algorithms are used efficiently in many areas such as health, economy and security. CNN architectures and deep learning algorithms are used for tasks such as image recognition and processing of pixel data for given pixel groups. Although CNN architectures are currently used effectively in many areas, they cannot be used in tabular data because tabular data consists of numerical data on horizontal and vertical axis instead of pixels like pictures.

In this project, I will first use the RNA-Seq dataset I have chosen, models such as XGBoost and LightGBM, which are very suitable for the classification of tabular data, and then convert the tabular data in the RNA-Seq dataset into images using the DeepInsight and IGTD algorithms published in Nature Journal in 2019 and 2020, respectively. I will perform my classification with the help of CNN algorithms. At the end of the project, I will have successfully used CNN architectures on tabular data and finally I will compare machine learning and CNN architectures using F-score and confusion matrix.

ÖZET

Günümüzde bilgisayarların işlem gücünün artmasıyla birlikte CNN mimarileri ve derin öğrenme algoritmaları sağlık, ekonomi, güvenlik gibi birçok alanda verimli bir şekilde kullanılmaktadır. CNN mimarileri ile derin öğrenme algoritmaları verilen piksel grupları için görüntü tanıma ve piksel verilerinin işlenmesi gibi görevler için kullanılır. CNN mimarileri hâlihazırda pek çok alanda etkili bir biçimde kullanılmasına karsın tablosal verilerde kullanılamamaktadır çünkü tablosal veriler resimler gibi piksellerden oluşmak yerine yatay ve dikey eksende sayısal verilerden oluşmaktadır.

Bu projede ise öncelikle seçmiş olduğum RNA-Seq veri setini XGBoost, LightGBM gibi tablosal verilerin sınıflandırmasında oldukça uygun olan modelleri kullanacağım sonrasında ise sırasıyla 2019 ve 2020 yıllarında Nature Dergisin'de yayınlanmış olan DeepInsight ve IGTD algoritmalarını kullanarak RNA-Seq veri kümesindeki tablosal verileri görüntüye çevirip CNN algoritmaları yardımıyla sınıflandırmamı gerçekleştireceğim. Proje sonunda ise CNN mimarilerini tablosal veriler üzerinde başarıyla kullanmış olacağım ve son olarak f1 puanı ve hata matrisini kullanarak makine ögrenmeleri ile CNN mimarilerini karşılaştıracağım.

ACKNOWLEDGEMENT

I would like to express my gratitude to my supervisor Dr. Burcu YILMAZ, who always guided us in the completion of the project in the best way, and to my teacher Başak BULUZ, who did not spare her support throughout the graduation project.

I would also like to express my love and respect to my family, who are in almost everything and give their full support.

Ömer Faruk AKDUMAN

LIST OF SYMBOLS AND ABBREVIATIONS

Symbol or **Abbreviation**		**Explanation**
CNN	:	Convolutional neural network
DL	:	Deep learning
IGTD	:	Image Genaration from Tabular Data (algorithm's name)
RNA-seq	:	RNA sequencing
ML	:	Machine learning

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Creating images from numerical data is a process that involves converting data in a numerical format, such as a tabular data or a text file, into a visual representation, such as a symbol or an image. This process can be used in a variety of fields, including scientific research, engineering, and medical area.

There are several methods used to create images from numerical data. One of the most popular method is DeepInsight way that is a library that generate images from numerical data, which involves converting data into a grid of pixels. Another method (algorithm) is called IGTD, where each pixel represents a specific value.

In today's digital age, data is being generated at an unprecedented rate. From satellite imagery to medical scans, this data holds valuable information that can be used to gain insights and make informed decisions. One of the ways to extract meaning from this data is by visualizing it. However, creating images from numerical data can be a complex task, requiring specialized skills and software. In this article, we will explore the various methods used to create images from numerical data, including the advantages and disadvantages of each method. We will also discuss the latest advancements in the field, such as deep learning techniques, and their potential impact on image creation from numerical data.

1.1. Data Set

1.1.1. Data Set Selection Criterias

Creating images from numerical data also has its challenges. One of the biggest challenges is dealing with large datasets, which can be time-consuming and require powerful computing resources. Additionally, creating images that effectively convey the underlying data can be difficult, as it requires a balance between accuracy and readability.

To convert from tabular data to pictures I selected a voluminous dataset in 2 dimensions (vertically and horizontally) to see if the chosen algorithms could handle this situation: RNA-Seq

1.1.2. Data Set: RNA-seq

RNA-seq is a powerful method for studying the transcriptome, and it has been applied to many different types of cancer. In the context of cancer, RNA-seq is used to study the changes in gene expression that occur as a result of the cancer. By comparing the transcriptomes of cancer cells to normal cells, researchers can identify genes that are differentially expressed in cancer, which can provide insight int the biology of the disease and help identify new therapeutic targets.

RNA-seq has been used to study a wide range of cancer types, including:

1. Breast cancer: RNA-seq has been used to study the molecular subtypes of breast cancer and identify new therapeutic targets.

2. Lung cancer: RNA-seq has been used to study the molecular subtypes of lung cancer, including non-small cell lung cancer (NSCLC) and small cell lung cancer (SCLC), and identify new therapeutic targets.

3. Prostate cancer: RNA-seq has been used to study the molecular subtypes of prostate cancer and identify new therapeutic targets.

4. Colorectal cancer: RNA-seq has been used to study the molecular subtypes of colorectal cancer and identify new therapeutic targets.

5. Blood cancer: RNA-seq has been used to study the molecular subtypes of blood cancer, such as leukemia and lymphoma, and identify new therapeutic targets.

These are just a few examples of the many cancer types that have been studied using RNA-seq. Overall, RNA-seq is a powerful tool for studying the transcriptome in cancer and can provide valuable insights into the biology of the disease and help identify new therapeutic targets.

Table 1.1: Summary of RNA-seq Dataset.

Data set	Samples	Features	Classes
RNA-Seq	4550	19385	8

1.1.3. Data Set: RNA-seq Feature Selected

In RNA-seq dataset there are 19385 features, Since the time complexity required for the IGTD algorithm to render data into a picture is excessive due to the fact that the implementation of the algorithm is not multiprocess[1], I have reduced the number of features as much as possible with the feature selection methods. Thus, the IGTD algorithm turned the data into pictures easily and reasonably fast.

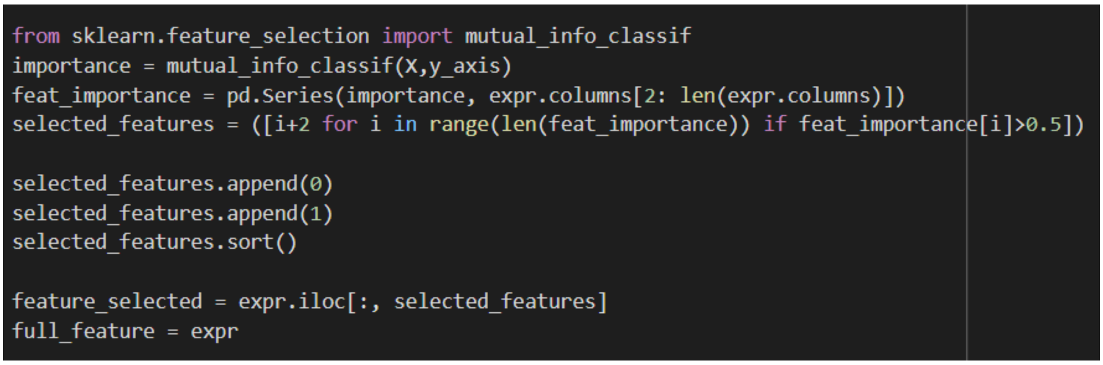
Note: DeepInsight 19385 can directly convert the featured data into a picture.

1.1.3.1. Feature Selection: Mutual information

I used the mutual information gain because of:

1- It considers the dependency between features and the target variable: Mutual information takes into account not only the individual feature's distribution but also the relationship between the feature and the target variable. This makes it a better measure of feature importance than methods that only consider the feature's distribution.

2- It's robust to feature scaling: Mutual information is not affected by feature scaling, which means that it does not require any preprocessing of the data.



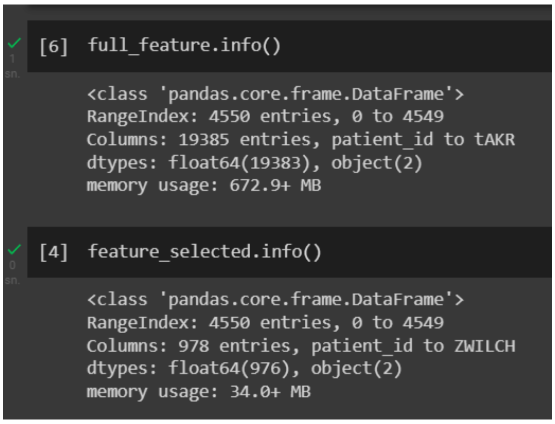
```
from sklearn.feature_selection import mutual_info_classif
importance = mutual_info_classif(X,y_axis)
feat_importance = pd.Series(importance, expr.columns[2: len(expr.columns)])
selected_features = ([i+2 for i in range(len(feat_importance)) if feat_importance[i]>0.5])

selected_features.append(0)
selected_features.append(1)
selected_features.sort()

feature_selected = expr.iloc[:, selected_features]
full_feature = expr
```

Figure 1.1: Implemantation of Mutual Information feature selection.

After feature selection new feature selected version RNA-seq's feature count is 978.



```
[6]  full_feature.info()

     <class 'pandas.core.frame.DataFrame'>
     RangeIndex: 4550 entries, 0 to 4549
     Columns: 19385 entries, patient_id to tAKR
     dtypes: float64(19383), object(2)
     memory usage: 672.9+ MB

[4]  feature_selected.info()

     <class 'pandas.core.frame.DataFrame'>
     RangeIndex: 4550 entries, 0 to 4549
     Columns: 978 entries, patient_id to ZWILCH
     dtypes: float64(976), object(2)
     memory usage: 34.0+ MB
```

Figure 1.2: General data set info before-after feature selection.

1.2. Machine Learning Algorithms

This categorical data I have chosen, RNA-seq data set, I will first classify cancer using classical machine learning algorithms. Criteria I paid attention to when choosing these machine learning algorithms:

1- Being suitable and ideal against categorical data

2- Being resistant to outliers

3- Modern and optimized

According to above criteria I decided to select tree-based ML models.

1.2.1. Random Forest

Random Forest is a widely used ML algorithm that is well-suited for working with categorical data.

A Random Forest model is an ensemble of decision trees, where each tree is trained on a different subset of the data. The final prediction is made by averaging the predictions of all the trees. This approach improves the overall accuracy of the model by reducing overfitting and variance.

So first model that I will use is Random Forest.

1.2.2. XGBoost

XGBoost (eXtreme Gradient Boosting) is a popular ML algorithm that is well-suited for working with tabular data, which is data that is organized into rows and columns. XGBoost is an extension of the gradient boosting algorithm and is often used in competitions and real-world problems

Another advantage of XGBoost is that it can handle large datasets with many features. It is also highly customizable and can be fine-tuned for a specific problem by adjusting the hyperparameters, such as the learning rate, depth of the trees, and the number of trees in the ensemble.

Overall, second ML model that will use in this project is XGBoost because it is a powerful algorithm that can handle tabular data well and can be used for both classification and regression problems. However, XGBoost can be computationally expensive and requires careful tuning of the hyperparameters to achieve optimal performance.

1.2.3. LightGBM

Main specialities of LightGBM that useful for this data set are:

1- Handling categorical features: LightGBM can handle categorical features by creating a histogram of the feature values, which allows it to process categorical data efficiently. It also supports categorical features with large cardinality, which is a common problem in many real-world applications.

2- Handling large datasets: LightGBM is designed to scale to large datasets and can be run on multiple machines in parallel.

1.3. Image Creation Algorithms

Today, there are very popular algorithms for generating pictures for training to CNN architectures from tabular data, the main ones being: Refined[2], IGTD[3] and DeepInsight[4]

I will use two of them: IGTD and DeepInsight for classify RNA-seq dataset.

1.3.1. IGTD

IGTD is an algorithm that turns data in a table format into images. It assigns each piece of data to a specific spot on the image, with similar data being close together and different data being farther apart. This creates an image for each piece of data, where the brightness of each spot on the image corresponds to the value of the data it represents. These generated images can then be used to create CNNs for further analysis.[5]

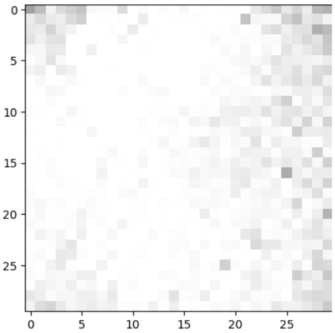


Figure 1.3: A BRCA cancer image representation that converted with IGTD algorithm.

1.3.2. DeepInsight

The concept of DeepInsight is to first transform a non-image sample to an image form and then supply it to the CNN architecture for the prediction or classification purpose. A simple illustration is given in Fig 1.3. a, where a feature vector x consisting of gene expression values is transformed to a feature matrix M by a transformation T. The location of features in the Cartesian coordinates depends on the similarity of features. For example, features g1, g3, g6 and are closer to each other in Fig 1.3. a. Once the locations of each feature are determined in a feature matrix, then the expression values or feature values are mapped. This will generate a unique image for each sample (or feature vector). N samples of d features will provide N samples of m×n feature matrices. This 2D matrix form will have all the d features. Thereafter, this set of N feature matrices are processed to the CNN architecture for learning the model and providing prediction. [6]

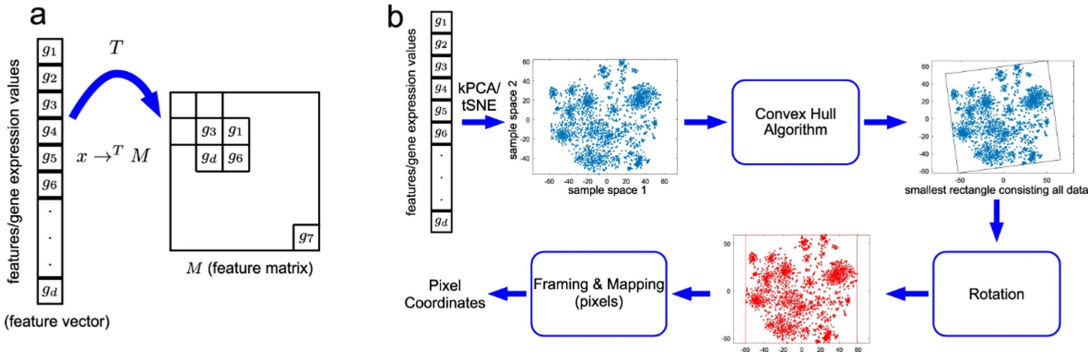


Figure 1.4: DeepInsight pipeline. (a) An illustration of transformation from feature vector to feature matrix. (b) An illustration of the DeepInsight methodology to transform a feature vector to image pixels.[7]

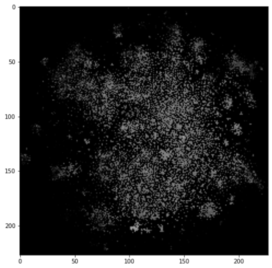


Figure 1.5: A BRCA cancer image representation that converted with DeepInsight algorithm.

1.4. CNN Architectures

1.4.1. ResNet

I will classify the images that converted with the ResNet CNN architecture, for this reasons: 1- Improved accuracy: ResNet's residual connections allow the gradients to flow more easily through the network during the training process, which enables the network to train much deeper architectures (hundreds of layers) without the problem of vanishing gradients. This can result in improved accuracy for a wide range of computer vision tasks 2- Computational efficiency: The use of bottleneck layers in ResNet can significantly reduce the number of feature maps in the network and improve computation efficiency, making it a good choice for resource-constrained devices such as mobile phones and embedded systems.

1.4.2. SquuezeNet

Moreover, if there is a classification accuracy difference between CNN architectures due to architecture structure, and to observe how the pictures behave in different architectures, I will also use SquuzeNet for RNA-seq dataset.

2. APPLICATION

I will apply RNA-seq, which is the data set I chose, to the machine learning models that I have determined, then I will produce images with the image generation methods IGTD and DeepInsight and apply these images in cancer classification with CNN architectures, in short, with classical machine learning architectures and CNN architectures. I will make a classification and evaluate the results separately.

2.1. Machine Learning Models

2.1.1. Random Forest

2.1.1.1. Random Forest: RNA-Seq Full Feature

Accuracy : 96.70
F1-Score: 0.96693562090522

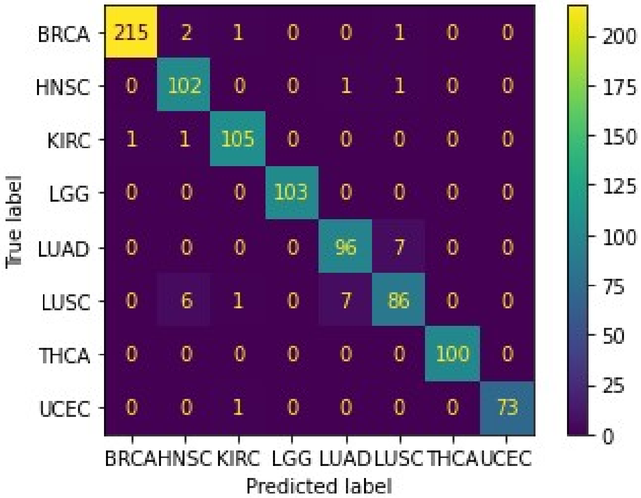


Figure 2.1: Random Forest Confusion Matrix RNA-seq full feature.

2.1.1.2. Random Forest: RNA-Seq Feature Selected

Accuracy : 97.36
F1-Score: 0.9735283569125691

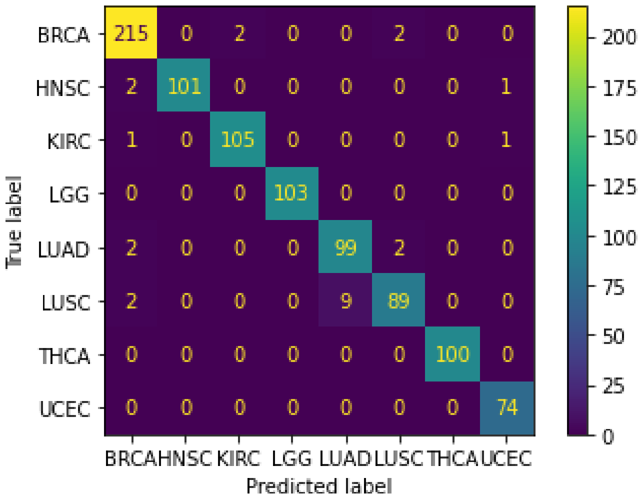


Figure 2.2: Random Forest Confusion Matrix RNA-seq feature selected.

2.1.2. XGBoost

2.1.2.1. XGBoost: RNA-Seq Full Feature

Accuracy : 98.68

F1-Score: 0.9868368856626165

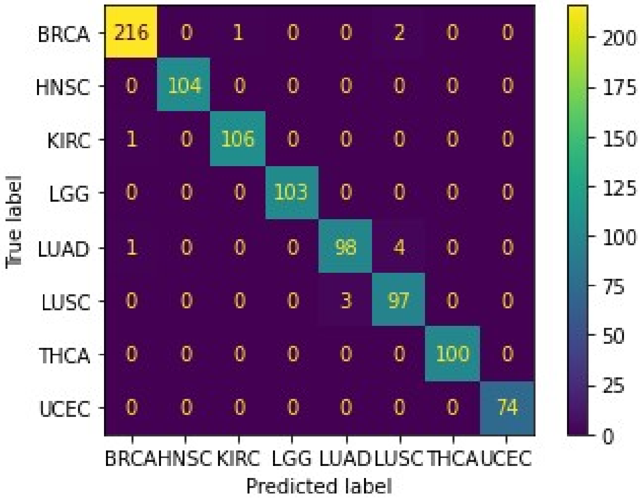


Figure 2.3: XGBoost Confusion Matrix RNA-seq full feature.

2.1.2.2. XGBoost: RNA-Seq Feature Selected

Accuracy : 98.13

F1-Score: 0.9813725875312491

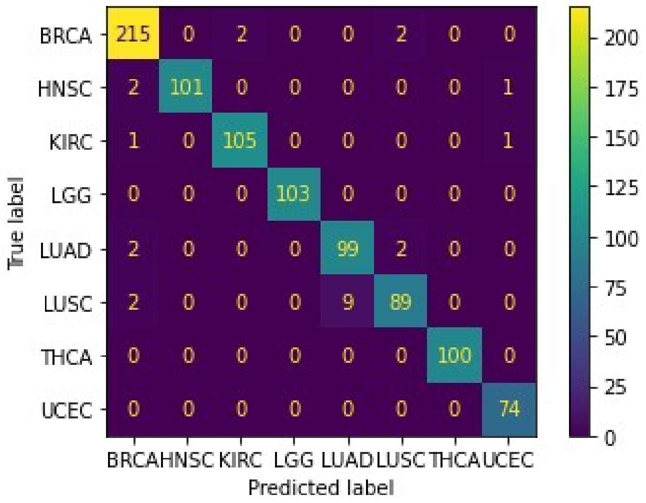


Figure 2.4: XGBoost Confusion Matrix RNA-seq feature selected.

2.1.3. LightGBM

2.1.3.1. LightGBM: RNA-Seq Full Feature

Accuracy : 98.35

F1-Score: 0.9834908999025253

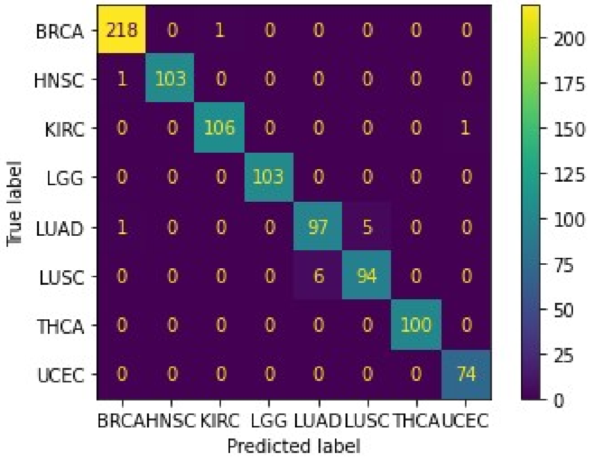


Figure 2.5: LightGBM Confusion Matrix full feature.

2.1.3.2. LightGBM: RNA-Seq Feature Selected

Accuracy : 98.68

F1-Score: 0.9867920920782038

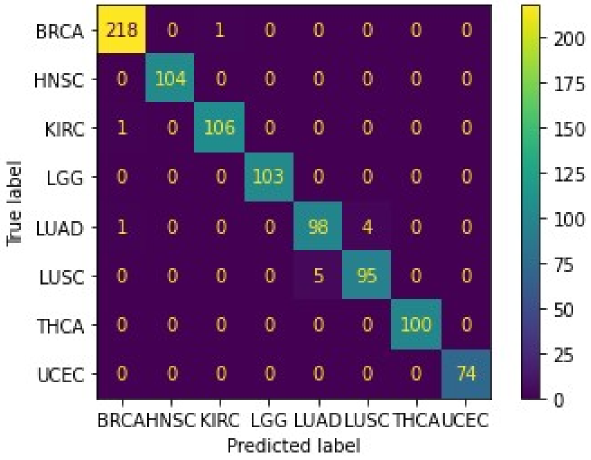


Figure 2.6: LightGBM Confusion Matrix RNA-seq feature selected.

2.2. CNNs with Image Generation Algorithms

2.2.1. DeepInsight

2.2.1.1. DeepInsight: SqueezeNet Full Feature

Accuracy : 97.03

F1-Score: 0.9703559723971841

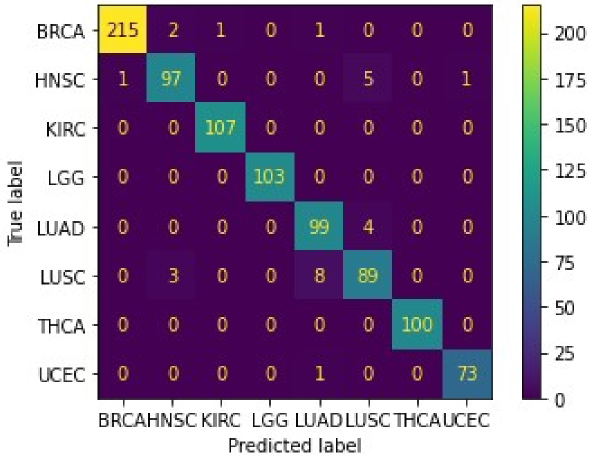


Figure 2.7: DeepInsight SqueezeNet Confusion Matrix RNA-seq full feature.

2.2.1.2. DeepInsight: ResNet Full Feature

Accuracy : 95

F1-Score: 0.9473245758088609

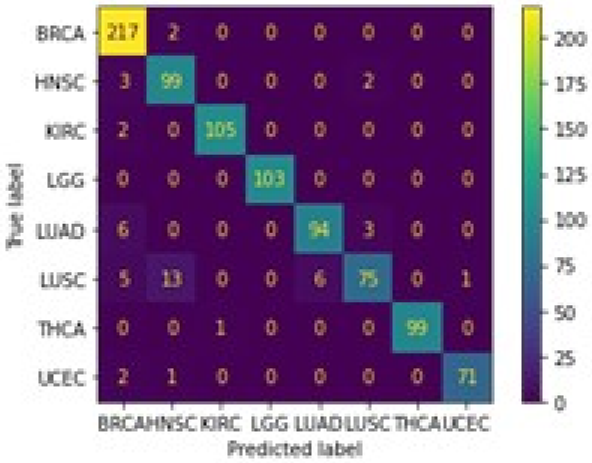


Figure 2.8: DeepInsight ResNet Confusion Matrix RNA-seq full feature.

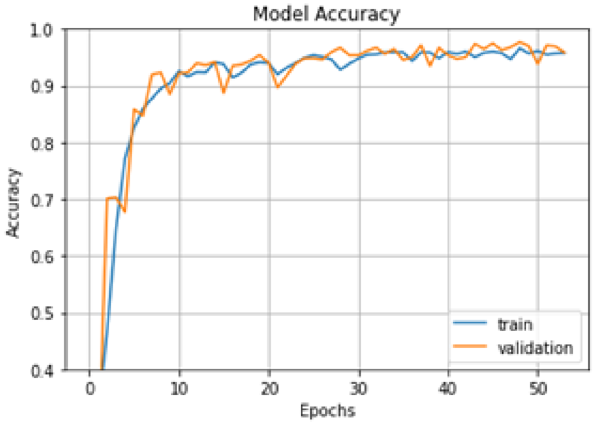


Figure 2.9: DeepInsight ResNet Model Accuracy RNA-seq full feature.

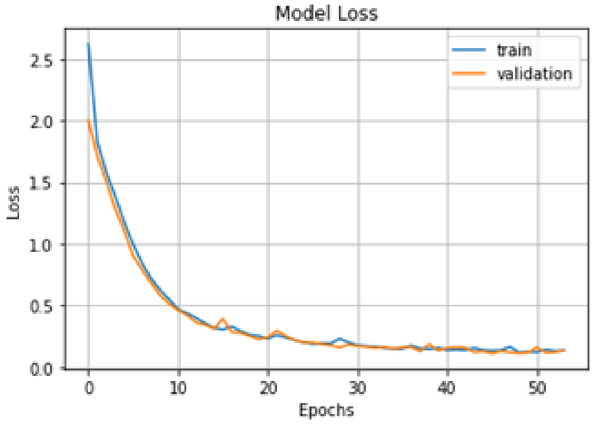


Figure 2.10: DeepInsight ResNet Model Loss RNA-seq full feature.

2.2.1.3. DeepInsight: ResNet Feature Selected

Accuracy : 96

F1-Score: 0.954752934031203

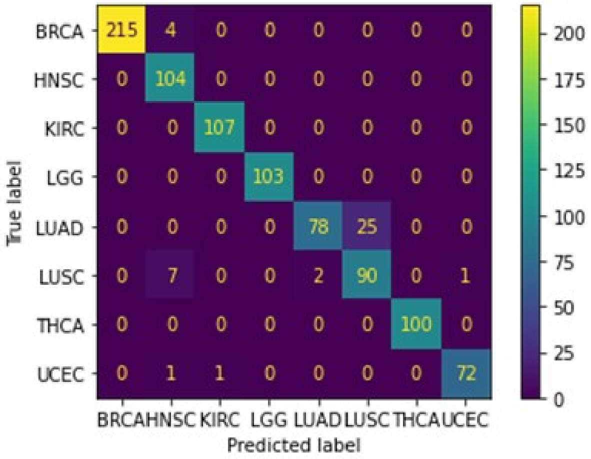


Figure 2.11: DeepInsight ResNet Confusion Matrix RNA-seq feature selected.

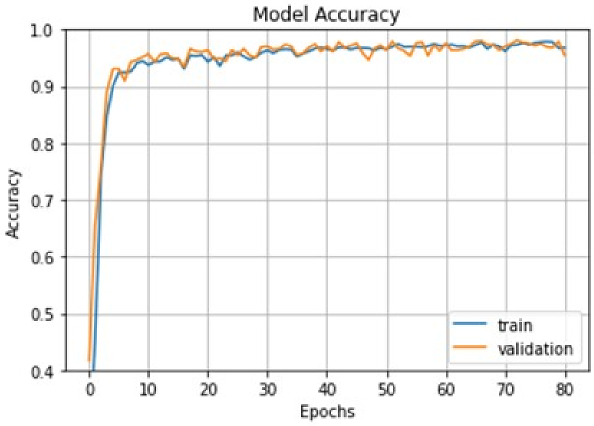


Figure 2.12: DeepInsight ResNet Model Accuracy RNA-seq feature selected.

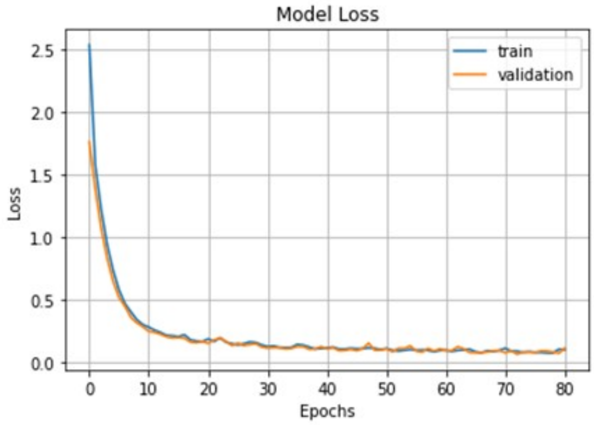


Figure 2.13: DeepInsight ResNet Model Loss RNA-seq feature selected.

2.2.2. IGTD

2.2.2.1. IGTD: ResNet Full Feature

Due to time complexity issues IGTD doesn’t work properly.[1]

2.2.2.2. DeepInsight: ResNet Feature Selected

Accuracy : 9648351648351648

F1-Score: 0.9642619017808982

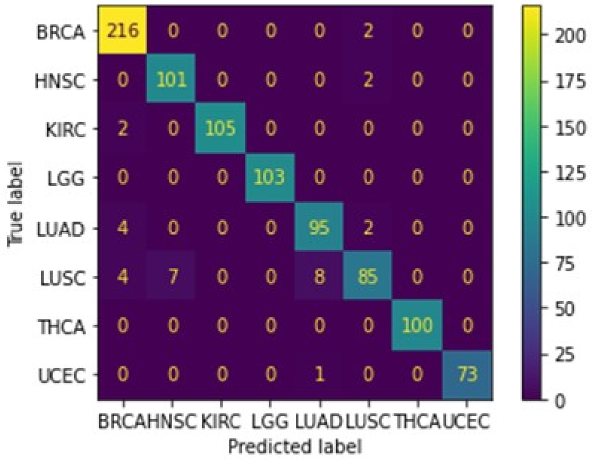


Figure 2.14: IGTD ResNet Confusion Matrix RNA-seq feature selected.

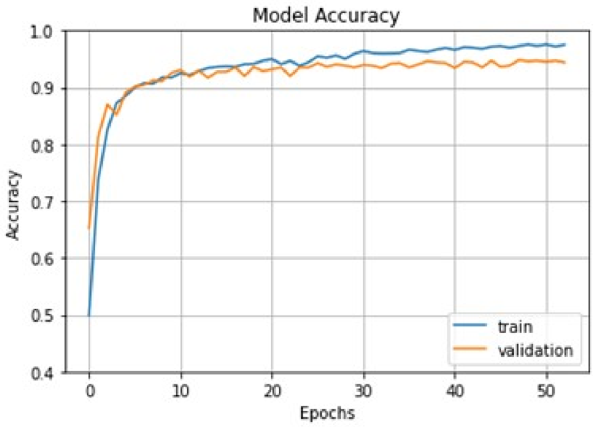


Figure 2.15: IGTD ResNet Model Accuracy RNA-seq feature selected.

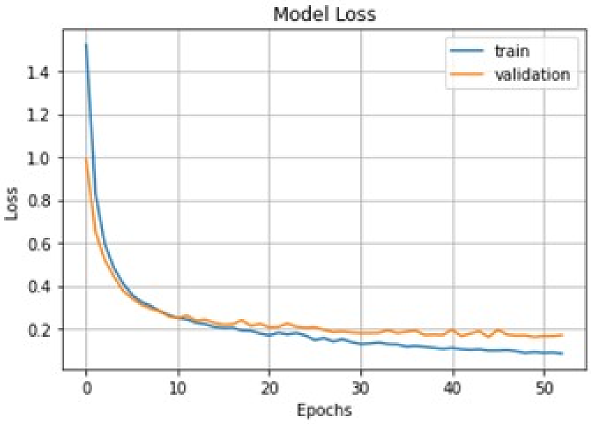
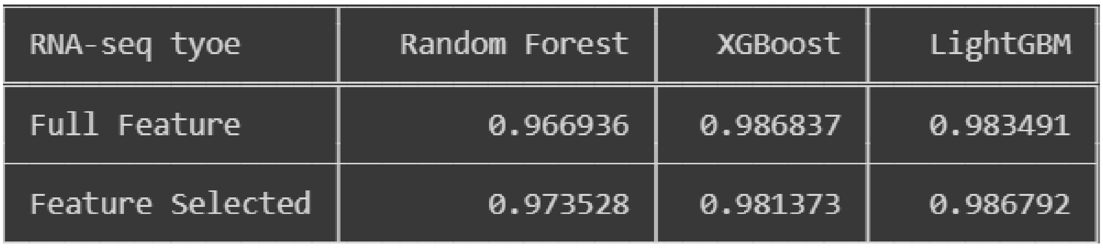


Figure 2.16: IGTD ResNet Model Loss RNA-seq feature selected.

3. COMPARISONS AND CONCLUSIONS

3.0.1. Comparison Results Between Machine Learning Models

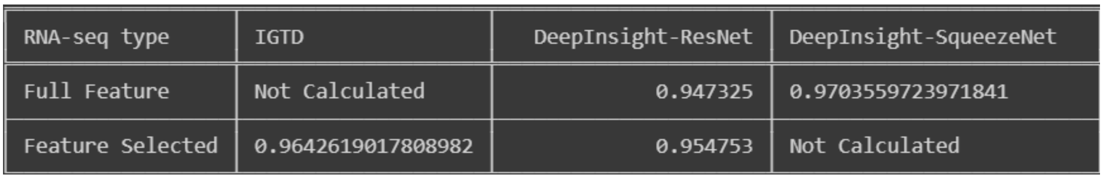


RNA-seq tyoe	Random Forest	XGBoost	LightGBM
Full Feature	0.966936	0.986837	0.983491
Feature Selected	0.973528	0.981373	0.986792

Figure 3.1: Machine Learning F1score.

It was expected that ensemble models would do better, the reason f1scores came out so high might be because the dataset is stable and tree-based machine learning models are especially viable against tabular data. In addition, the increase in f1score after feature selection in random-forest and LightGBM may be an indication that the feature selection process was performed successfully.

3.0.2. Comparison Results Between Image Creation Algorithms with CNN Architectures



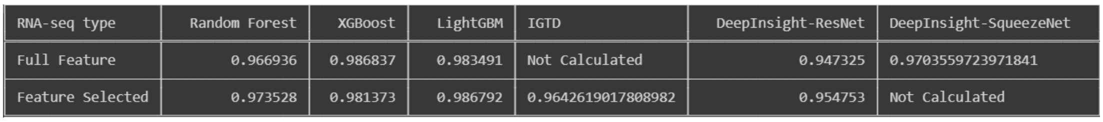
RNA-seq type	IGTD	DeepInsight-ResNet	DeepInsight-SqueezeNet
Full Feature	Not Calculated	0.947325	0.9703559723971841
Feature Selected	0.9642619017808982	0.954753	Not Calculated

Figure 3.2: Image Creation CNN F1Scores.

The biggest disadvantage of IGTD is that IGTD is not implemented in a multi-processor way, and that the time complexity in datasets with large features is quite large compared to the DeepInsight algorithm. But in small datasets, it works quite well, unlike large datasets.

Squueze net gives the best results with a margin of 0.3 points, this may be because I didn't use early stopping on squeeze net and let it get a little overfit. Apart from that, I used the early stopping parameter in order not to be overift while training in all ResNet models, I tried to get the best values by changing the parameters, and I got the f1scores shown above as a result of my intense efforts. Since the IGTD algorithm is not suitable for datasets with large features, I could not produce any images, so I used the information gain filter method, which is one of the feature filter methods, and there was a slight increase in the DeepInsight ResNet f1 score with the feature reduction, which shows that the feature selection was done properly.

3.0.3. Comparison Results Between All Models and Discussion



RNA-seq type	Random Forest	XGBoost	LightGBM	IGTD	DeepInsight-ResNet	DeepInsight-SqueezeNet
Full Feature	0.966936	0.986837	0.983491	Not Calculated	0.947325	0.9703559723971841
Feature Selected	0.973528	0.981373	0.986792	0.9642619017808982	0.954753	Not Calculated

Figure 3.3: All Classifying Models F1Scores.

If we look at all f1scores together and evaluate them holistically, we can talk about a slight, not obvious, superiority of machine learning algorithms over image creation + CNN algorithms for this dataset and test set. To confirm this hypothesis or to argue the opposite, we need to make observations with more and different datasets.

However, the increase in processing capability and the rapid developments in the field of image processing are the results that I have drawn from this project, that the image creation field will be valued day by day and that we can use Image Creation + CNN architectures instead of classical machine learning algorithms in projects for tabular data today.

BIBLIOGRAPHY

[1] Y. Zhu, “Nonmultiprocessing problem in igtd algorithm,” [Online]. Available: `https://github.com/zhuyitan/IGTD/issues/9` (visited on 01/15/2023).

[2] R. Z. Omid Bazgir, “Representation of features as images with neighborhood dependencies for compatibility with convolutional neural networks,” *NATURE COMMUNICATIONS*, 2019.

[3] Y. Zhu. “Converting tabular data into images for deep learning with convolutional neural networks.” (2021).

[4] A. Sharma, “Deepinsight: A methodology to transform a non-image data to an image for convolution neural network architecture,” *NATURE*, 2019.

[5] Y. Zhu, “Converting tabular data into images for deep learning with convolutional neural networks,” [Online]. Available: `https://github.com/CBIIT/NCI-DOE-Collab-Pilot1-Image-Generator-for-Tabular-Data` (visited on 01/15/2023).

[6] A. Sharma, “Deepinsight: A methodology to transform a non-image data to an image for convolution neural network architecture,” *NATURE*, p. 7, 2019.

[7] A. Sharma, “Deepinsight: A methodology to transform a non-image data to an image for convolution neural network architecture,” *NATURE*, 2019. [Online]. Available: `https://www.nature.com/articles/s41598-019-47765-6/figures/1` (visited on 01/15/2023).