Norwegian University
of Life Sciences

Master's Thesis 2020 30 ECTS
Faculty of Science and Technology
Professor Cecilia Marie Futsæther

Development of a user-friendly radiomics framework

Ahmed Albuni
Data Science
Faculty of Science and Technology

Abstract

The goal of this thesis is to implement an easy to use, user-friendly application to help researchers in the field of radiomics and image processing to extract radiomics features. The application also includes an easy way to test various feature selection methods and multiple machine learning algorithms.

The application named Biorad was developed using the *Python*^{TM} programming language. The code is available at https://github.com/ahmedalbuni/biorad.

The first version of Biorad was developed by Severin Langberg and it was intended for his research on head and neck cancer. The code for the first version of Biorad is available at https://github.com/gsel9/biorad.

Biorad consists of two separate modules. The first module is the features extraction which is a command-line tool that allows the users to easily extract radiomics features from a set of images, with or without masks. Available features are:

- First Order Statistics
- Shape-based features (for both two-dimensional and three-dimensional images)
- Grey Level Cooccurrence Matrix
- Grey Level Run Length Matrix
- Grey Level Size Zone Matrix
- Neighbouring Grey Tone Difference Matrix
- Grey Level Dependence Matrix

All the radiomics features in the feature extraction module are extracted using a third-party Python library called pyradiomics (Griethuysen et al., 2017).

The second module is the feature analysis which will give the user a cross-analysis of various feature selection tools and machine learning algorithms. Four different feature selection methods are available in the feature analysis module, and they are; ReliefF, Mutual Information, Fisher Score and Variance Threshold. Additionally, six different classifiers are available; Ridge, Light gradient boosting machine, C-Support Vector Classification, Decision Tree, Logistic Regression and Extra Tree Classifier.

In the testing of the application, the main dataset of 198 head and neck cancer patients was used. One hundred ninety-two radiomics features were obtained from the CT and PET scan images and 13 clinical factors were added later. Other datasets used also are the wine dataset and the Breast cancer Wisconsin (diagnostic) dataset. Two other students, Grünbeck from NMBU and Langan from NTNU, used the application to analyse their datasets, Grünbeck in her study about the Effect of Methylphenidate (MPH) treatment in Attention deficit hyperactivity disorder (ADHD) Diagnosed Children (I. A. Grünbeck, 2020). And Langan in her study regarding MRI-based radiomics analysis for predicting treatment outcome in rectal cancer (Langan, 2020).

In the head and neck cancer dataset, the ReliefF feature selector was superior to the other feature selectors used, and the most informative features to the response variable (The disease-free survival) were mostly shape features.

Acknowledgements

First, I would like to thank Prof. Cecilia M. Futsæther, my primary supervisor, for her extensive help and guidance throughout the project and to my second supervisor Prof. Oliver Tomic for his great help

Also, I would like to thank Severin Langberg, another master student who submitted his master thesis in radiomics the past semester, for all the help he provided. And to Yngve Mardal Moe for his help in Python programming. Along with my fellow students Inger Annett Grünbeck, and Isak Biringvad Lande at NMBU and Aase Mellingen Langan at NTNU, for their feedback on the application while using it for their work. And to my friend Johan Tryti for testing and verifying the installation instructions of the application. Also, to Dr Aurora Rosvoll Grøndahl for helping me with the dataset that I used for testing.

Contents

List of Figures

List of Tables

List of Abbreviations

CI	Concordance-index
CLI	Command-line interface
CSV	Comma-separated values
CT	Computed tomography
CV	Cross-validation
DT	Decision tree
ECOG	Eastern Cooperative Oncology Group
ET	Extremely randomised trees
GLCM	Grey Level Co-Occurrence Matrix
GLDM	Grey Level Dependence Matrix
GLRLM	Grey Level Run Length Matrix
GLSZM	Grey Level Size Zone Matrix
IT	Information technology
JSON	JavaScript Object Notation
KNN	K-nearest neighbours
LGBM	Light gradient boosting machine
LR	Logistic regression
MRI	Magnetic resonance imaging
NGTDM	Neighbouring Grey Tone Difference Matrix
NMBU	Norwegian University of Life Sciences
NTNU	Norwegian University of Science and Technology
PCA	Principle component analysis
PET	Positron emission tomography
RF	Random forest
Ridge	Ridge classifier
ROI	Region of interest
SVC	C-Support vector classifier

1 Introduction

Radiomics is a process that extracts quantitative numerical features from medical images. Radiomics began in the field of oncology - the study and the treatment of cancer tumours, but it has the potential to be used in other diseases (Gillies et al., 2015). Moreover, there are possibilities of using radiomics in areas other than medical research (Lande, 2020). There are multiple types of medical images, including computed tomography (CT) scans, magnetic resonance imaging (MRI) scans, positron emission tomography (PET) scans and ultrasound (Bogowicz et al., 2019),(Chaddad et al., 2019). In radiomics, the extracted quantitative numerical features can describe the shape, size and the texture of a cancer tumour to help in diagnosis and selecting a proper treatment (*Biological Basis of Radiomcs | ELife*, n.d.). Radiomics features can provide additional information on top of the clinical data, as shown in research (*Biological Basis of Radiomcs | ELife*, n.d.).

The radiomics field has several challenges like the lack of standardization of the radiomics analysis, which affects the reproducibility of the results (Griethuysen et al., 2017). This issue had been addressed by an open-source Python package called pyradiomics which offer the user a framework to extract both two-dimensional and three-dimensional features from images (*Pyradiomics Documentation*, n.d.).

However, using radiomics in research requires programming knowledge and a deep understanding of machine learning. As a result, the need to make radiomics simpler for researchers becomes apparent. Having user-friendly tools to extract radiomics features and analyse them will help researchers to utilise the potentials of radiomics and machine learning without the need for programming knowledge.

The main goal of this thesis is to develop user-friendly tools to extract radiomics features from various images. Furthermore, these radiomics features will be analysed with multiple feature selectors and machine learning classifiers. These tools should not require any programming knowledge, and the usage instructions should be easy to understand for non-IT expert users.

The tools were tested for user-friendliness by collaborating with other master students at NMBU and NTNU in order to gain valuable input and feedback. Having other students from our university, Norwegian University of Life Sciences, and from the Norwegian University of Science and Technology (NTNU), working with image data that requires analysis made developing these tools more interesting. Working in parallel with these students helped all of us in getting continuous real-time feedback.

In this thesis, there will be a brief introduction about radiomics and its importance and challenges, a description of the package used to extract the radiomics features (pyradiomics), a list of radiomics features extracted by pyradiomics, as well as descriptions of the two separate modules of the Biorad application (the feature extraction and the analysis), the features selectors used in the analysis module, the machine learning classifiers, and results of using the application on various datasets with various settings.

2 Radiomics

According to research, radiomics can provide some insights regarding the clinical characteristics of cancer tumours, such as the spread of the cancer cells, predictions of treatment outcomes and the likelihood of the disease-free survival of the patient (Gillies et al., 2015). However, those characteristics have not yet been linked to the actual biological process of cancer tumour development and spread (Gillies et al., 2015).

The predictive power of the radiomics features is shown in Figure 2-1. The clinical data is the most informative to predict the outcome. However, by combining the radiomics data with it, we can produce a more robust estimation model (*Biological Basis of Radiomcs | ELife*, n.d.).

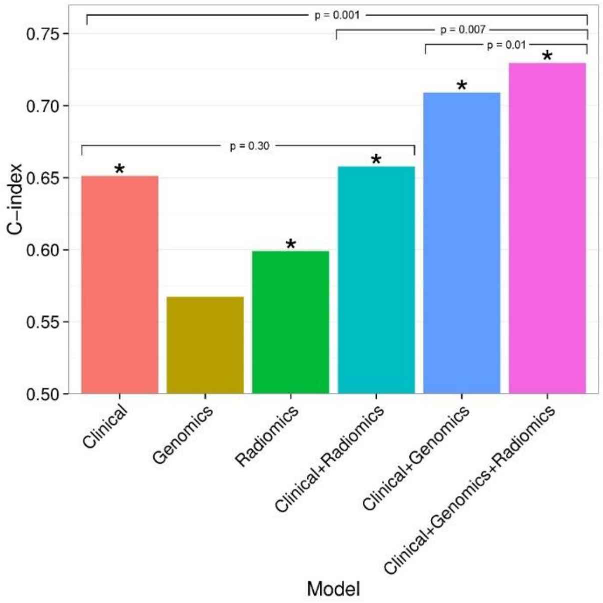


*Figure 2-1: Concordance-index (CI) showing the importance of Radiomics features as compared to Clinical and Genomics features (*Biological Basis of Radiomcs | ELife*, n.d.)*

Radiomics consists of several steps, image acquisition, image pre-processing, defining the area of the tumour – the region of interest also called image segmentation, and lastly applying machine learning for feature selection, and predicting the response variable.

2.1 Image acquisition

Radiomics starts with image acquisition; some of the most common medical images are:

- Computerized tomography (CT) scan: multiple X-ray images are taken from different angles and combined by an algorithm to create slices of a three-dimensional image (*CT Scan - Mayo Clinic*, n.d.).

- Positron emission tomography (PET) scan: a radioactive drug is either injected or swallowed by the patient, then the scan captures how different tissues and organs react to the drug. Radioactive glucose, for example, is used because cancer cells consume more energy than healthy cells. This may sometimes detect cancer cells earlier than other imaging tests (*PET - Mayo Clinic*, n.d.).
- Magnetic resonance imaging (MRI): a magnetic field and computer-generated radio waves create highly detailed images of the scanned area of the body (*MRI - Mayo Clinic*, n.d.).

2.2 Image pre-processing

The next step of radiomics is the image pre-processing; medical images can be affected by artefacts. For CT scan images, the most common types of artefacts are metal streaks, mostly from dental fillings. This can be seen in Figure 2-2. Another common type of artefact is beam-hardening, where the edges of an object such as bone appear brighter than the centre (*Artifacts and Partial-Volume Effects – UTCT – University of Texas*, n.d.).

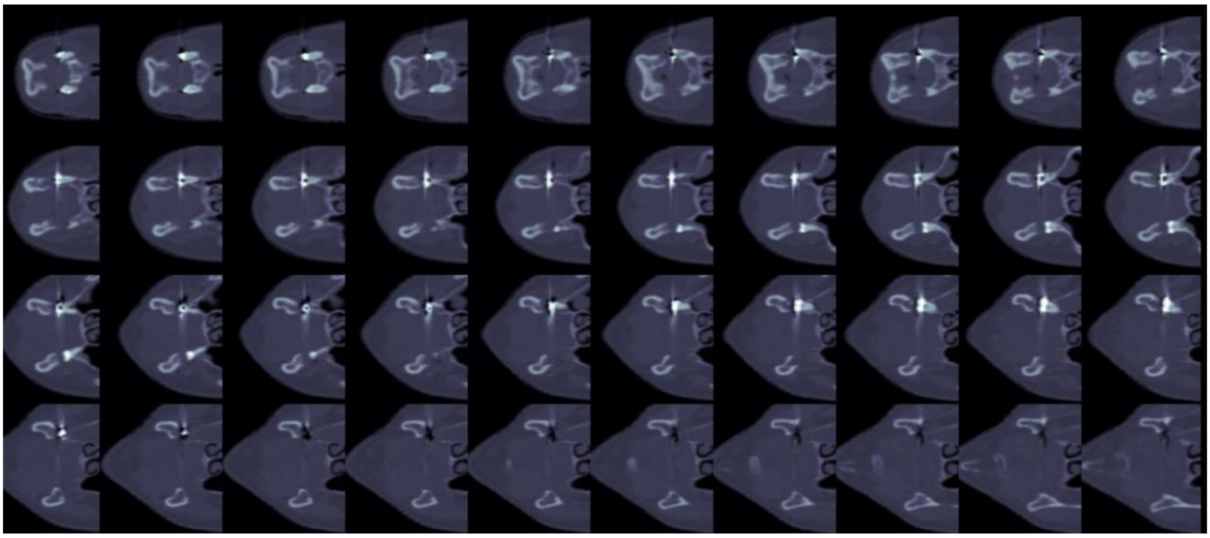


Figure 2-2 Stacks from a CT scan image for a cancer patient that shows streaks from a dental filling.

2.3 Image segmentation

One of the essential steps in radiomics is the image segmentation. It is a very challenging and critical step because all the next steps of feature generation will be done based on the segmented image (Gillies et al., 2015).

In image segmentation we define the Region of Interest (ROI), the unfolded stacked CT image is shown in Figure 2-3, and one slice of the CT scan with the mask is shown in Figure 2-4.

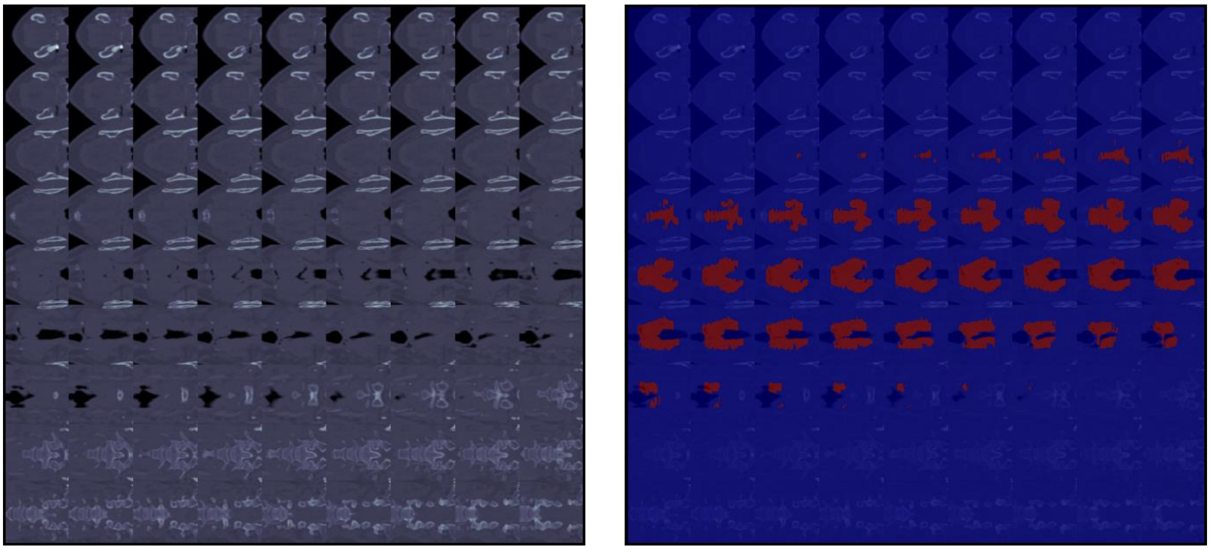


Figure 2-3 Stacked CT scan images on the left, the mask that identifies the cancer tumour (Region of Interest) is shown on the right image.

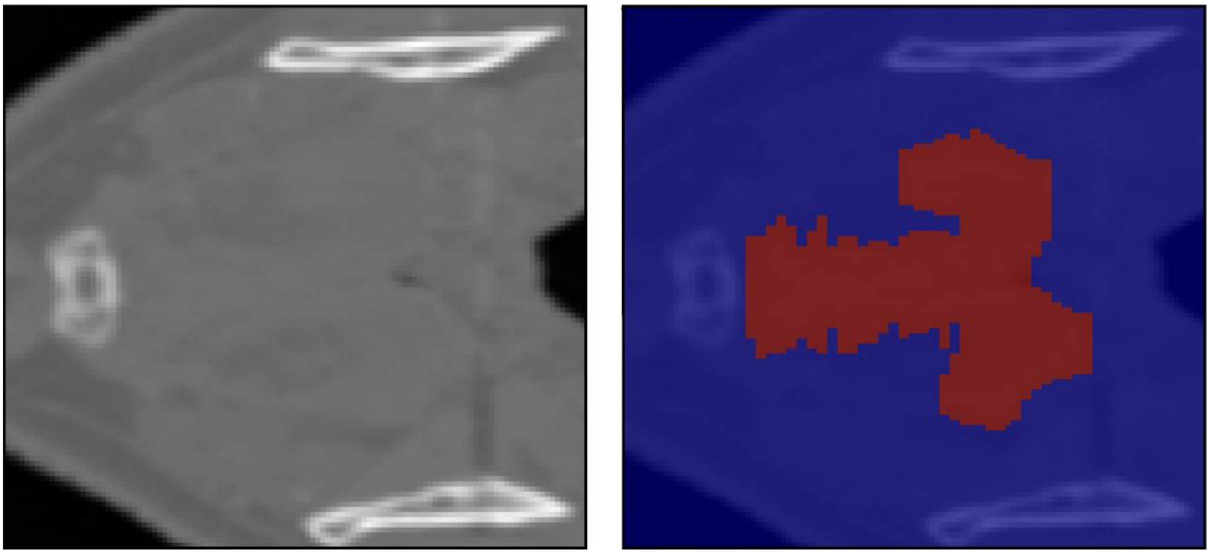


Figure 2-4 One slice of the stacked images of a CT scan. The mask applied to the right picture shows the ROI (Region of Interest).

Segmentation can be done either manually, semi-automated or fully automated. In many research studies, the manual segmentation by experts is considered as the ground truth (*Radiomics: The Process and the Challenges*, n.d.). However, there are many issues with manual segmentation. First, it suffers from high inter-reader variability. Second, it takes a very long time from the expert readers. Many semi-automated and fully automated segmentation methods have been developed for various regions like the brain, lung and breast, and for various image types, like CT, PET and MRI scan images. All segmentation methods should be as automated as possible, with minimal human interaction, and the results should be reproducible (*Radiomics: The Process and the Challenges*, n.d.).

2.4 Radiomics features

After image pre-processing and image segmentation, we can extract the radiomics features which can be divided into three groups, size and shape-based features, image intensity histogram or first-order features and features regarding the relationships between image voxels (Rizzo et al., 2018).

2.4.1 Size and shape-based features

Size and shape-based features are extracted using the masks only (the mask is what defines the region of interest (ROI)), which means that shape features are independent of the distribution of grey level intensities in the image. Examples of shape features are volume, surface, maximum diameter and sphericity – which is a measure of roundness.

2.4.2 Image intensity histogram features

Image intensity histogram features, also known as the first-order features involve the histogram and is generated based on the intensity level, and the number of bins as shown in Figure 2-5, and in Figure 2-6. That shows how different bins can affect the histogram and hence the features extracted from it.

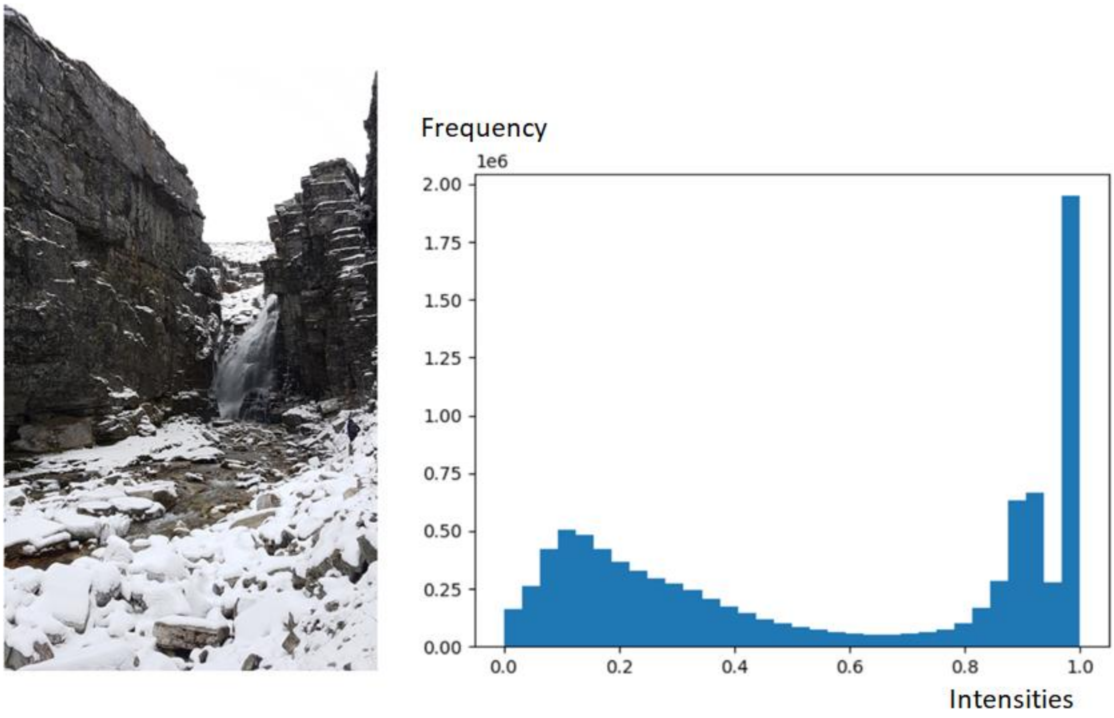


Figure 2-5 The graph on the right side shows a histogram for the image on the left side after converting it to greyscale, 32 bins used in this histogram.

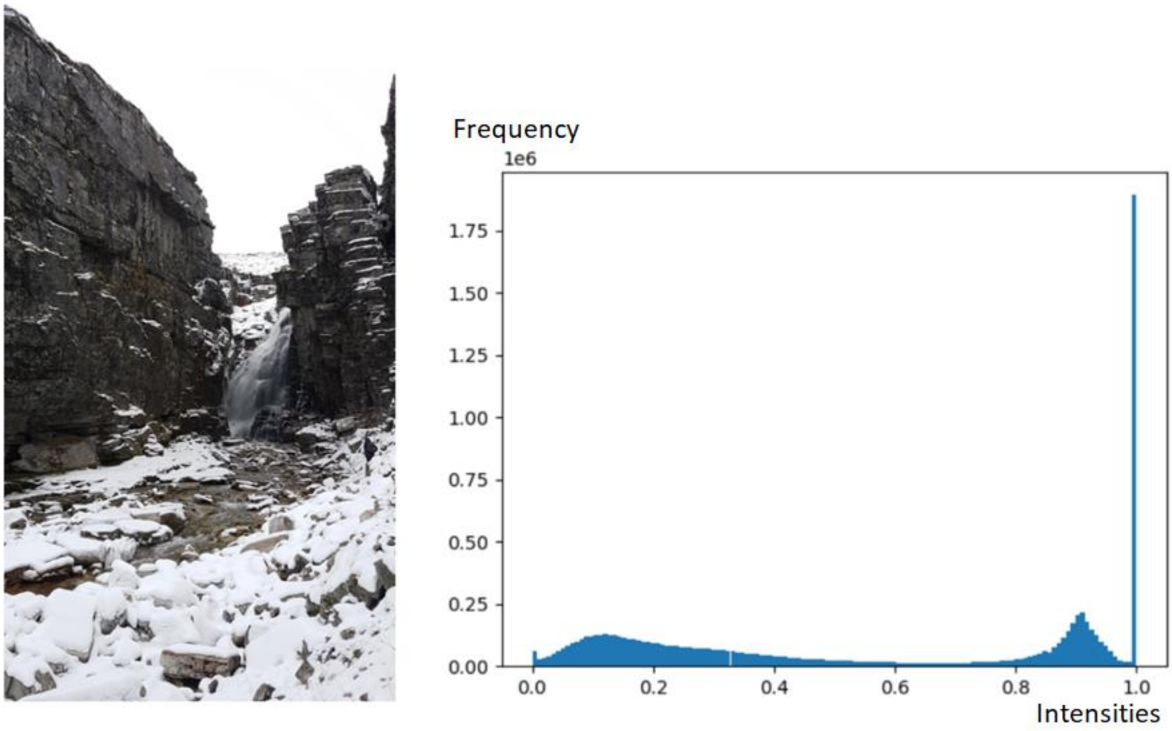


Figure 2-6 The graph on the right side shows a histogram for the image on the left side after converting it to greyscale, 128 bins used in this histogram.

Examples of histogram (first-order) features are mean, maximum, minimum, median, range, kurtosis illustrated in Figure 2-7 and skewness illustrated in Figure 2-8.

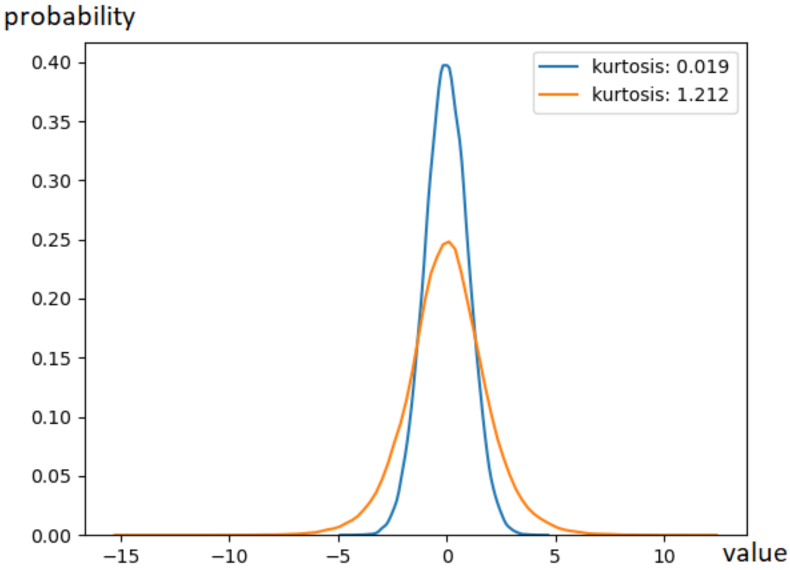


Figure 2-7 Kurtosis values of a normal distribution and a logistic distribution, the fisher=False which means that 3 is subtracted from the kurtosis value

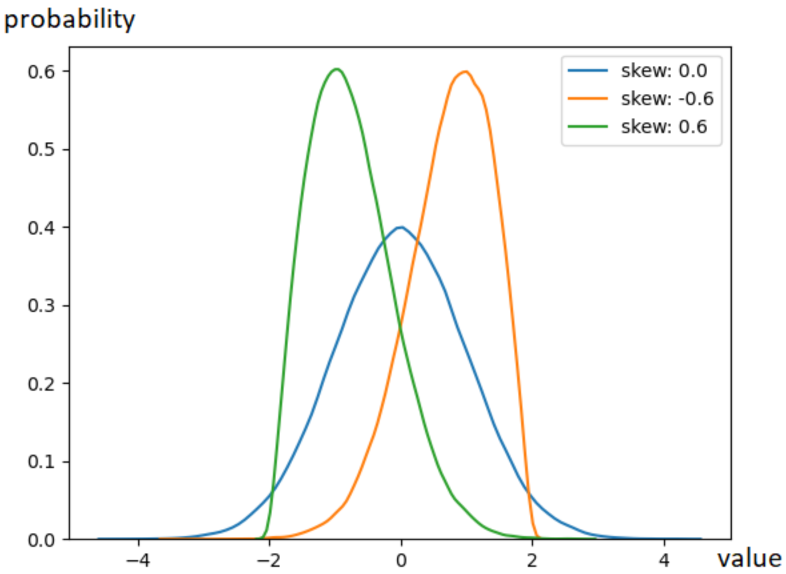


Figure 2-8 Normal distribution with skewness = 0, positive skewness and negative skewness

2.4.3 Voxels relationship features

Voxels relationship features are features regarding the relationships between image voxels such as the Grey Level Cooccurrence Matrix (GLCM), Grey Level Run Length Matrix (GLRLM), Grey Level Size Zone Matrix (GLSZM) and Neighbouring Grey Tone Difference Matrix features (Griethuysen et al., 2017). These features describe the changes in the images that cannot be described using the histogram, as shown in Figure 2-9.

A table of the relationship between the voxels is constructed as shown in Figure 2-10 This figure shows how the GLCM table is constructed. An example of GLCM features is the contrast, two images with different contrast are shown in Figure 2-14.

GLRLM table construction is illustrated in Figure 2-11, and GLSZM construction is illustrated in Figure 2-12, and finally, the NGTDM construction is illustrated in Figure 2-13.

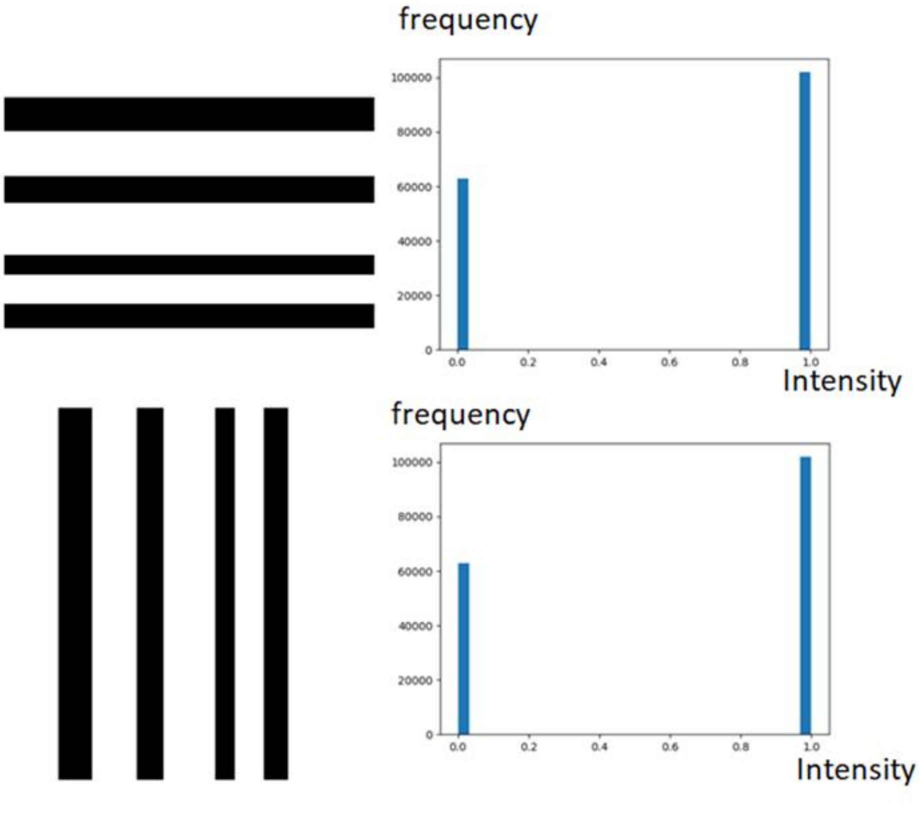
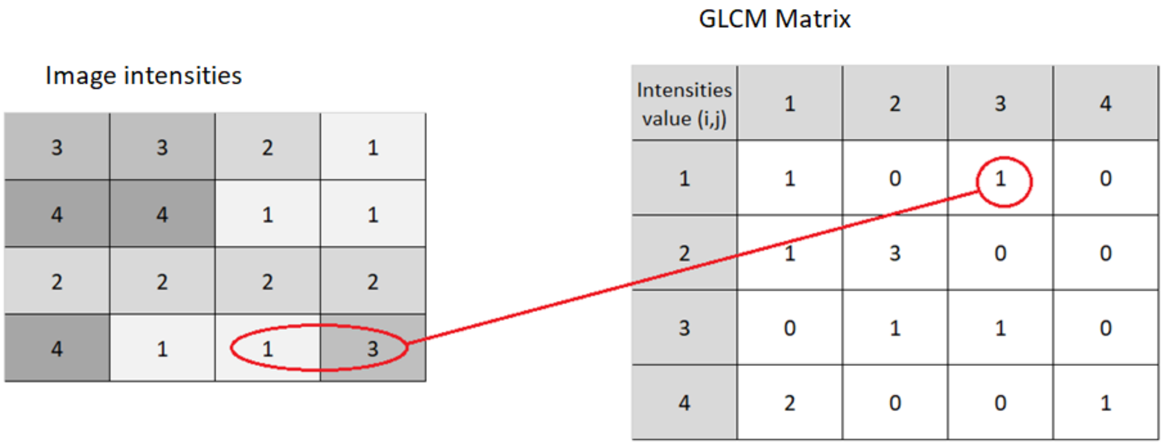


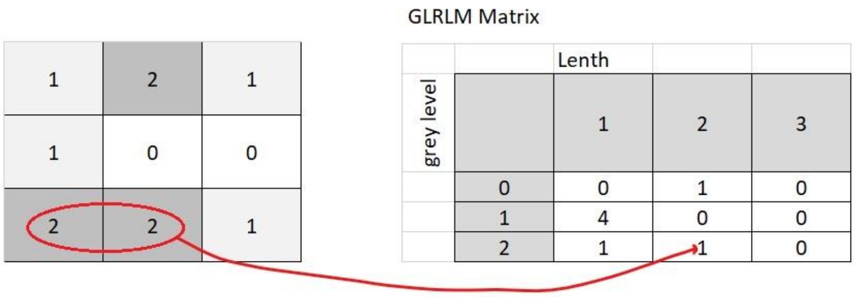
Figure 2-9 Different images can have similar histograms. The histograms on the right side are similar, but they represent the different images shown on the left.



3	3	2	1
4	4	1	1
2	2	2	2
4	1	1	3

Intensities value (i,j)	1	2	3	4
1	1	0	1	0
2	1	3	0	0
3	0	1	1	0
4	2	0	0	1

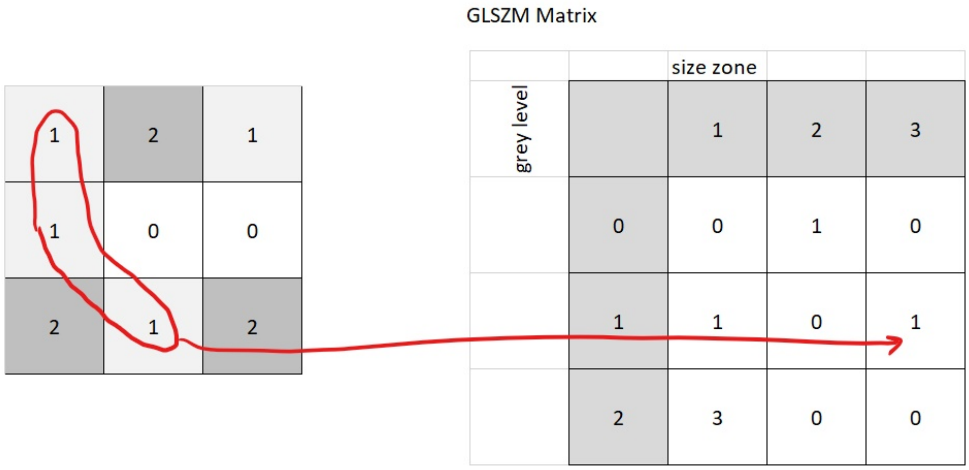
Figure 2-10 This figure shows how the GLCM table is constructed. The direction chosen here is from left to right, and the GLCM matrix shows the combination of the two values frequency



1	2	1
1	0	0
2	2	1

		Lenth		
grey level		1	2	3
	0	0	1	0
	1	4	0	0
	2	1	1	0

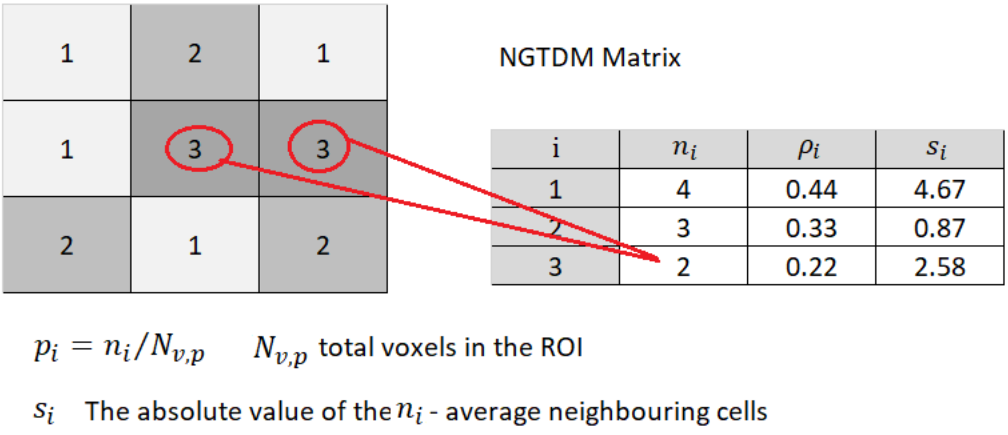
Figure 2-11 This figure shows how the GLRLM matrix is constructed. The direction chosen here is from left to right, and the GLRLM matrix shows the length of the "run".



1	2	1
1	0	0
2	1	2

		size zone		
grey level		1	2	3
	0	0	1	0
	1	1	0	1
	2	3	0	0

Figure 2-12 This figure shows how the GLSZM matrix is constructed. The values show the frequency of the size zones for each grey level



1	2	1
1	3	3
2	1	2

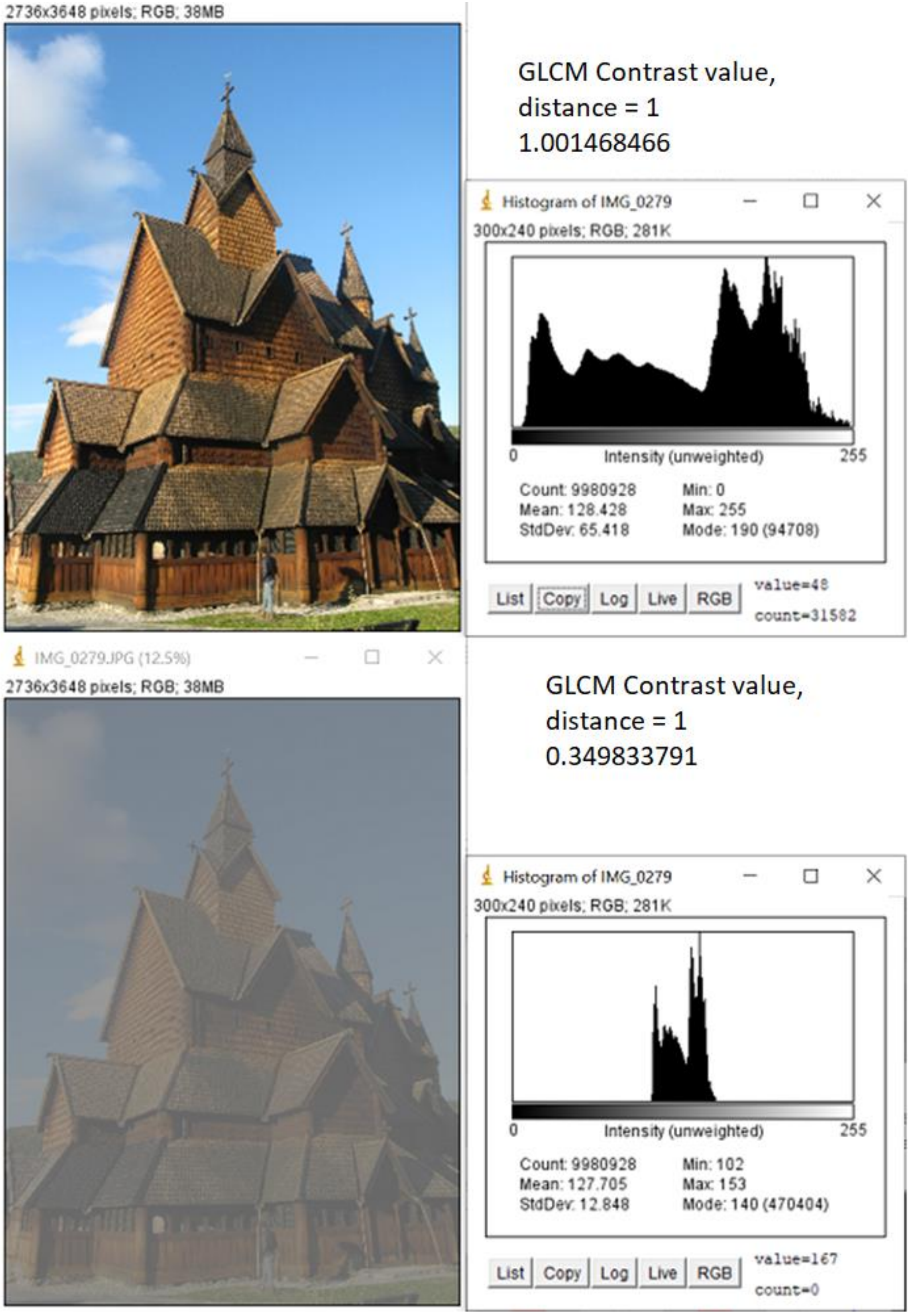
NGTDM Matrix

i	n_i	ρ_i	s_i
1	4	0.44	4.67
2	3	0.33	0.87
3	2	0.22	2.58

$p_i = n_i/N_{v,p}$ $N_{v,p}$ total voxels in the ROI

s_i The absolute value of the n_i - average neighbouring cells

Figure 2-13 This figure shows how the NGTDM matrix is constructed, on the left is the image intensities, and the NGTDM matrix is on the right



2736x3648 pixels; RGB; 38MB

GLCM Contrast value,
distance = 1
1.001468466

Histogram of IMG_0279

300x240 pixels; RGB; 281K

Count: 9980928 Min: 0
Mean: 128.428 Max: 255
StdDev: 65.418 Mode: 190 (94708)

value=48
count=31582

IMG_0279.JPG (12.5%)

2736x3648 pixels; RGB; 38MB

GLCM Contrast value,
distance = 1
0.349833791

Histogram of IMG_0279

300x240 pixels; RGB; 281K

Count: 9980928 Min: 102
Mean: 127.705 Max: 153
StdDev: 12.848 Mode: 140 (470404)

value=167
count=0

*Figure 2-14 The original image is on the top; the image on the bottom is modified to have low contrast value as shown in the histogram. Histogram images were generated using ImageJ (*ImageJ*, n.d.)*

3 Materials and Methods

All tools used in this research were developed in the Python programming language. The radiomics features extractions were completed with the help of pyradiomics package (*Computational Radiomics System*, n.d.).

The machine used for testing was running Windows 10 on Intel Core i7 8th Generation, eight cores, 8 GB of RAM.

3.1 Datasets

The Biorad application was tested using several datasets, the head and neck cancer, the Breast cancer Wisconsin (diagnostic) and the wine datasets.

The head and neck cancer dataset was the primary dataset used to test the application. The dataset includes CT scan images and 18F-fluorodeoxyglucose PET scan images of 198 cancer patient that received radiotherapy at Oslo University Hospital between January 2007 and December 2013. Details about the dataset and value distribution are highlighted in Appendix A.1 Head and neck cancer patients' dataset. A sample from the dataset is shown in Figure 3-1, and Figure 3-2. The clinical data in Appendix A was available for all patients and were added to the features list.

For binary classification, the disease-free survival data was used. The dataset is balanced, out of 198 patients, the survival rate was 45.5%.

Langberg used the same dataset in his thesis (Langberg, 2019), and we are going to compare his results with the results obtained from the Biorad application.

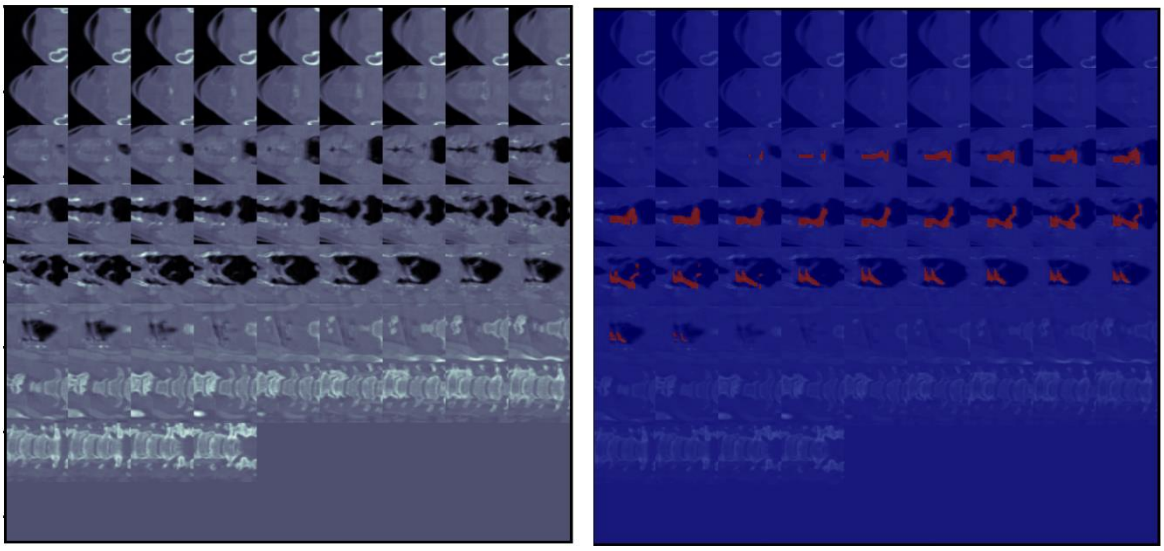


Figure 3-1 Stacked CT scan images on the left, the mask that identifies the cancer tumour (Region of Interest) is shown on the right image. The images are from one of the patients in the head and neck cancer dataset

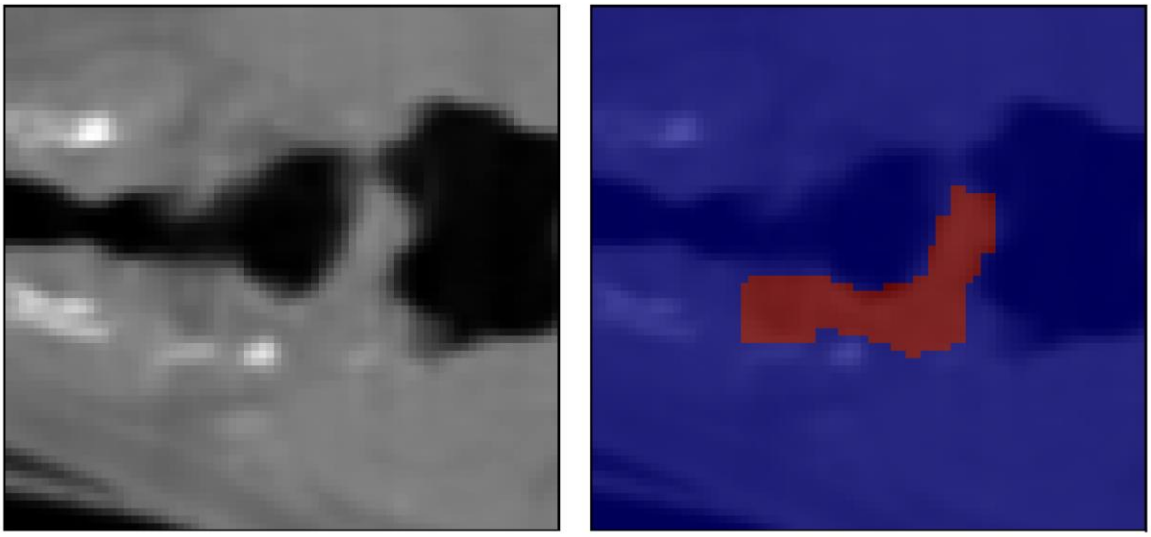


Figure 3-2 One slice from a cancer patient CT scan, with the mask applied to the right image

Both Breast cancer Wisconsin (diagnostic) and the wine recognition datasets are part of the scikit-learn datasets (*Dataset - Scikit-Learn*, n.d.). The wine recognition dataset has 178 samples, 3 classes and 13 features, while the Breast cancer Wisconsin (diagnostic) dataset has 569 samples, 2 classes and 30 features. Details on these two datasets can be found in Appendix A.2 Wine recognition dataset and Appendix A.3 Breast cancer Wisconsin (diagnostic) dataset.

3.2 Grid and Randomized Search CV

Grid Search CV is an exhaustive model selection from Scikit-learn. It will check all the combinations of different hyper-parameter values to get the best model. It can be very slow for large datasets, or a large domain of hyperparameters, which makes it not practical in some cases (Raschka & Mirjalili, 2017).

Same as the GridSearchCV, the RandomizedSearchCV is a model selection approach from Scikit-learn. The difference is that in RandomizedSearchCV, only a fixed number of parameter settings is picked from the distribution domain, the values of the parameters are picked randomly and not every combination is tested. Figure 3-3 shows the difference between the grid layout and the random layout (Bergstra & Bengio, 2012).

GridSearchCV is optimal for small domains of hyperparameters. Otherwise, GridSearchCV can take a very long time to fit if the domain of hyperparameters to choose from is big. RandomizedSearchCV can give us very close results to GridSearchCV much faster. The model performance might be slightly lower than GridSearchCV, but usually, that would not be carried over to the hold-out test set (*Comparing Randomized Search and Grid Search - Scikit-Learn*, n.d.).

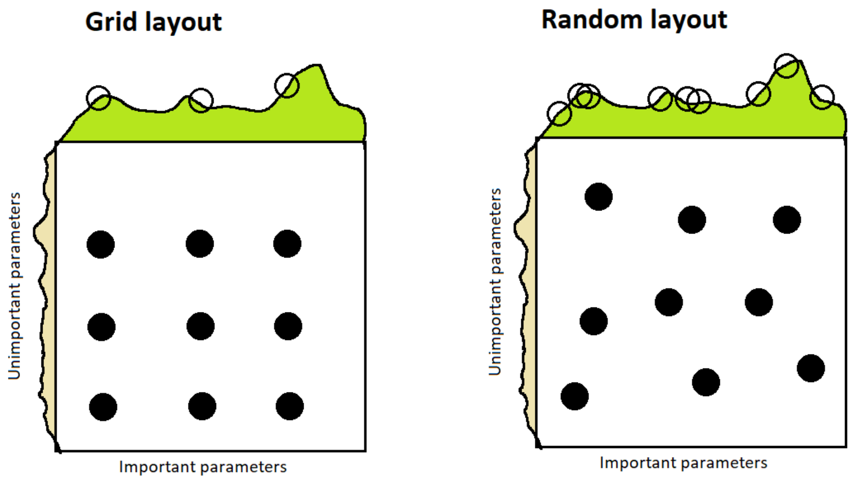
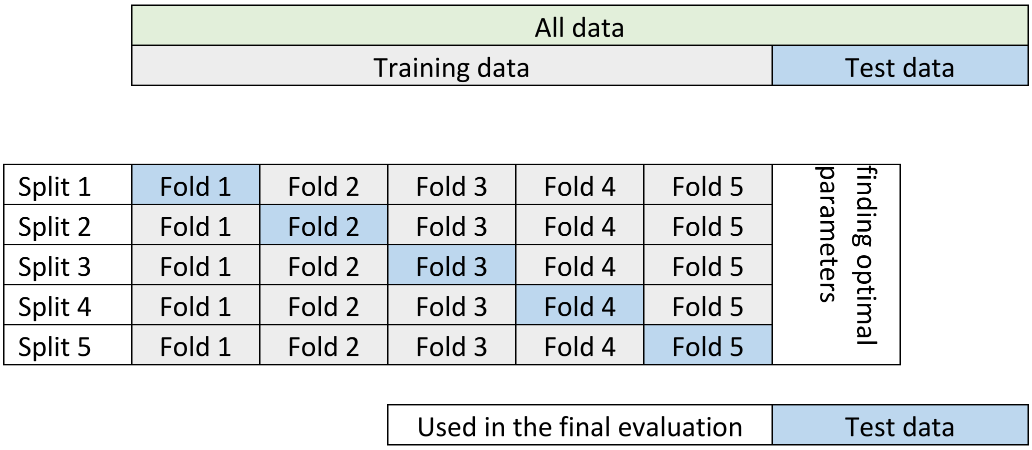


Figure 3-3 This graph shows the difference in the layout between the grid, and the random search, in the X-Axis we have the important parameters to tune, and in the Y-Axis the unimportant parameters, with 9 iterations, we see that the grid search tests only 3 combinations of the important variables, whereas the random search tests 9 different combinations. Modified from (Bergstra & Bengio, 2012)

RandomizedSearchCV uses sampling without replacement if all the tuning parameters are presented as a list. If at least one of the parameter is a distribution, then sampling with replacement is used when selecting the training set samples(*RandomizedSearchCV - Scikit-Learn*, n.d.). RandomizedSearchCV uses the k-folds cross-validation, which is illustrated in Figure 3-4



All data	
Training data	Test data

Split 1	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	finding optimal parameters
Split 2	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	
Split 3	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	
Split 4	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	
Split 5	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	

Used in the final evaluation	Test data

Figure 3-4 k-folds cross-validation with k=5 (Raschka & Mirjalili, 2017).

3.3 Pyradiomics

Pyradiomics is an open-source package written in the Python programming language to extract radiomics features for images. The aim of this package was to establish a reference standard for radiomics to assist for reproducibility of results. This package supports both the features extractions for 2-dimensional and 3-dimensional images (*Radiomic Features - Pyradiomics*, n.d.).

Since pyradiomics uses an open-source library called SimpleITK to load and handle images, the same applies to the Biorad tool. However, currently, the Biorad application has been tested for the following image formats only: TIFF, NRRD and Nifty.

The mask is what defines the Region Of Interest (ROI), as shown in Figure 2-3 Stacked CT scan images on the left, the mask that identifies the cancer tumour (Region of Interest) is shown on the right and in Figure 2-4 One slice of the stacked images of a CT scan. The mask applied to the right picture shows the ROI (Region of Interest). The provided mask should match the image dimensions, and this application assumes the value '1' represents the area to be cropped. Pyradiomics supports using different values, but it should be passed in a parameter called 'label'. However, in Biorad, the value should not be other than '1'.

If no mask is provided, then the application will create a mask that covers the whole image; in this case, the shape features will not be extracted.

3.4 Pyradiomics Features

Features that can be selected are:

- First Order Statistics (19 features).
- Shape-based (3D) (16 features) – Mask should be provided, and the provided image should have three dimensions.
- Shape-based (2D) (10 features) – Mask should be provided, and the provided image should have two dimensions.
- Grey Level Cooccurrence Matrix (24 features) – Default distance is 1, the user can select other values.
- Grey Level Run Length Matrix (16 features).
- Grey Level Size Zone Matrix (16 features).
- Neighbouring Grey Tone Difference Matrix (5 features) – Default distance is 1, the user can select other values.
- Grey Level Dependence Matrix (14 features)- default cut-off value is zero, the user can select other values (*Computational Radiomics System*, n.d.).

Details of the pyradiomics features are available in Appendix C: Pyradiomics features.

3.5 SimpleITK

SimpleITK is an open-source image analysis library, available in multiple programming languages, including Python (Lowekamp et al., 2013).

In pyradiomics, the loading and the pre-processing of the images are done by SimpleITK, for that reason, image formats that are supported by SimpleITK are supported by pyradiomics library (*Pyradiomics Documentation*, n.d.). The image formats that have been tested in this thesis are NRRD, Nifty and TIFF.

Care should be taken while converting images from one format to another. Spacing and direction are two properties that can affect the images if not taken into consideration while converting an image from one format to another. The spacing describes the scale of the pixels in each axis, and wrongly assigning image direction can rotate the image (*SimpleITK Documentation*, n.d.). For a demonstration, see Figure 3-5.

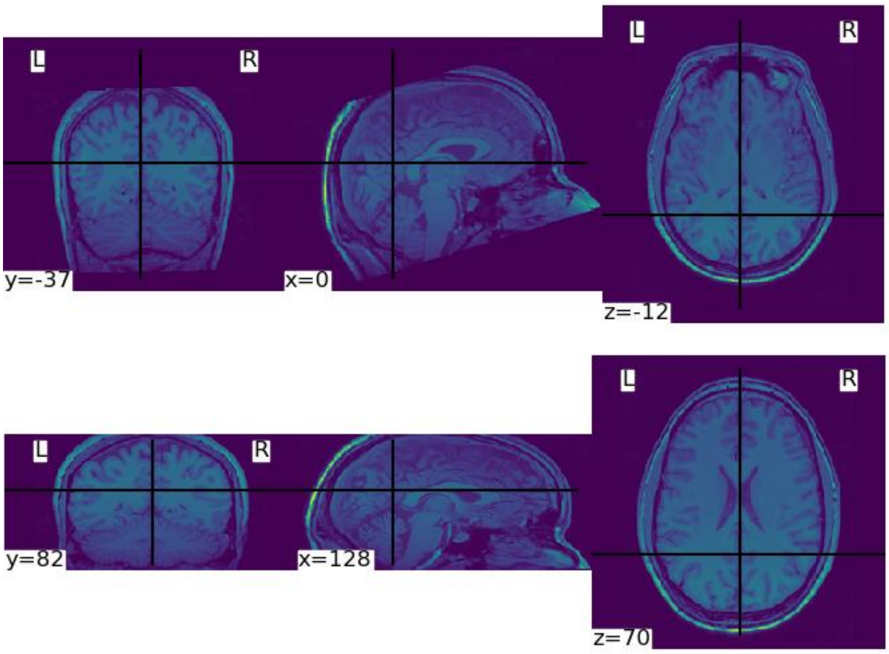


Figure 3-5 The original MRI image on top, and at the bottom, the image after conversion from nifty to NRRD without preserving the spacing and the direction.

3.6 Scikit-learn

Scikit-learn is a free Python machine learning library (Pedregosa et al., 2011). It includes various classification and regression algorithms and various feature selection algorithms as well (*Scikit-Learn*, n.d.).

In this thesis, the following classifiers from scikit-learn were used:

- Extra Trees Classifier
- Ridge Classifier
- Logistic Regression
- Decision Tree Classifier
- C-Support Vector Classification (SVC)

And the following feature selectors were used:

- Mutual information classifier, mutual_info_classif
- Univariate feature selector with configurable strategy, GenericUnivariateSelect
- Variance Threshold, VarianceThreshold

A few other classifiers and feature selectors were also used, and they are covered in later sections.

3.7 Feature selection

In most real-world classification problems, many of the candidate features are often partially or entirely irrelevant to the target value or are redundant. Those features do not add anything to the target value. Furthermore, with large datasets, it is necessary to reduce the number of features to improve the running time of the classification algorithms (Dash & Liu, 1997).

Having irrelevant features in the dataset can negatively affect the performance and the accuracy of the models because it makes the model learn based on those irrelevant features. Feature selection is done either by manually or automatically selecting those features which are more descriptive to the response variable (Shaikh, 2018). Therefore, in addition to reducing the training time, using feature selection can also improve accuracy by minimising the misleading data. Moreover, reducing the overfitting - the less amount of redundant data, results in the lesser the chance of making decisions based on noise. Keeping irrelevant data in the dataset can cause the machine-learning algorithm to make decisions based on those data that can be by chance relevant only to the training set, and for the test set the result will be negatively affected (Brownlee, 2014). The error in the classifiers usually decreases then increases as the number of features grows (Hua et al., 2005). For datasets with small samples, a large number of features can result in overfitting, and it is suggested that the optimal number of features is the optimal feature size which is around $n - 1$ where n is the number of samples (Hua et al., 2005).

Five feature selection algorithms are used in this application, Univariate Filter Methods (Mutual Information, Fisher Score) and Multivariate Filter Methods (ReliefF) and the Variance Threshold. These algorithms were chosen because they run fast for a large number of features, and many of them gave good results for an experiment done by Langberg in his thesis (Langberg, 2019)

3.7.1 Univariate Filter Methods

Univariate filter methods for feature selection examine each feature individually and examine it for its relationship with the response variable. These methods are simple and fast to run, and they give a good understanding of the data. However, they are not always good in optimizing the features for better generalization and can lead to a sub-optimal subset of features (*Feature Selection – Part I*, n.d.).

Two univariate filter methods are used in Biorad application; Mutual information and Fisher score.

3.7.1.1 Mutual Information

Mutual information selector estimates the values of the mutual information (MI) between the feature and the response variable, which is a non-negative value that measures the dependencies between two variables. The only parameter available in Biorad for mutual information is the number of features to select (Brown et al., 2012).

The mutual information selector used in Biorad is part of the scikit-learn Python package.

3.7.1.2 Fisher Score

Fisher score is one of the most used feature selection methods; it is a measure of the amount of information a variable is carrying about another variable. Fisher score has the same limitation as other univariate filter methods (Gu et al., n.d.).

The Fisher score method used in Biorad is part of the skfeature-chappers Python package (Siu, 2017/2020).

3.7.2 Multivariate Filter Methods

In Biorad, one multivariate filter method is used, which is the ReliefF. While univariate methods only examine one feature at a time, the multivariate filter methods consider the mutual relationship between features. For that reason, multivariate filter methods are effective in removing the redundancy in features (R. J. Urbanowicz, Meeker, et al., 2018).

3.7.2.1 ReliefF

ReliefF assigns scores for all the features. These scores range from -1 (worst) to 1 (best). The weight estimates the relevance of the feature to the response variable and since it is a multivariate filter method, it takes into account the relationship between the features (R. J. Urbanowicz, Meeker, et al., 2018).

For tuning, in addition to the number of features to select, one more hyperparameter can be tuned in Biorad which is the number of neighbours (n_neighbors). The n_neighbors defines the number of neighbours to consider in assigning features scores, for more clarification, refer to Figure 3-6. Larger numbers may give more accurate scores but it takes a longer time to process (*Using Skrebate - Scikit-Rebate*, n.d.).

ReliefF method used in Biorad is a part of skrebate Python package (R. S. O. Urbanowicz Pete Schmitt, and Ryan J., n.d.).

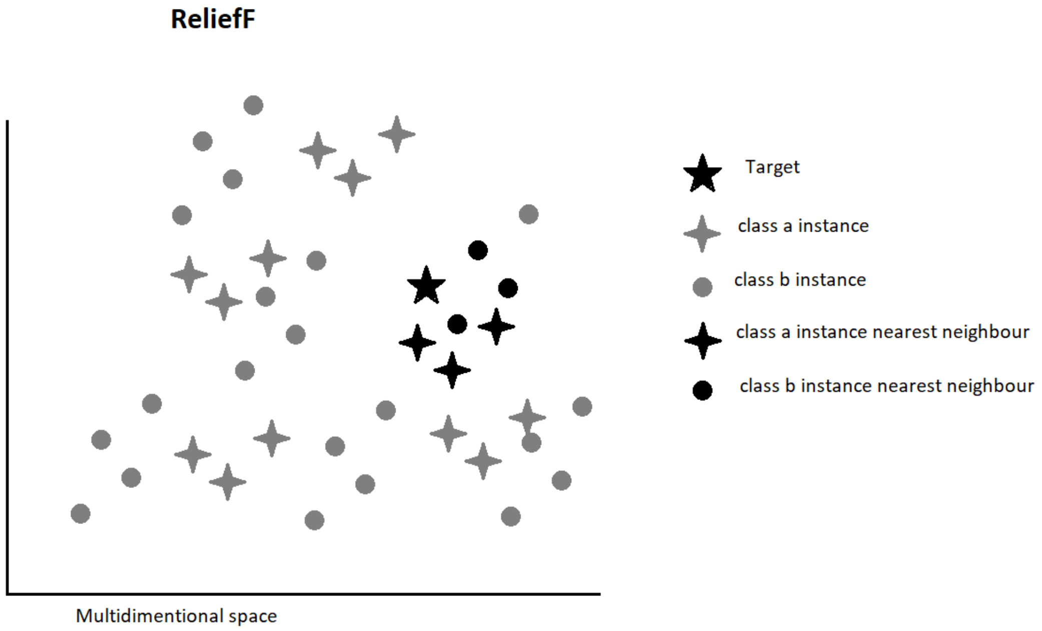

ReliefF
Target
class a instance
class b instance
class a instance nearest neighbour
class b instance nearest neighbour
Multidimentional space

Figure 3-6 ReliefF number of neighbours, the target is the average distance of all pairs of the training data, and we are looking for the nearest neighbours from the target, Modified from (R. J. Urbanowicz, Olson, et al., 2018).

3.7.2.2 MultiSURF

MultiSURF is another multivariate selection method that has been tested by Langberg in his thesis (Langberg, 2019). It is an extension of the ReliefF algorithm, and the advantage of using it instead of ReliefF is that it can automatically determine the ideal value of the number of neighbours (*Using Skrebate - Scikit-Rebate*, n.d.). The classification scores were good in Langberg's thesis and also in the early testing of Biorad. Nevertheless, as mentioned earlier, one of the criteria for choosing algorithms in Biorad was the execution speed, and the MultiSURF is a very slow algorithm for a large number of features. That is why the MultiSURF was not added to Biorad.

3.7.3 Variance Threshold

Variance threshold selector removes features with variance below a threshold value (*VarianceThreshold - Scikit-Learn*, n.d.). For that reason, it is crucial to avoid scaling the features before using this method. In Biorad, for variance threshold, the scaling of the features is done after the feature selection. The threshold value used for data selection is the only hyperparameter to tune in Biorad.

The variance threshold selector used in Biorad is part of the scikit-learn Python package.

3.8 Classifications

In Biorad, six different binary classifications are used: Ridge, Light Gradient Boosting Machine (LightGBM), Support Vector Classification (SVC), Decision Tree, Logistic Regression, and Extra Tree. All of the classifier implementations used are from skit-learn, except the LightGBM, which is provided by LightGBM python package (*LightGBM Documentation*, n.d.). These classifiers were used in the first version of Biorad, and they performed well with the radiomics data (Langberg, 2019).

3.8.1 Ridge regression

Ridge classifier treats the classification problem as a regression after converting the target values into -1 and 1.

In the Biorad application, the alpha parameter is used for the regularization, which is used to reduce the variance and control the overfitting. The bigger the alpha value, the stronger is the regularization. The type of regularization in ridge regression is L2 (Raschka & Mirjalili, 2017).

The ridge classifier used in Biorad is part of the scikit-learn Python package.

3.8.2 Light gradient boosting machine (LightGBM)

Light gradient boosting machine (LightGBM) is a tree-based learning algorithm. This algorithm was selected to be used in Biorad because of its fast training speed and low memory usage. It also supports parallelisation, and it is capable of handling large-scale data (Mandot, 2018).

The hyperparameters used in Biorad for tuning the model are:

- max_depth: limit the maximum depth of the tree model; smaller values can help to deal with overfitting.
- num_leaves: limit the maximum number of leaves in each single tree.
- min_child_samples: also known as minimum data in leaf, and it helps to deal with overfitting.

LightGBM is available via a free Python package called lightgbm (*LightGBM Documentation*, n.d.).

3.8.3 C-Support Vector Classification

SVC is known as C-Support Vector Classification. This classifier is not practical for a large number of samples, because the training time exhibits quadratic growth with the number of samples (*Sklearn.Svm.SVC — Scikit-Learn 0.23.1 Documentation*, n.d.). Nevertheless, the number of samples usually is not very large in radiomics which makes this classifier practical.

The regularization parameter used in Biorad for this classifier is the "C". The strength of the regularization and the value of "C" is inversely proportional; the type of regularization is L2 (*Sklearn.Svm.SVC — Scikit-Learn 0.23.1 Documentation*, n.d.). The kernel used in Biorad is the default in the classifier, which is 'rbf'.

The SVC classifier used in Biorad is part of the scikit-learn Python package.

3.8.4 Decision Tree

Decision tree is a supervised machine learning method that infers a set of decisions by portioning the features. They usually tend to overfit when the dataset has many features, like in the case of radiomics data (*Decision Trees - Scikit-Learn*, n.d.).

Two regularization parameters are used in Biorad to tune the decision tree classifier:

- min_samples_leaf: If the number of training samples in either the left or right of the leaf is not greater than or equal to the min_samples_leaf value, then the split will not be considered.
- max_depth: The maximum depth of the tree, the default value is None, where the nodes of the tree are expanded until the number of samples per leaf is less than the min_samples_split value, or until all leaves are pure.

The decision tree classifier used in Biorad is part of the scikit-learn Python package.

3.8.5 Logistic Regression

Logistic regression is a linear classifier; it assigns probabilities for each class. For regularization the parameter 'C' was used, and it is similar to the same parameter in SVC, it is inversely proportional to the regularization strength(*Sklearn.Linear_model.LogisticRegression — Scikit-Learn 0.23.1 Documentation*, n.d.). The regularization used in Biorad is the classifier default value which is 'L2'.

The logistic regression classifier used in Biorad is part of the scikit-learn Python package.

3.8.6 Extra Tree classifier

Extra tree classifier, also known as Extremely randomized trees is a tree-based classifier. It is an ensemble classifier that fits multiples of randomized decision trees on different subsets of the training data. The classifier uses the average of all the trees which helps control the overfitting and improves the results (Geurts et al., 2006).

For regularization, min_samples_leaf parameter is used, and it is similar to the same parameter in the decision tree classifier. It restricts the splits of the leaves which helps to control the overfitting.

The extra tree classifier used in Biorad is part of the scikit-learn Python package.

3.9 *t*-test for difference of means between two samples

To check if the two results from the Biorad are statistically significantly different, we used the *t*-test. In this test, we assume a hypothesis about the distribution of the variables in the population, then we either accept or reject this hypothesis with a certain probability of error (Sá, 2007). In Biorad, our hypothesis would be that there is no difference between two scores from the selectors/classifiers cross-validation, then based on the probability of error we either accept or reject this hypothesis.

The calculation of the *t*-test was done using an online calculation tool from GraphPad (*GraphPad QuickCalcs: T Test Calculator*, n.d.).

4 Biorad Application

The biorad application was developed using the *Python*^{TM} programming language. The code is available at https://github.com/ahmedalbuni/biorad.

Biorad consists of two different modules, the feature extraction, and the feature selection and classification module. These two modules are entirely independent of each other, and the user can run each one separately. The radiomics features extracted from the feature extraction module can be analysed using other applications, and the features selection and classification module can be used to analyse any binary or multiclass classification problem, not just radiomics data.

The installation guide of the software is available in Appendix B: Biorad installations and use instructions.

4.1 Features extraction module

The feature extraction module is used to generate radiomics features for medical and non-medical images. The user can select the group of radiomics feature needed for the analysis. This module provides a command-line interface for the users, which makes the feature extraction possible without programming.

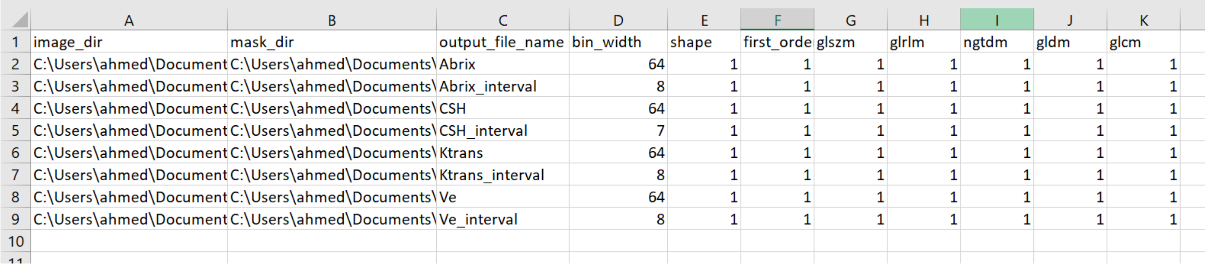
4.1.1 Input and configurations

The command-line interface of the feature extraction module requires a specific CSV file format as shown in Figure 4-1. Sample extraction module parameters for the CSV input file

This CSV file should have the following fields:

- image_dir: This should contain the paths of the images.
- mask_dir: This should provide the paths of the images' masks, make sure the mask names match the names of the corresponding images. If there is more than one mask for each image, then a new row for each mask should be inserted in this CSV file.
- output_file_name: The desired name of the output file. If the path is not included with the file name, then the files will be created at the current working directory in the command line window.
- bin_width: The default bin width in pyradiomics in 25, each bin represents specific greyscale intensity values, for demonstration of the effect of different bin widths on the results check Figure 2-5 and Figure 2-6. The user can use a different value if required.
- shape: if it has a value of '1', then the shape features will be generated. These features will depend on the image dimensions. 2D Shape features will be extracted for 2-dimensional images, and 3D shape features for 3-dimensional images.
- first_order: if it has a value of '1', first-order features will be extracted.
- glszm: if it has a value of '1', the grey level size zone matrix features will be extracted.
- glrlm: if it has a value of '1', grey level run length matrix features will be extracted.

- ngtdm: if it has a value of '1', neighbouring grey tone difference matrix features will be extracted.
- gldm: if it has a value of '1', grey level dependence matrix features will be extracted.
- glcm: if it has a value of '1', grey level cooccurrence matrix features will be extracted.



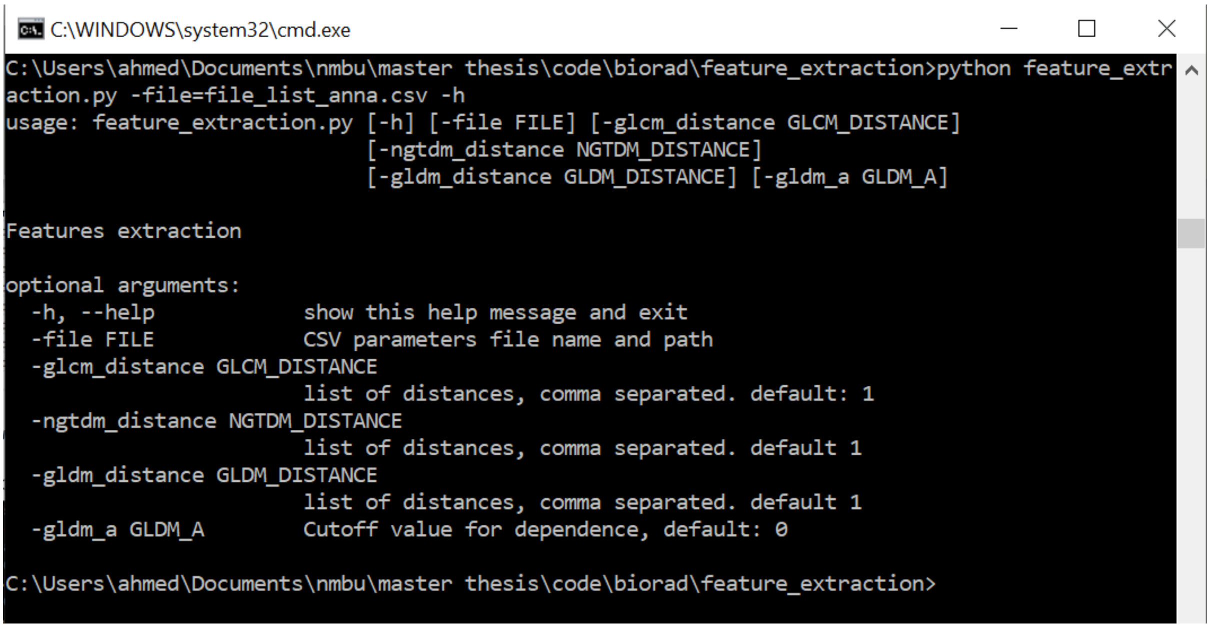
	A	B	C	D	E	F	G	H	I	J	K
1	image_dir	mask_dir	output_file_name	bin_width	shape	first_orde	glszm	glrlm	ngtdm	gldm	glcm
2	C:\Users\ahmed\Document	C:\Users\ahmed\Documents\	Abrix	64	1	1	1	1	1	1	1
3	C:\Users\ahmed\Document	C:\Users\ahmed\Documents\	Abrix_interval	8	1	1	1	1	1	1	1
4	C:\Users\ahmed\Document	C:\Users\ahmed\Documents\	CSH	64	1	1	1	1	1	1	1
5	C:\Users\ahmed\Document	C:\Users\ahmed\Documents\	CSH_interval	7	1	1	1	1	1	1	1
6	C:\Users\ahmed\Document	C:\Users\ahmed\Documents\	Ktrans	64	1	1	1	1	1	1	1
7	C:\Users\ahmed\Document	C:\Users\ahmed\Documents\	Ktrans_interval	8	1	1	1	1	1	1	1
8	C:\Users\ahmed\Document	C:\Users\ahmed\Documents\	Ve	64	1	1	1	1	1	1	1
9	C:\Users\ahmed\Document	C:\Users\ahmed\Documents\	Ve_interval	8	1	1	1	1	1	1	1
10											

Figure 4-1. Sample extraction module parameters for the CSV input file.

The features selection tool has additional parameters, as shown in Figure 4-2.

These parameters are:

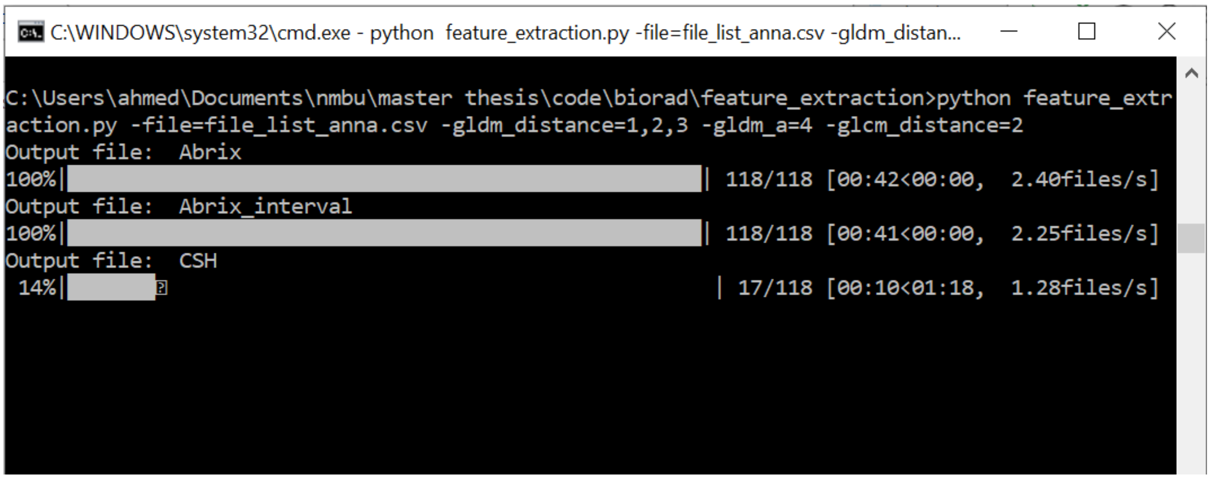
- glcm_distance: This specifies the distances between the centre voxel and the neighbours used for GLCM features generation. The list should be provided with a comma-separated list, without spaces. More on GLCM table in 2.4.3 above.
- ngtdm_distance: This specifies the distances between the centre voxel and the neighbours used for NGTDM features generation. The list should be provided with a comma-separated list, without spaces.
- gldm_distance: This specifies the distances between the centre voxel and the neighbours used for GLDM features generation. The list should be provided with a comma-separated list, without spaces.
- gldm_a: An integer value, α cut-off value for dependence. A neighbouring voxel with grey level *j* is considered dependent on centre voxel with grey level *i* if |*i*–*j*|≤α (*Radiomic Features - Pyradiomics*, n.d.).



```
C:\Users\ahmed\Documents\nmbu\master thesis\code\biorad\feature_extraction>python feature_extr
action.py -file=file_list_anna.csv -h
usage: feature_extraction.py [-h] [-file FILE] [-glcm_distance GLCM_DISTANCE]
                             [-ngtdm_distance NGTDM_DISTANCE]
                             [-gldm_distance GLDM_DISTANCE] [-gldm_a GLDM_A]

Features extraction

optional arguments:
  -h, --help            show this help message and exit
  -file FILE            CSV parameters file name and path
  -glcm_distance GLCM_DISTANCE
                        list of distances, comma separated. default: 1
  -ngtdm_distance NGTDM_DISTANCE
                        list of distances, comma separated. default 1
  -gldm_distance GLDM_DISTANCE
                        list of distances, comma separated. default 1
  -gldm_a GLDM_A        Cutoff value for dependence, default: 0

C:\Users\ahmed\Documents\nmbu\master thesis\code\biorad\feature_extraction>
```

Figure 4-2. Additional parameters to be specified for the feature selection tool.

Once the user enters all the required parameters, the progress screen will look like the one in Figure 4-3.



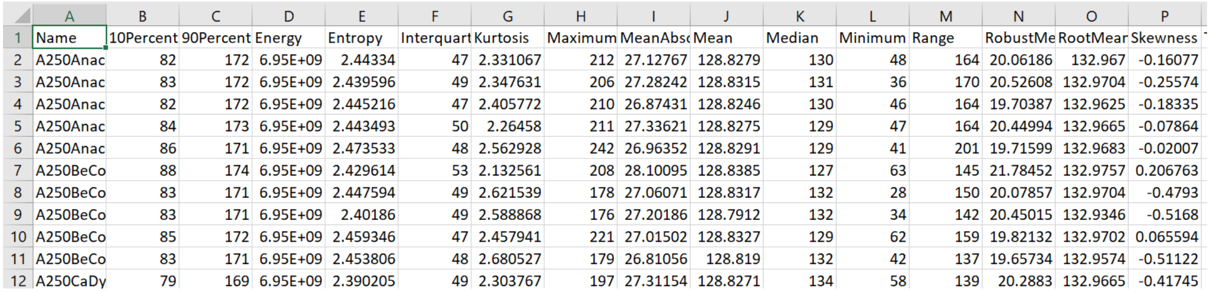
```
C:\WINDOWS\system32\cmd.exe - python  feature_extraction.py -file=file_list_anna.csv -gldm_distan...

C:\Users\ahmed\Documents\nmbu\master thesis\code\biorad\feature_extraction>python feature_extr
action.py -file=file_list_anna.csv -gldm_distance=1,2,3 -gldm_a=4 -glcm_distance=2
Output file:  Abrix
100%|██████████████████████████████████████████████████| 118/118 [00:42<00:00,  2.40files/s]
Output file:  Abrix_interval
100%|██████████████████████████████████████████████████| 118/118 [00:41<00:00,  2.25files/s]
Output file:  CSH
 14%|███████▯                                           | 17/118 [00:10<01:18,  1.28files/s]
```

Figure 4-3. A screenshot of the command-line interface (CLI) of the feature extraction tool.

4.1.2 The output

The feature extraction module generates as output, CSV files that contain the name of the images, along with the features.

For each folder provided, the tool will generate a CSV file that contains the file names along with the selected groups of pyradiomics features, like the sample output in Figure 4-4.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	Name	10Percent	90Percent	Energy	Entropy	Interquart	Kurtosis	Maximum	MeanAbs	Mean	Median	Minimum	Range	RobustMe	RootMear	Skewness
2	A250Anac	82	172	6.95E+09	2.44334	47	2.331067	212	27.12767	128.8279	130	48	164	20.06186	132.967	-0.16077
3	A250Anac	83	172	6.95E+09	2.439596	49	2.347631	206	27.28242	128.8315	131	36	170	20.52608	132.9704	-0.25574
4	A250Anac	82	172	6.95E+09	2.445216	47	2.405772	210	26.87431	128.8246	130	46	164	19.70387	132.9625	-0.18335
5	A250Anac	84	173	6.95E+09	2.443493	50	2.26458	211	27.33621	128.8275	129	47	164	20.44994	132.9665	-0.07864
6	A250Anac	86	171	6.95E+09	2.473533	48	2.562928	242	26.96352	128.8291	129	41	201	19.71599	132.9683	-0.02007
7	A250BeCo	88	174	6.95E+09	2.429614	53	2.132561	208	28.10095	128.8385	127	63	145	21.78452	132.9757	0.206763
8	A250BeCo	83	171	6.95E+09	2.447594	49	2.621539	178	27.06071	128.8317	132	28	150	20.07857	132.9704	-0.4793
9	A250BeCo	83	171	6.95E+09	2.40186	49	2.588868	176	27.20186	128.7912	132	34	142	20.45015	132.9346	-0.5168
10	A250BeCo	85	172	6.95E+09	2.459346	47	2.457941	221	27.01502	128.8327	129	62	159	19.82132	132.9702	0.065594
11	A250BeCo	83	171	6.95E+09	2.453806	48	2.680527	179	26.81056	128.819	132	42	137	19.65734	132.9574	-0.51122
12	A250CaDy	79	169	6.95E+09	2.390205	49	2.303767	197	27.31154	128.8271	134	58	139	20.2883	132.9665	-0.41745

Figure 4-4: Sample feature extraction output CSV file generated by the feature extraction tool.

4.2 Feature selection and classification module

The feature selection and classification module is one of the two modules of the Biorad application. Optimal parameters will be selected using Randomized Search CV that was discussed in 3.2 above.

4.2.1 Data Scaling

Many machine learning algorithms required the data to be standardized, and they might misbehave if the data does not look like a standard distribution. Scikit-learn standard scaler standardizes the features by removing the mean and scales them by the unit variance (*StandardScaler - Scikit-Learn*, n.d.).

In Biorad application, the StandardScaler from scikit-learn is used to scale the data. The data are scaled before the feature selection except for Variance Threshold because this algorithm is based on the variance of the data which will be lost if the scaling is done before.

4.2.2 Scoring

The Biorad application supports different scoring metrics. For binary classification the following are supported (Raschka & Mirjalili, 2017):

- roc_auc
- accuracy
- f1-score
- precision
- recall

For multiclass classification the following are supported (Raschka & Mirjalili, 2017):

- accuracy
- f1_micro
- f1_macro
- f1_weighted
- precision_micro
- precision_macro
- precision_weighted
- recall_micro
- recall_macro
- recall_weighted

By using the confusion matrix illustrated in Table 4-1 Confusion matrix, we can calculate some of the different scoring used in the Biorad feature selection and classification module.

Table 4-1 Confusion matrix

Confusion matrix	Predicted False	Predicted True
Actual value False	True Negative (TN)	False Positive (FP)
Actual value True	False Negative (FN)	True Positive (TP)

The following shows the calculation of some scoring metrics using the confusion matrix:

- Accuracy = (TP+TN)/total predictions
- Precision = TP/(FP+TP)
- Recall = TP/(FN+TP)
- F1 = 2 x (Precision * Recall)/(Precision + Recall)

The micro average for multiclass classification is calculated from individual confusion matrixes, while the macro average is calculated as the average of the different systems. The micro average is used when the user wants to evaluate each prediction equally, and the macro average is used to weight all classes equally to get the overall performance (Raschka & Mirjalili, 2017).

For imbalanced datasets, the accuracy usually is not the best choice for scoring the classification model, the best scoring depends on what we care about in the classification problem, for example, if our goal is to identify most of the malignant cancer patients, then the recall should be used. However, if we are identifying spam emails, and we do not want to label a genuine email as spam by mistake, then the precision would be more suitable in this case. F1 score also is good to deal with imbalanced data, and it is a combination of precision and recall (Raschka & Mirjalili, 2017).

4.2.3 Parallelisation

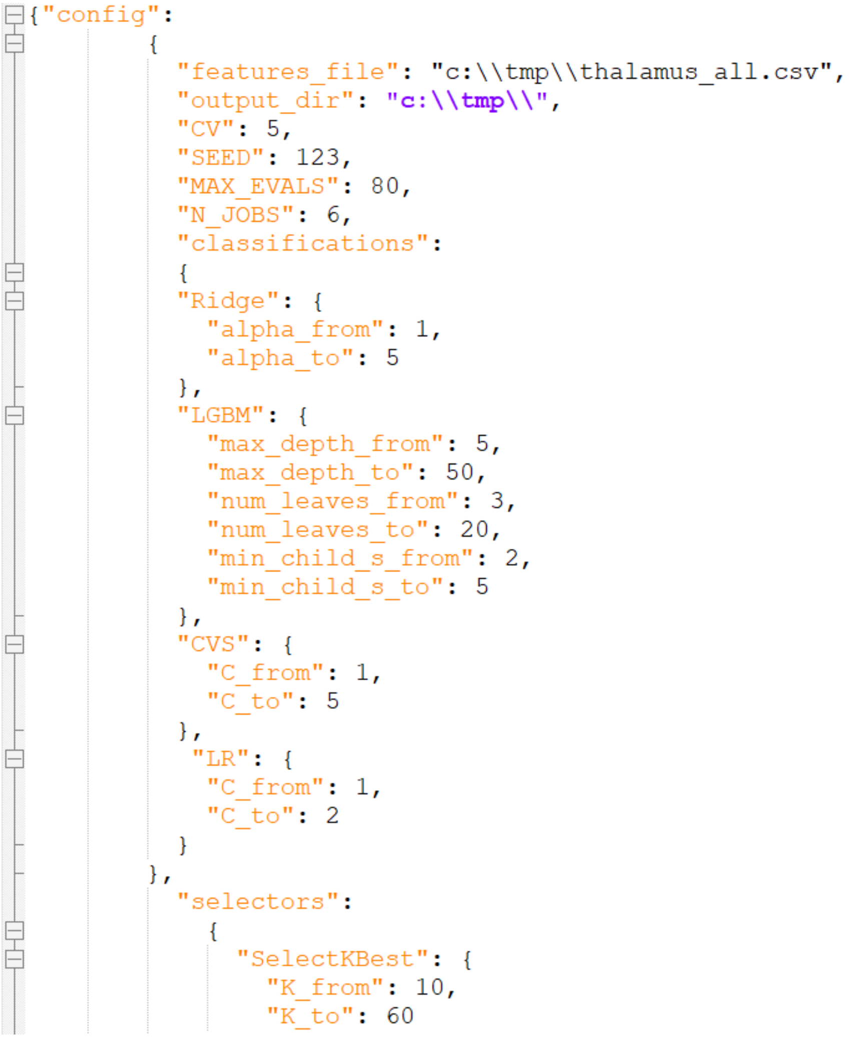
Parallelisation is a part of the Scikit-learn implementation of the RandomizedSearchCV used in the Biorad tool. It is possible to choose the number of jobs in the configuration JSON file shown in Figure 4-5. However, by utilising the power of parallelisation, we lose the producibility of the results because the order of running the various jobs cannot be guaranteed, which means every time we run the same configuration file the results can differ even though we are using the same seed number. So, if the producibility is essential, then the user should opt off this feature.

4.2.4 Settings

The settings to be provided in order to use the tool should be in a specific JSON format. A sample configuration JSON file is provided in the root directory of the Biorad application on GitHub (config.json). This file is to be validated using a JSON schema file for errors before being processed. A snippet from the sample JSON file is shown in Figure 4-5.

The JSON file consists of several parts:

- The General configurations
- Feature selectors configurations
- Classifiers configurations



```
{"config":
        {
          "features_file": "c:\\tmp\\thalamus_all.csv",
          "output_dir": "c:\\tmp\\",
          "CV": 5,
          "SEED": 123,
          "MAX_EVALS": 80,
          "N_JOBS": 6,
          "classifications":
          {
          "Ridge": {
            "alpha_from": 1,
            "alpha_to": 5
          },
          "LGBM": {
            "max_depth_from": 5,
            "max_depth_to": 50,
            "num_leaves_from": 3,
            "num_leaves_to": 20,
            "min_child_s_from": 2,
            "min_child_s_to": 5
          },
          "CVS": {
            "C_from": 1,
            "C_to": 5
          },
           "LR": {
            "C_from": 1,
            "C_to": 2
          }
        },
          "selectors":
            {
              "SelectKBest": {
                "K_from": 10,
                "K_to": 60
```

Figure 4-5: A snippet of a sample configuration JSON file.

4.2.4.1 General configurations

The general configurations such as the number of CV folds, seed number, parallelisation, the number of iterations to try out the hyperparameters combinations, and the dataset, the user will need to provide a JSON file with the parameters are specified in Table 4-2. There are no default values for these parameters, but a sample configuration file with sample parameter values is available within the application.

Table 4-2 General configurations for Biorad feature selection and classification module.

CV	Integer, the number of cross-validation folds should be greater than 2.
SEED	Integer: The random seed number – used for reproducing the results.
N_JOBS	Parallelisation 1: No parallelisation – choose this for reproducibility -1: Use all available cores.

	Other positive integers – max (number of available CPU cores, provided number) will be used.
MAX_EVALS	Integer: Maximum number of parameter settings for both classifiers and features selectors together, to be tried out. Choose a higher number for better accuracy, and a lower number for faster processing.
features_file	The path of the input CSV file the contains the dataset along with the response variable as the last field.
output_dir	The directory to store the output files

4.2.4.2 Feature selector configurations

The configurations related to the hyperparameters for the feature selectors are described in Table 4-3. Same as the general configuration, there are no default values for these parameters.

Table 4-3 The hyperparameters configurations for the feature selectors in Biorad.

ReliefF	
n_neighbors_from	The number of neighbours to consider when assigning feature importance scores.
n_neighbors_to	Integer, the maximum number of neighbours to consider.
n_features_to_select_from	Integer, the minimum number of features to select.
n_features_to_select_to	Integer, the maximum number of features to select.
VarianceThreshold	
threshold_from	Features with variance less than this value will be removed.
threshold_to	The maximum threshold value to consider.
mutual_info	
param_from	Integer, the minimum number of features to select.
param_from	Integer, the maximum number of features to select.
fisher_score	
param_from	Integer, the minimum number of features to select
param_from	Integer, the maximum number of features to select

4.2.4.3 Classifier configurations

The configurations related to the hyperparameters for the feature selectors are described in Table 4-4. Same as the general configuration, there are no default values for these parameters.

Table 4-4 The hyperparameters configurations for the classifiers in Biorad

Ridge	
alpha_from	Regularisation strength. Should be a positive float value.
alpha_to	Maximum Alpha value to consider.
LGBM	
max_depth_from	Integer, the depth of the tree model start value, to deal with overfitting
max_depth_to	Integer, the maximum depth of the tree model.
num_leaves_from	Integer, 1 < num_leaves <= 131072
num_leaves_to	Integer, 1 < num_leaves <= 131072
min_child_s_from	Integer, > 0, Minimum child samples start value, also called min_data_in_leaf.
min_child_s_to	Integer, > 0, Minimum child samples end value, also called min_data_in_leaf.
SVC	
C_from	Positive float value. It is the inverse of regularisation strength.
C_to	The maximum C value for the regularisation.
LR	
C_from	Positive float value. It is the inverse of regularisation strength.
C_to	The maximum C value for the regularisation.

4.2.5 The output

The Biorad feature selection and classification module generates several output files, one of them is the heatmap of the cross-validation scores of all the Biorad feature selectors and classifiers, an example of which is shown in Figure 4-6. The heatmap data will be stored in a CSV file to make it easier for further analysis as shown in Figure 4-7.

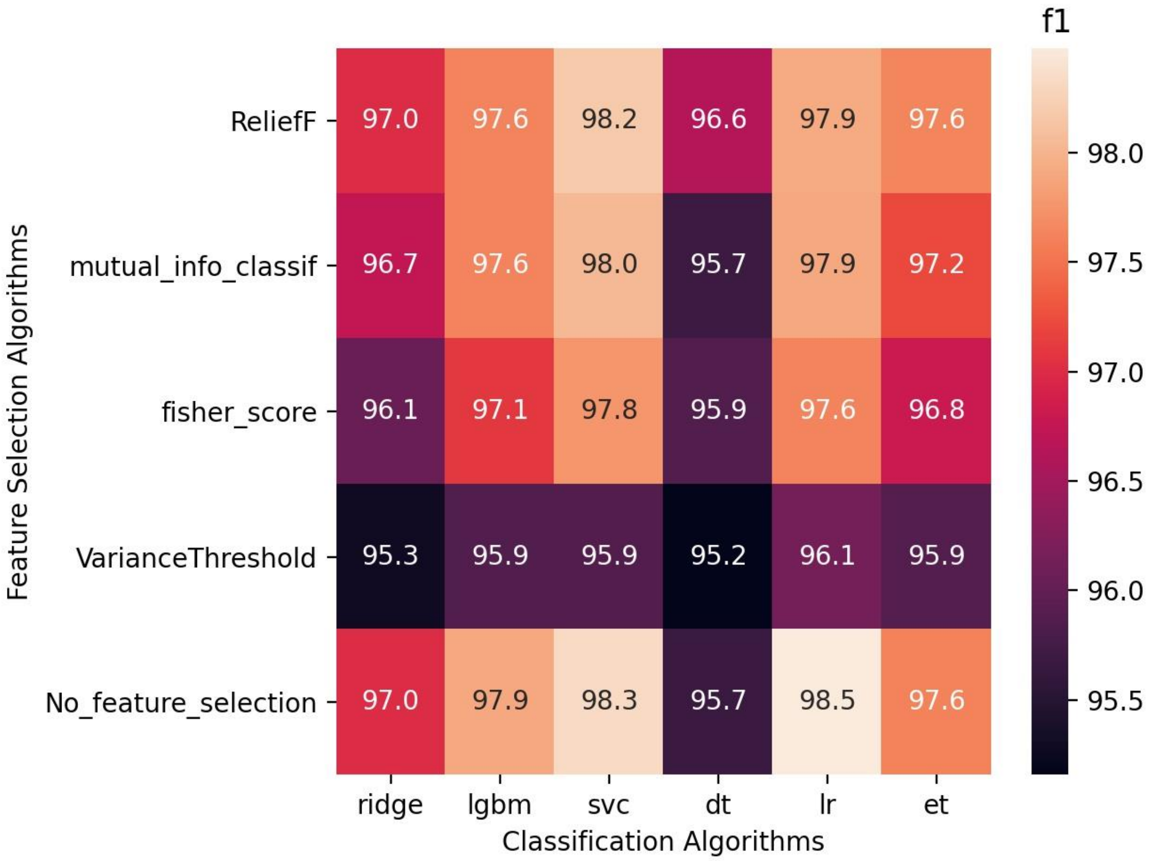
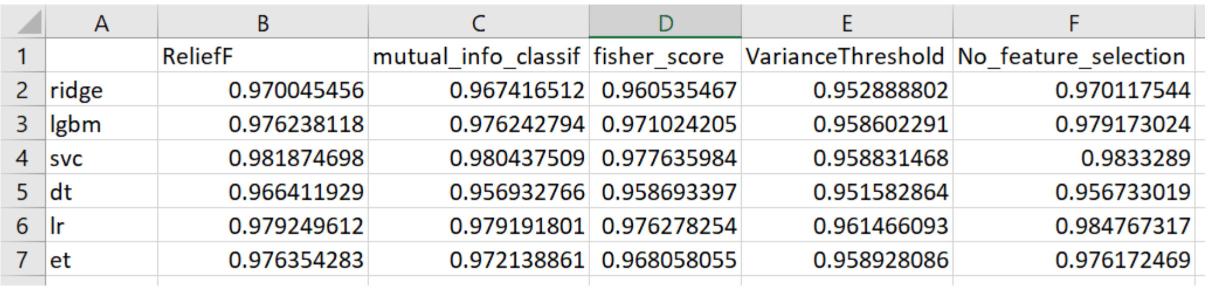


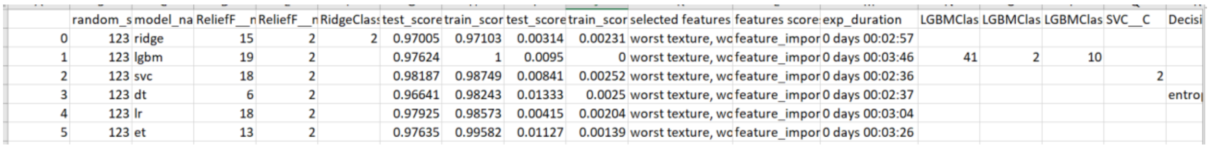
Figure 4-6 An example of a heatmap of the cross-validation for the Scikit-learn breast cancer dataset.



	A	B	C	D	E	F
1		ReliefF	mutual_info_classif	fisher_score	VarianceThreshold	No_feature_selection
2	ridge	0.970045456	0.967416512	0.960535467	0.952888802	0.970117544
3	lgbm	0.976238118	0.976242794	0.971024205	0.958602291	0.979173024
4	svc	0.981874698	0.980437509	0.977635984	0.958831468	0.9833289
5	dt	0.966411929	0.956932766	0.958693397	0.951582864	0.956733019
6	lr	0.979249612	0.979191801	0.976278254	0.961466093	0.984767317
7	et	0.976354283	0.972138861	0.968058055	0.958928086	0.976172469

Figure 4-7 The cross-analysis scores for the breast cancer dataset.

Also, one CSV file per feature selector is generated by this module. The CSV file includes the optimal hyperparameters selected, train and test scores, the standard deviation of train and test scores, time elapsed in each test, the features selected and the features scores given by the feature selector algorithm, see Figure 4-8. One additional CSV file will be created for running the classifiers without feature selection.



	random_s	model_na	ReliefF__n	ReliefF__n	RidgeClass	test_score	train_scor	test_score	train_scor	selected features	features score	exp_duration	LGBMClas	LGBMClas	LGBMClas	SVC__C	Decisi
0	123	ridge	15	2	2	0.97005	0.97103	0.00314	0.00231	worst texture, wc	feature_impor	0 days 00:02:57					
1	123	lgbm	19	2		0.97624	1	0.0095	0	worst texture, wc	feature_impor	0 days 00:03:46	41	2	10		
2	123	svc	18	2		0.98187	0.98749	0.00841	0.00252	worst texture, wc	feature_impor	0 days 00:02:36				2	
3	123	dt	6	2		0.96641	0.98243	0.01333	0.0025	worst texture, wc	feature_impor	0 days 00:02:37					entro
4	123	lr	18	2		0.97925	0.98573	0.00415	0.00204	worst texture, wc	feature_impor	0 days 00:03:04					
5	123	et	13	2		0.97635	0.99582	0.01127	0.00139	worst texture, wc	feature_impor	0 days 00:03:26					

Figure 4-8 The CSV file that contains the details of experiments run with the ReliefF feature selector.

Another output is the features frequency file that displays how many times each feature got selected by the different feature selectors across all the cross-validation tests. An example is shown in Table 4-5.

Table 4-5 Most selected features using Biorad from the scikit-learn breast cancer dataset.

Features	**Selection count**	**Frequency**
worst texture	24	100%
worst radius	24	100%
worst area	23	96%
mean perimeter	23	96%
perimeter error	21	88%
area error	20	83%
worst concave points	18	75%
worst perimeter	18	75%
mean concave points	18	75%
mean radius	17	71%
mean area	17	71%
mean concavity	17	71%
worst concavity	16	67%
radius error	16	67%
mean texture	12	50%

As we are running 24 different tests with feature selectors, which is four different features selector multiplied by six different classifiers, then the selection count of 24 means that this feature was selected in all the tests.

And finally, the log file which includes the start time, end time and the JSON file used for configurations.

5 Results and discussions

Multiple tests were performed to test the Biorad application in both extracting the radiomics features and the cross-analysis between the feature selectors and the classifiers. In this chapter, we discuss those results and compare some of them to other studies that were done using the same data.

5.1 Extracting radiomics features

Tests in chapters 5.2 and 5.3 were done using the head and neck dataset. Overview of the dataset is available in Appendix A.1 Head and neck cancer patients' dataset.

The radiomics features were extracted from the head and neck cancer dataset using the feature extraction module in Biorad with default settings, bin width used for CT scan images was five and for PET scan images were 0.2. A total of 89 textural features from the PET scan images and 89 textural features from the CT scan images were extracted in addition to 14 shape features.

5.2 Selecting the optimal parameters

An experiment was done with both Grid search CV and Randomized Search CV, which are both used for parameter tuning by selecting the optimal hyperparameters for both feature selector and classifiers among a pool of provided ranges and discreet values. From the result of the experiment, it shows that RandomizedSearchCV can give very close results to the GridSearchCV, but it takes a much shorter time to do so, as shown in Figure 5-1 GridSearchCV results with 80 iterations, the time elapsed was 14.92 seconds. And in Figure 5-2 RandomizedSearchCV with 80 iterations results, the time elapsed 2.66 seconds.

According to Scikit-learn, the performance of RandomizedSearchCV might be slightly worse, but that is likely due to noise, and would not be carried to the test set (*Comparing Randomized Search and Grid Search - Scikit-Learn*, n.d.).

5.2.1 Grid Search CV

A simple test was performed with the head and neck cancer dataset mentioned in 3.1 above, including all the radiomics features from both CT/PET scan images and the clinical factors. The domain of the hyperparameters was:

- Ridge regression alpha: 1 to 20
- Fisher score number of features to select: 10 to 40

The total number of hyperparameters combinations to try in GridSearchCV was 20 multiplied by 31 + 20 where no feature selector is used, so the total size of the domain is 640. Two CV folds were used in the experiment. The results are shown in Figure 5-1, the total time taken was 14.92 seconds.

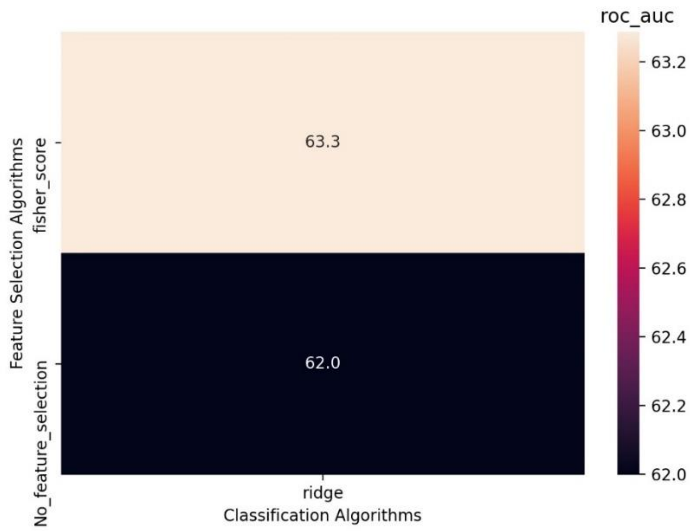


Figure 5-1 GridSearchCV results with 80 iterations, the time elapsed was 14.92 seconds.

5.2.2 Randomized Search CV

A similar experiment to the one in 5.2.1 above was conducted, but with RandomizedSearchCV, using '80' as the maximum number of interactions to try out hyperparameters configurations. Results are shown in Figure 5-2.

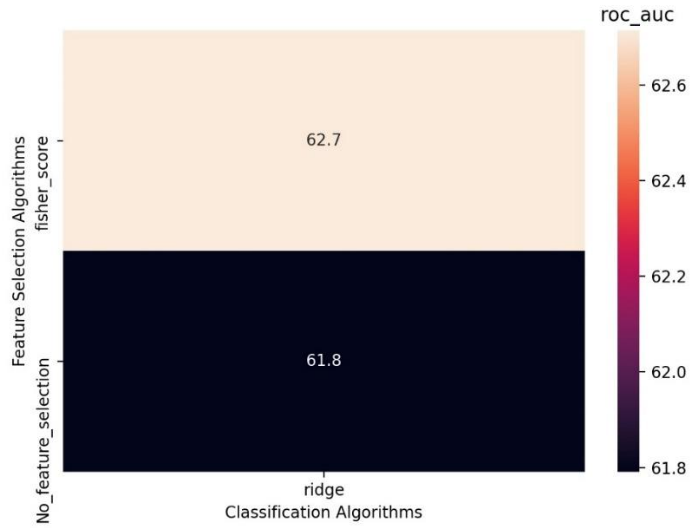


Figure 5-2 RandomizedSearchCV with 80 iterations results, the time elapsed 2.66 seconds.

5.2.3 MultiSURF

For testing the MultiSURF, it was added temporary to Biorad, and we ran the application for the head and neck dataset using all radiomics and clinical features. The number of iterations was 10, and the number of CV folds was 5. The ReliefF experiment took 46 seconds to complete (number of neighbours' range was from 1 to 3), while the MultiSURF took 11 minutes and 3 seconds, which means it is more than 14 times slower than the ReliefF.

On the other hand, the ReliefF results on average were only 0.04% better than MultiSURF, as shown in Figure 5-3.

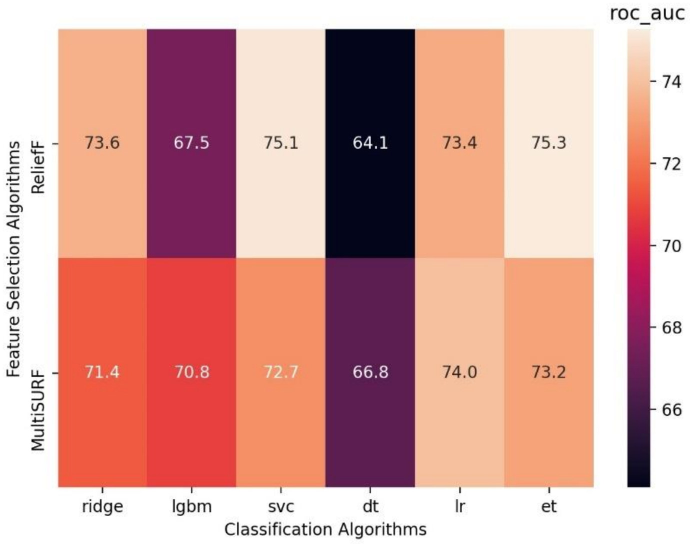


Figure 5-3 MultiSURF and ReliefF performance for the head and neck cancer dataset

5.3 Head and neck cancer dataset

In the first test, only clinical data in Appendix A.1 Head and neck cancer patients' dataset were used, the purpose of the test is to assess how the clinical data alone will perform in predicting the disease-free survival rate, compare the results to other studies on the same dataset and the added value of the radiomics features when they are added later. The maximum number of iterations to try different hyperparameter configurations was 80.

All settings used to run the test are stated in Table 5-1 for general settings, Table 5-2 for features selectors settings, and Table 5-3 for classifiers settings.

Table 5-1 General settings for the first test.

CV	5
SEED	123
N_JOBS	1

MAX_EVALS	80

Table 5-2 Feature selectors configurations for the first test.

ReliefF	
n_neighbors_from	1
n_neighbors_to	3
n_features_to_select_from	5
n_features_to_select_to	10
VarianceThreshold	
threshold_from	0.1
threshold_to	0.9
mutual_info	
param_from	5
param_from	10
fisher_score	
param_from	5
param_from	10

Table 5-3 Classifiers configurations for the first test.

Ridge	
alpha_from	1
alpha_to	5
LGBM	
max_depth_from	2
max_depth_to	10
num_leaves_from	3
num_leaves_to	20
min_child_s_from	2
min_child_s_to	5
SVC	
C_from	1
C_to	5
LR	
C_from	1
C_to	2

The test took 11 minutes and 20 seconds. The best result we got from this test was from the Ridge classifier with combination with ReliefF feature selection method 0.745±0.035, 0.035 is the standard error Logistic Regression and Extra Tree both gave good results also, 0.744±0.036 and 0.72±0.023, respectively. Cross-validation results are shown in Figure 5-4.

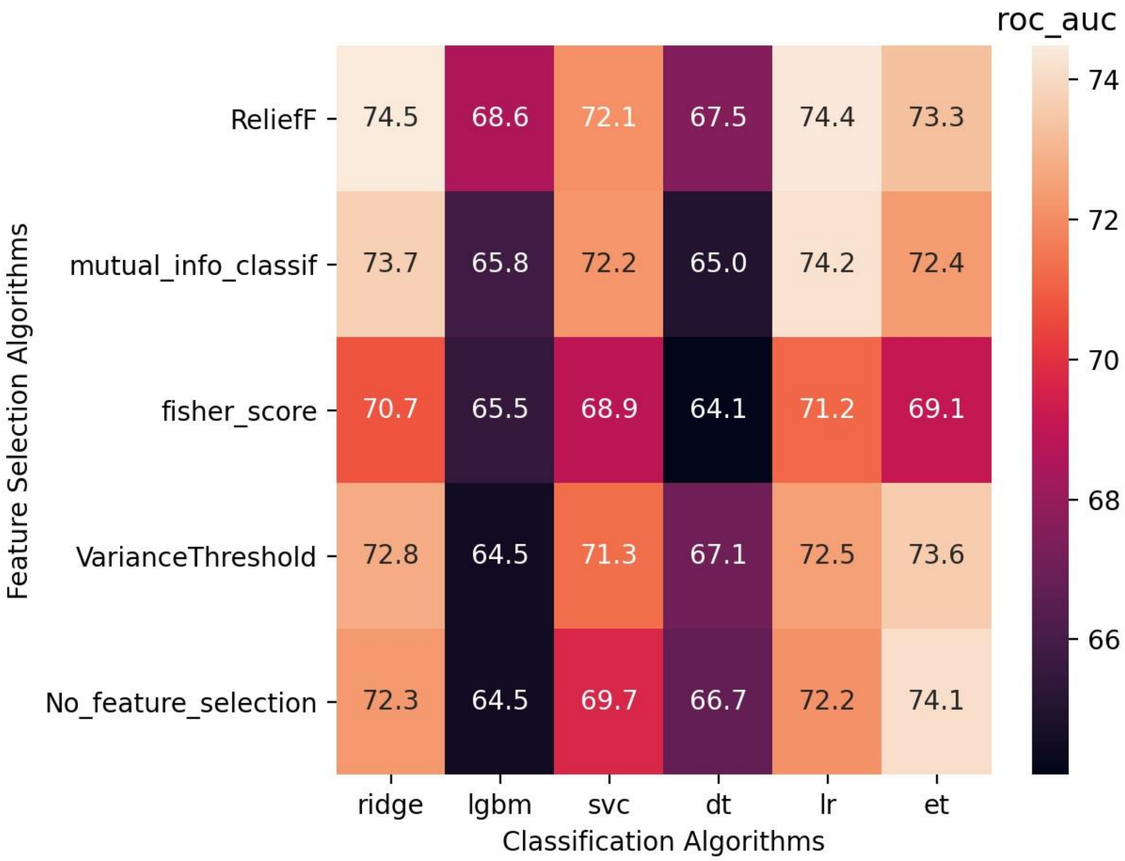


Figure 5-4 Heatmap for running the classification tool with 80 iterations, only clinical data from the head and neck cancer dataset were used to generate this graph.

Selected features in all the feature selectors and classifiers combinations are shown in Figure 5-5, A total of 24 experiments with features selection was conducted, and this table shows how many times each feature was selected among these experiments. The Pack Years Smoking features were selected by all feature selection methods in combination with all classifiers.

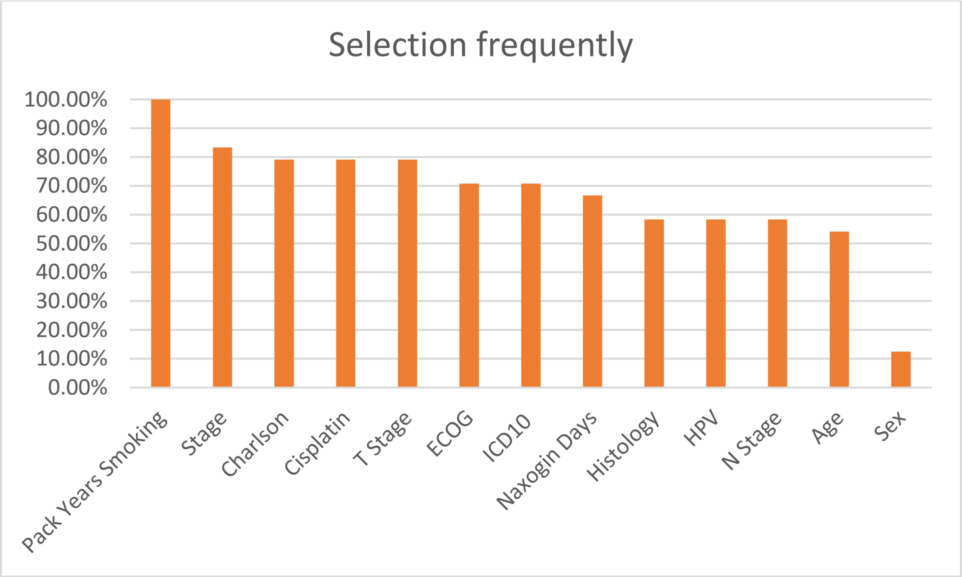


Figure 5-5 Most selected features in all classifiers and feature selectors in Biorad for the head and neck cancer dataset when using the clinical data only.

Table 5-4 shows the number of selected features, test scores, standard deviation between the five cross-validation folds and the standard error for the best score by algorithm. From Figure 5-6, we picked the least overlapping two values by the standard error, which are the mutual information and the fisher score and performed the *t*-test to check if they are statistically significant.

P-value and statistical significance:
The two-tailed P value equals 0.2920
By conventional criteria, this difference is considered to be not statistically significant.
Confidence interval:
The mean of Mutual Info minus Fisher Score equals 0.03000
95% confidence interval of this difference: From -0.03132 to 0.09132
Intermediate values used in calculations:
t = 1.1281
df = 8
standard error of difference = 0.027
This statistical calculation was done by GraphPad (*GraphPad QuickCalcs: T Test Calculator*, n.d.).

Table 5-4 Number of selected features, test scores, standard deviation and the standard error for the best score by algorithm.

Selection algorithm	**Number of selected features**	**Test scores**	**Test scores standard deviation**	**Standard error**
ReliefF	8	0.745	0.078	0.035
Mutual Information	9	0.742	0.040	0.018
Fisher Score	9	0.712	0.044	0.020
Variance Threshold	13	0.736	0.064	0.029

No feature selection	13	0.741	0.077	0.034

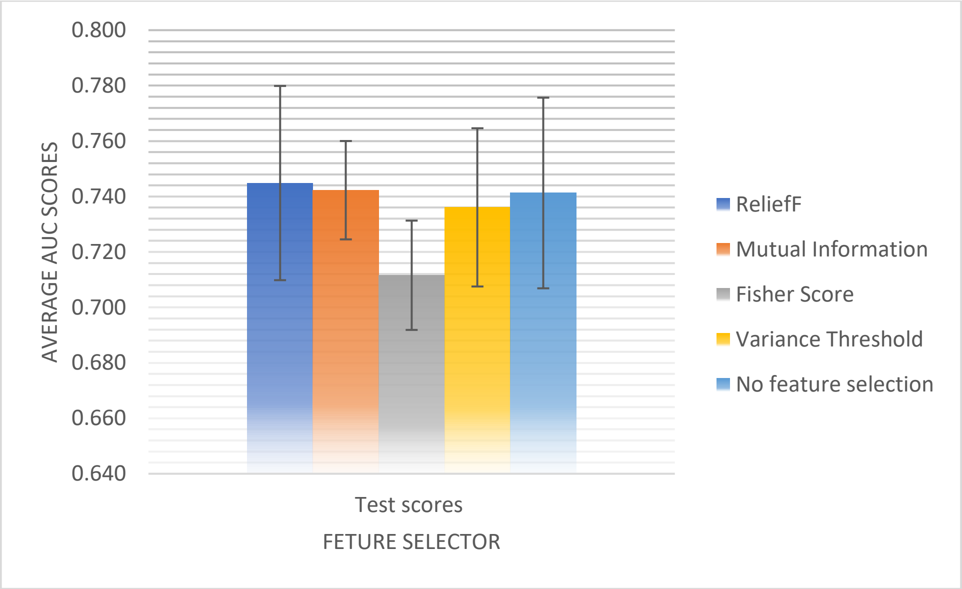


Figure 5-6 Feature selectors average scores with the standard error.

Figure 5-7 from Langberg thesis shows the results he got from the clinical factors only: In these results and in Biorad results, the Logistic regression and the ridge classifier performed better than the other classifiers. In Biorad, some of the tests scores were better than expected, and the variance was small in Mutual Information selector, they were better than the results obtained by Langberg experiments in Figure 5-7. But we do not have the standard deviation of Langberg's results so we could not do statistical tests to check the significance of the difference in these results.

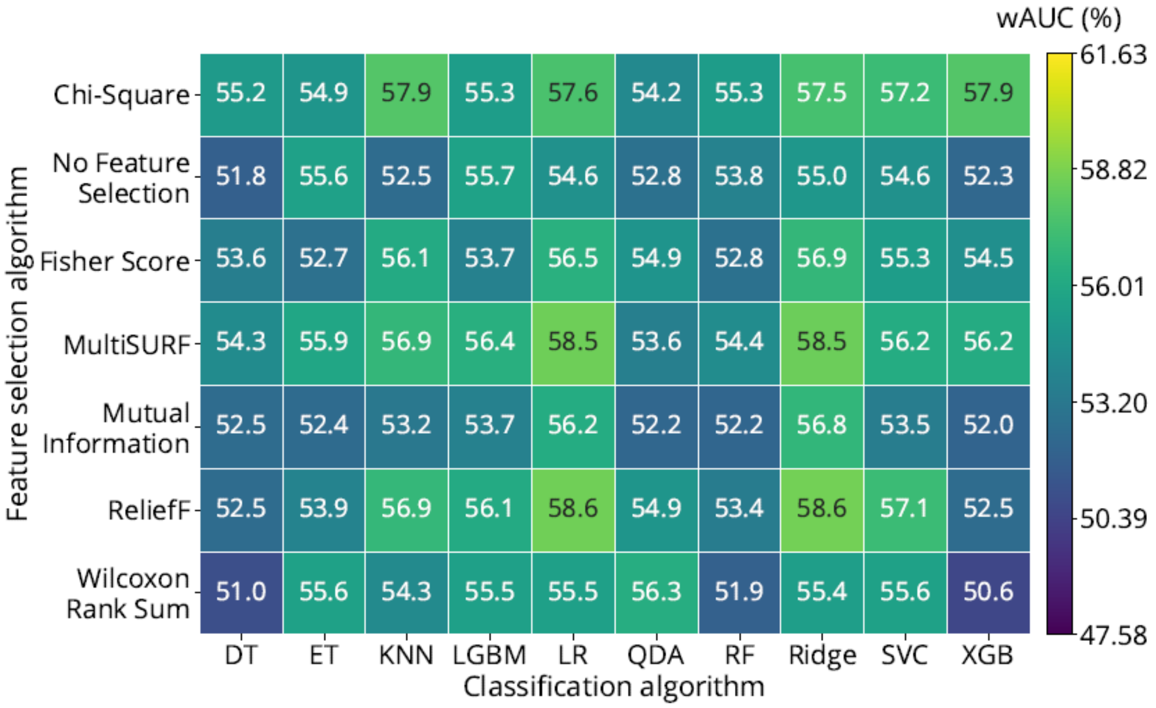


Figure 5-7 scores from the clinical factors only (Langberg, 2019), with permission.

The second test was done using the radiomics features from both CT scan images and PET scan images, without including the clinical data. The purpose of this test was to see if the radiomics features alone had enough information to describe the response variable (disease-free survival) and also to compare the results with the clinical data in the previous test. The configurations used were very similar, except for the number of features to select. Here we used 10 to 35 instead of 5 to 10, and the maximum depth of the lgbm tree used was 5 to 50 instead of 2 to 10, and the maximum depth for decision tree was 10, 20, 50 instead of 2, 5, 10. Those changes were necessary because of the size of the features; 192 features compared to only 13 features when the clinical data were used.

The test took 47 minutes and 24 seconds to complete, and the results are shown in Figure 5-8.

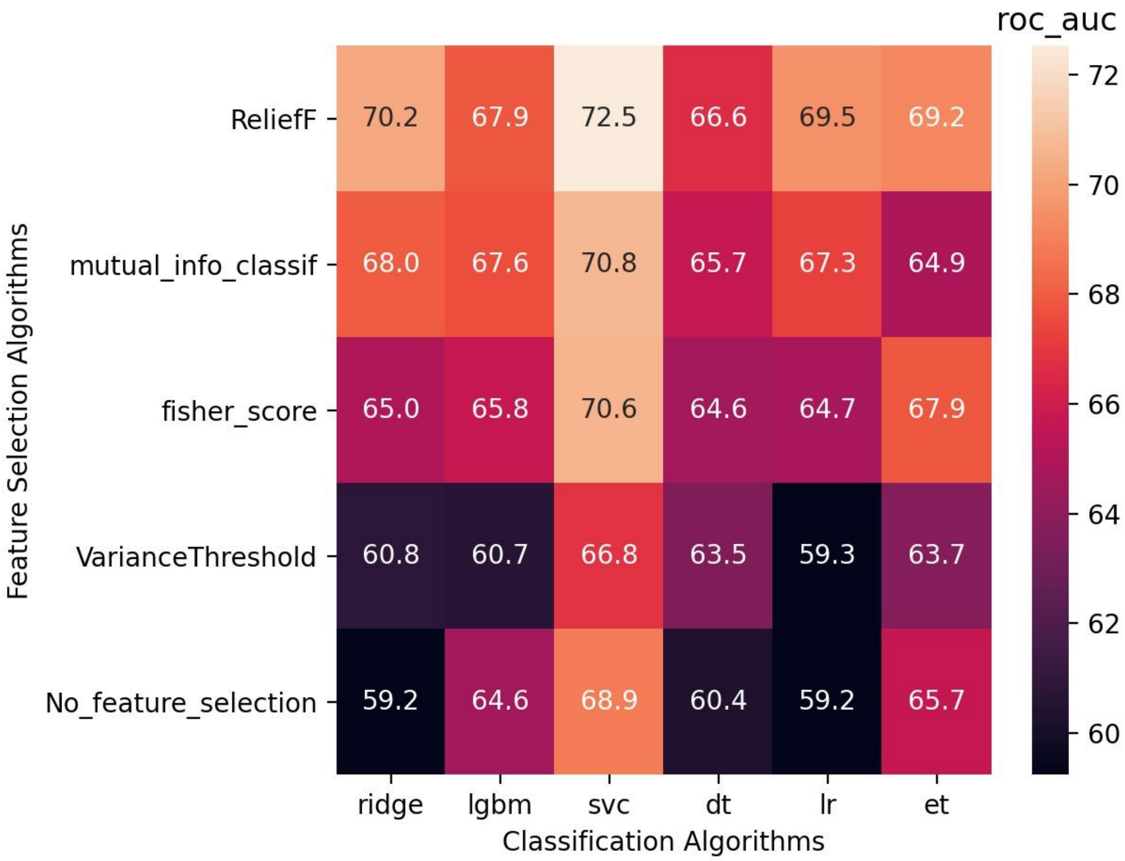


Figure 5-8 Heatmap for running the classification tool with 80 iterations, both CT scan images and PET scan images from the head and neck cancer dataset were used to generate this graph

In this dataset, the SVC classifier has done the best among the other classifiers, and with combination with ReliefF feature selector the AUC was 0.725±0.047. However, the standard deviation among the five different CV fold was 0.10475, which is much higher than the clinical data test. Table 5-5 shows the number of selected features, test scores, standard deviation between the five cross-validation folds and the standard error for the best score by algorithm. From Figure 5-9, we picked the least overlapping two values by the standard error, which are the Refieff and the variance threshold and performed the *t*-test to check if they are statistically significant.

P-value and statistical significance:
The two-tailed P value equals 0.2217
By conventional criteria, this difference is considered to be not statistically significant.
Confidence interval:
The mean of ReliefF minus VarianceThreshold equals 0.08800
95% confidence interval of this difference: From -0.06514 to 0.24114
Intermediate values used in calculations:
t = 1.3251
df = 8
standard error of difference = 0.066 (*GraphPad QuickCalcs: T Test Calculator*, n.d.).

Features that were selected more than 50% of the times in all the feature selectors, classifiers combinations are shown in Figure 5-10.

Table 5-5 Number of selected features, test scores, standard deviation and the standard error for the best score by algorithm.

Selection algorithm	**Number of selected features**	**Test scores**	**Test scores standard deviation**	**Standard error**
ReliefF	25	0.725	0.105	0.047
Mutual Information	29	0.708	0.107	0.048
Fisher Score	29	0.706	0.107	0.048
Variance Threshold	128	0.637	0.105	0.047
No feature selection	129	0.689	0.096	0.043

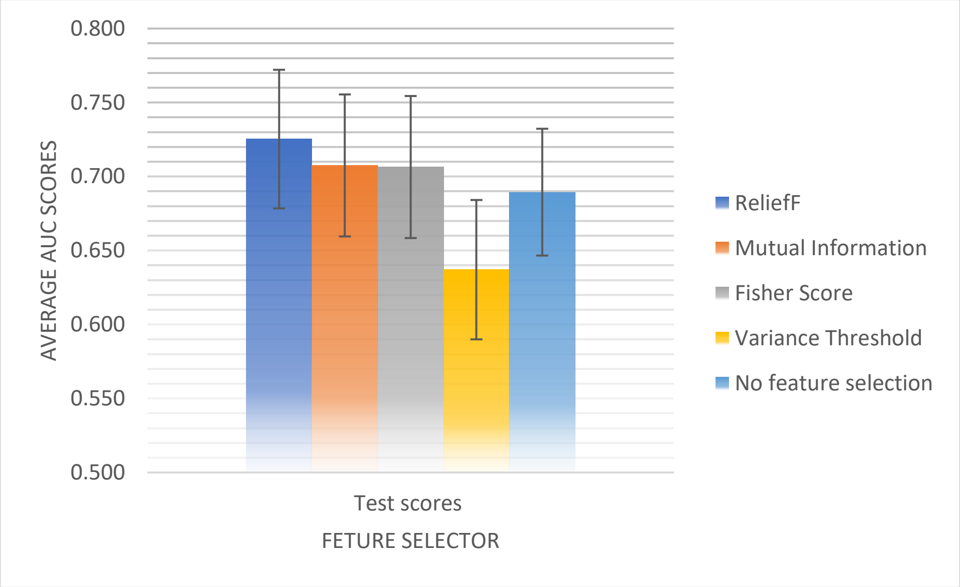


Figure 5-9 Feature selectors average scores with the standard error.

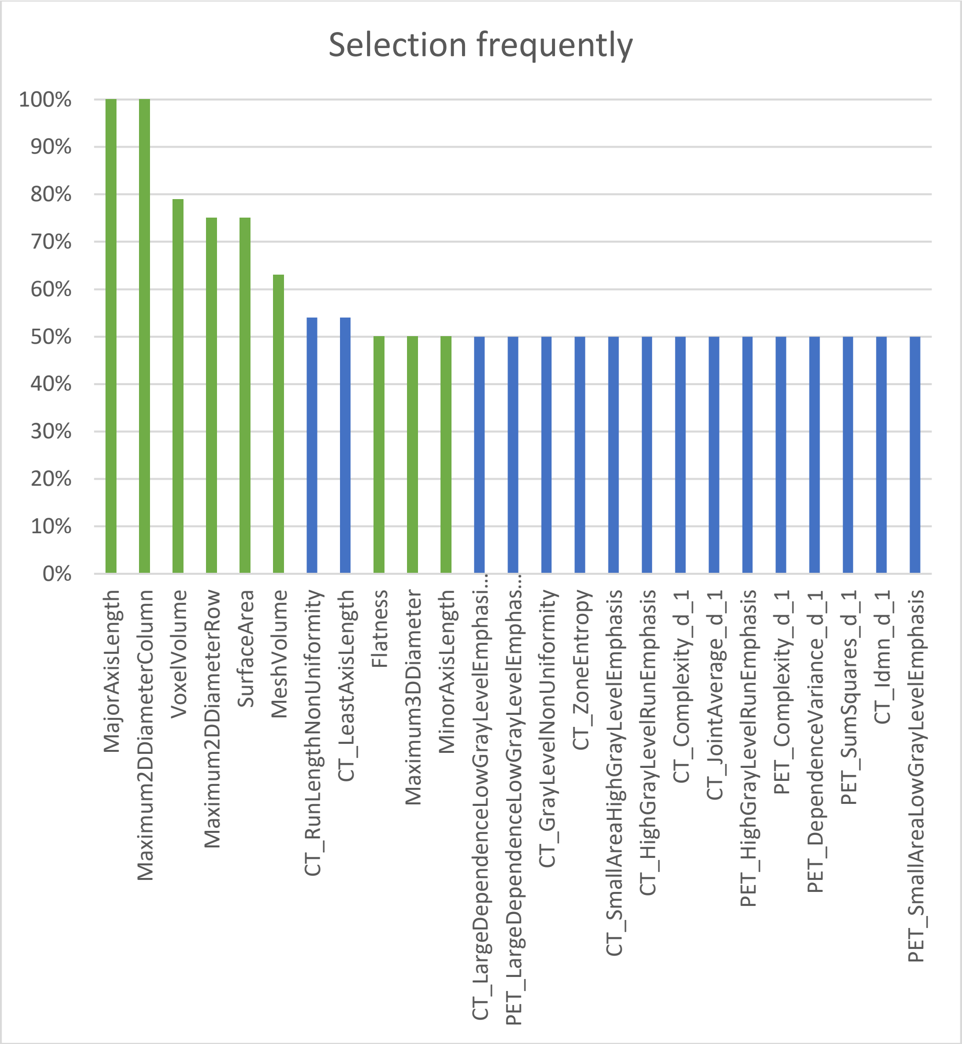


Figure 5-10 Most selected features in all classifiers and feature selectors in Biorad for the head and neck cancer dataset while using the radiomics features only for both CT and PET scan images. Shape features are in green, and texture features are in blue

The shape features were the most informative to the output variable. The top six selected features were all shape features. The MajorAxisLength and the Maximum2DDiameterColumn were selected by all the features selectors with combination with all classifiers. This means that the model is mostly using the gross tumour volume for classifications. The correlations between the selected textural features and the volume should be examined. In another study by Welch, the model predictions were made using the volume information only, and that is one of the radiomics vulnerabilities (Welch et al., 2019).

The third test was conducted using both clinical data and radiomics features from both CT scan images and PET scan images. The purpose of this test was to see if adding the radiomics feature can provide us with additional information about the response variable or not. The configurations used were similar to the second test. Results are shown in Figure 5-11 The test took 49 minutes and 34 seconds to complete.

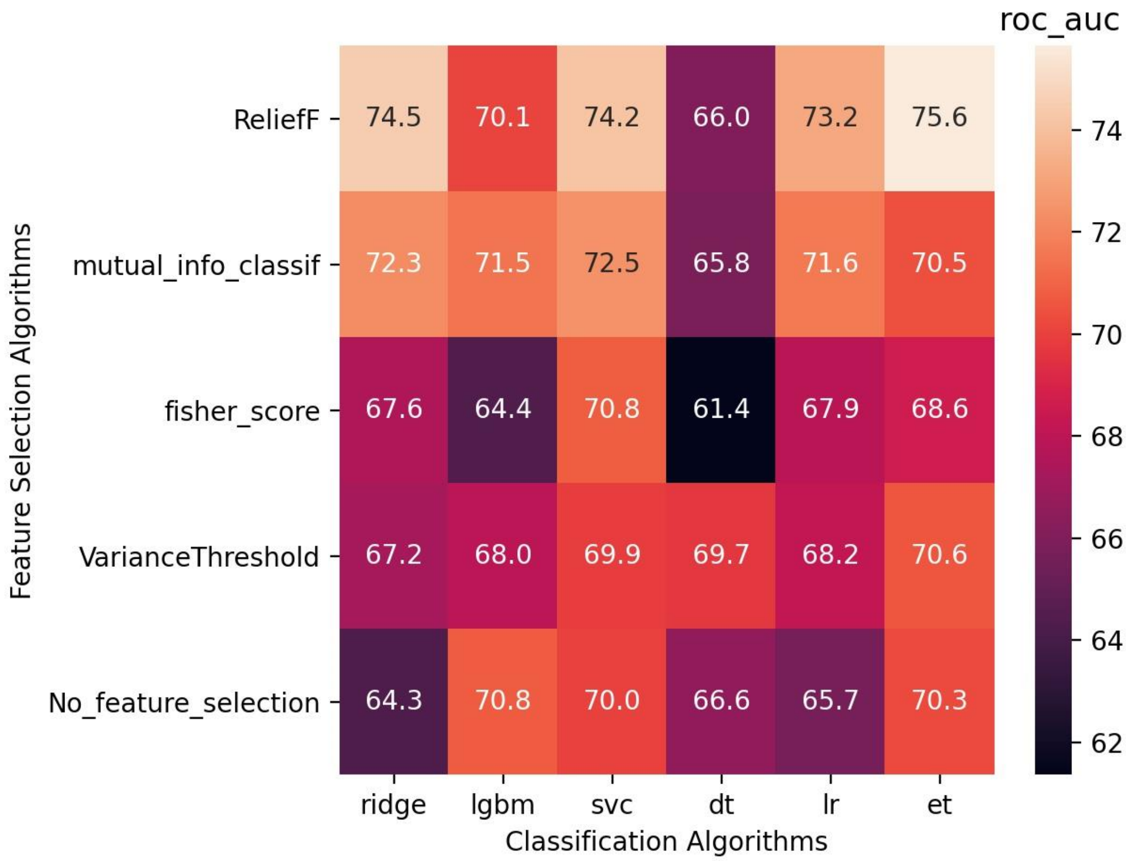


Figure 5-11 Heatmap for running the classification tool with 80 iterations, both CT scan images and PET scan images from the head and neck cancer dataset were used to generate this graph, the clinical data also were added.

The best result is improved compared to using clinical data, or radiomics only, with an AUC of 75.6±0.028 here, 72.5±0.47 for radiomics data only, and 74.5±0.035 when using clinical data only. The test score standard deviation for the best result was also improved, and it was 0.06168. However, that small improvement in the results does not give us hard evidence that the radiomics features had actually given us advantages, especially with the high variance in the results shown by the standard deviation. Table 5-6 shows the number of selected features, test scores, standard deviation between the five cross-validation folds and the standard error for the best score by algorithm. From Figure 5-12, we picked the least overlapping two values by the standard error, which are the Relief and no selection algorithm, and performed the *t*-test to check if they are statistically significant.

P-value and statistical significance:
The two-tailed P value equals 0.2522
By conventional criteria, this difference is considered to be not statistically significant.
Confidence interval:
The mean of Group One minus Group Two equals 0.04800
95% confidence interval of this difference: From -0.04170 to 0.13770
Intermediate values used in calculations:
t = 1.2340
df = 8
standard error of difference = 0.039 (*GraphPad QuickCalcs: T Test Calculator*, n.d.).

Table 5-6 Number of selected features, test scores, standard deviation and the standard error for the best score by algorithm.

Selection algorithm	Number of selected features	Test scores	Test scores standard deviation	Standard error
ReliefF	11	0.756	0.062	0.028
Mutual Information	29	0.725	0.099	0.044
Fisher Score	19	0.708	0.115	0.052
Variance Threshold	136	0.706	0.072	0.032
No feature selection	205	0.708	0.061	0.027

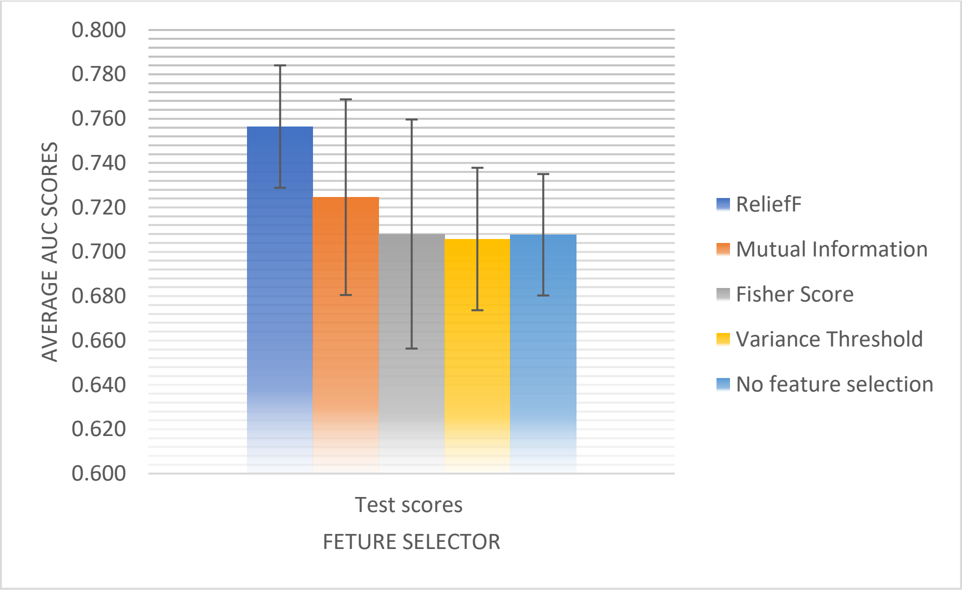
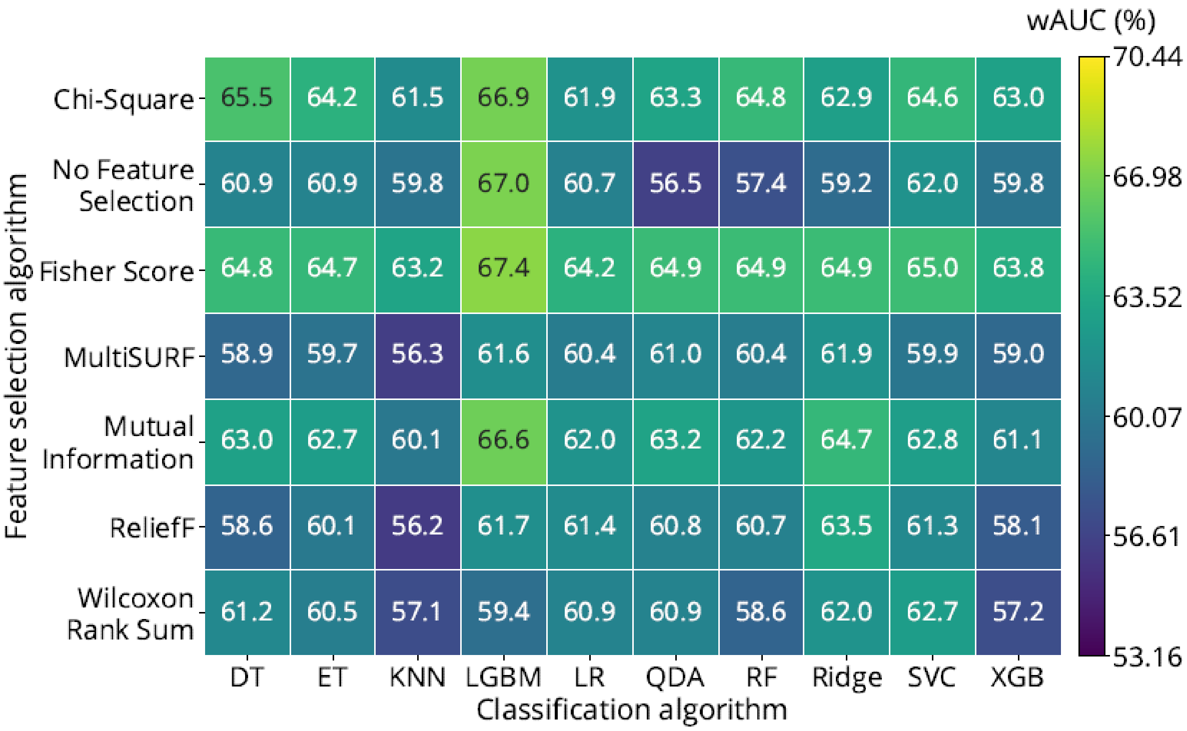
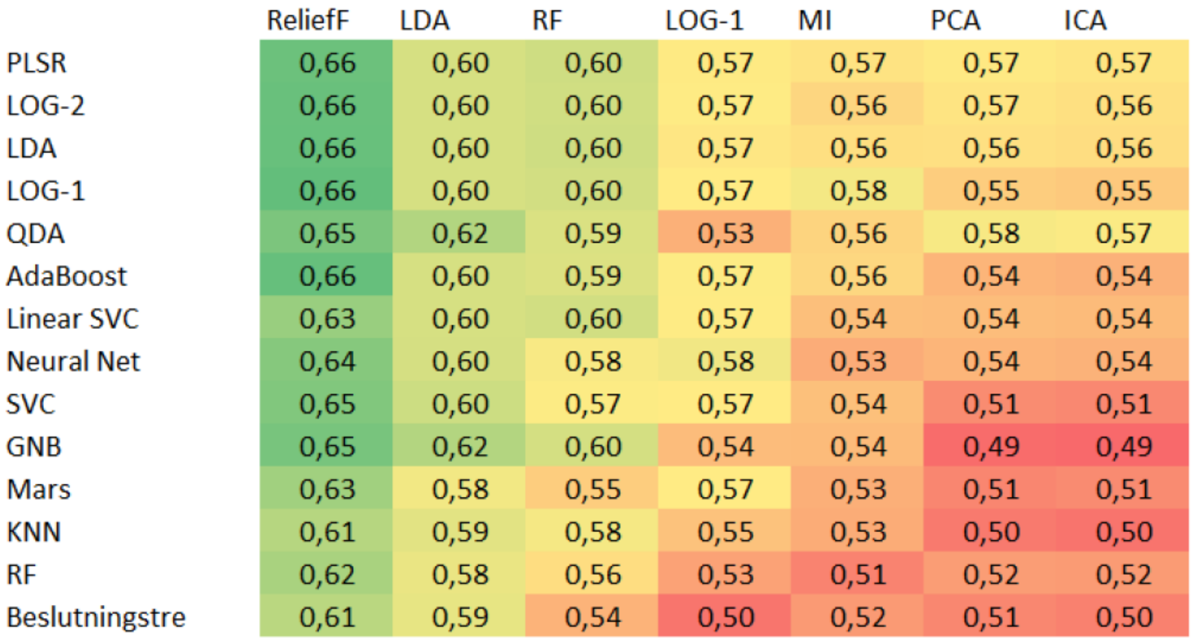


Figure 5-12 Feature selectors average scores with the standard error.



	DT	ET	KNN	LGBM	LR	QDA	RF	Ridge	SVC	XGB
Chi-Square	65.5	64.2	61.5	66.9	61.9	63.3	64.8	62.9	64.6	63.0
No Feature Selection	60.9	60.9	59.8	67.0	60.7	56.5	57.4	59.2	62.0	59.8
Fisher Score	64.8	64.7	63.2	67.4	64.2	64.9	64.9	64.9	65.0	63.8
MultiSURF	58.9	59.7	56.3	61.6	60.4	61.0	60.4	61.9	59.9	59.0
Mutual Information	63.0	62.7	60.1	66.6	62.0	63.2	62.2	64.7	62.8	61.1
ReliefF	58.6	60.1	56.2	61.7	61.4	60.8	60.7	63.5	61.3	58.1
Wilcoxon Rank Sum	61.2	60.5	57.1	59.4	60.9	60.9	58.6	62.0	62.7	57.2

Figure 5-13 Results of clinical factors in addition to the radiomics data (Langberg, 2019), with permission.



	ReliefF	LDA	RF	LOG-1	MI	PCA	ICA
PLSR	0,66	0,60	0,60	0,57	0,57	0,57	0,57
LOG-2	0,66	0,60	0,60	0,57	0,56	0,57	0,56
LDA	0,66	0,60	0,60	0,57	0,56	0,56	0,56
LOG-1	0,66	0,60	0,60	0,57	0,58	0,55	0,55
QDA	0,65	0,62	0,59	0,53	0,56	0,58	0,57
AdaBoost	0,66	0,60	0,59	0,57	0,56	0,54	0,54
Linear SVC	0,63	0,60	0,60	0,57	0,54	0,54	0,54
Neural Net	0,64	0,60	0,58	0,58	0,53	0,54	0,54
SVC	0,65	0,60	0,57	0,57	0,54	0,51	0,51
GNB	0,65	0,62	0,60	0,54	0,54	0,49	0,49
Mars	0,63	0,58	0,55	0,57	0,53	0,51	0,51
KNN	0,61	0,59	0,58	0,55	0,53	0,50	0,50
RF	0,62	0,58	0,56	0,53	0,51	0,52	0,52
Beslutningstre	0,61	0,59	0,54	0,50	0,52	0,51	0,50

Figure 5-14 Average AUC for 40 tests of classification algorithms in combination with the feature selectors. The dataset includes features extracted from the square root transformed PET and CT images, shape properties and clinical factors. These results were obtained by Midtfjord in her thesis using the same head and neck cancer dataset (Midtfjord, 2018), with permission.

By comparing the results of the test done in Biorad with both Langberg results in Figure 5-13 and Midtfjord results in Figure 5-14, we notice that the best results are very similar between the last two, but the Biorad achieved higher AUC, that might be partially because of the high variance in the results between the different CV folds. However, in both Biorad and Midtfjord results, the ReliefF was the best feature selector.

The two most selected features were the same as the previous test, and the third one was the first one on the first test, where we used the clinical data only as shown in Figure 5-15.

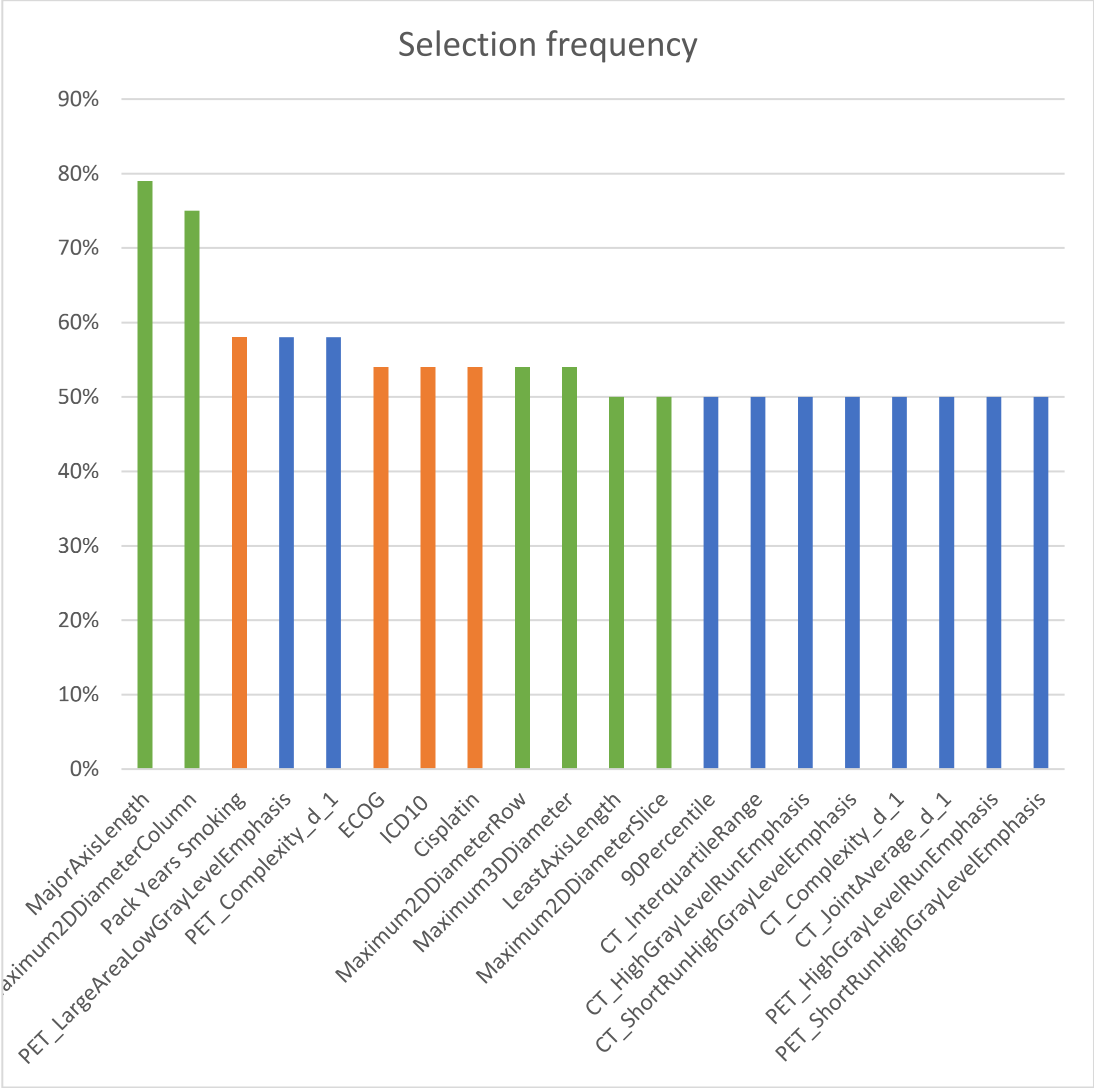


Figure 5-15 Most selected features in all classifiers and feature selectors in Biorad for the head and neck cancer dataset while using both radiomics and the clinical data. Shape features are in green, texture features are in blue and medical factors are in orange.

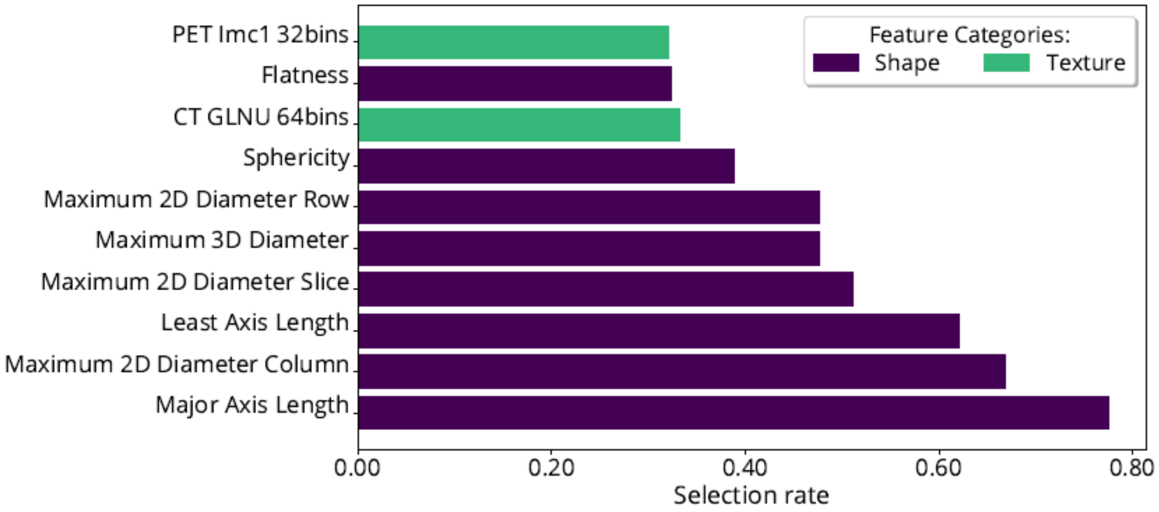


Figure 5-16 The rate of the most selected features regardless of the category, results from (Langberg, 2019), with permission.

In both Figure 5-15, the most selected features by Biorad and Figure 5-16, the most selected features in Langberg's thesis, we notice the dominance of the shape features in the list, which brings us back to Welch's study (Welch et al., 2019).

Below are more details about the selected features by each selector. The mutual information score for the top 15 features is shown in Figure 5-17. The mutual info score did not give the clinical data high scores. The highest scores are for the ECOG and was rated as the 23rd feature, and the Pack Years Smoking which was rated as the 35th feature. Most selected features by Variance Threshold are shown in Figure 5-18, and it includes textural features only.

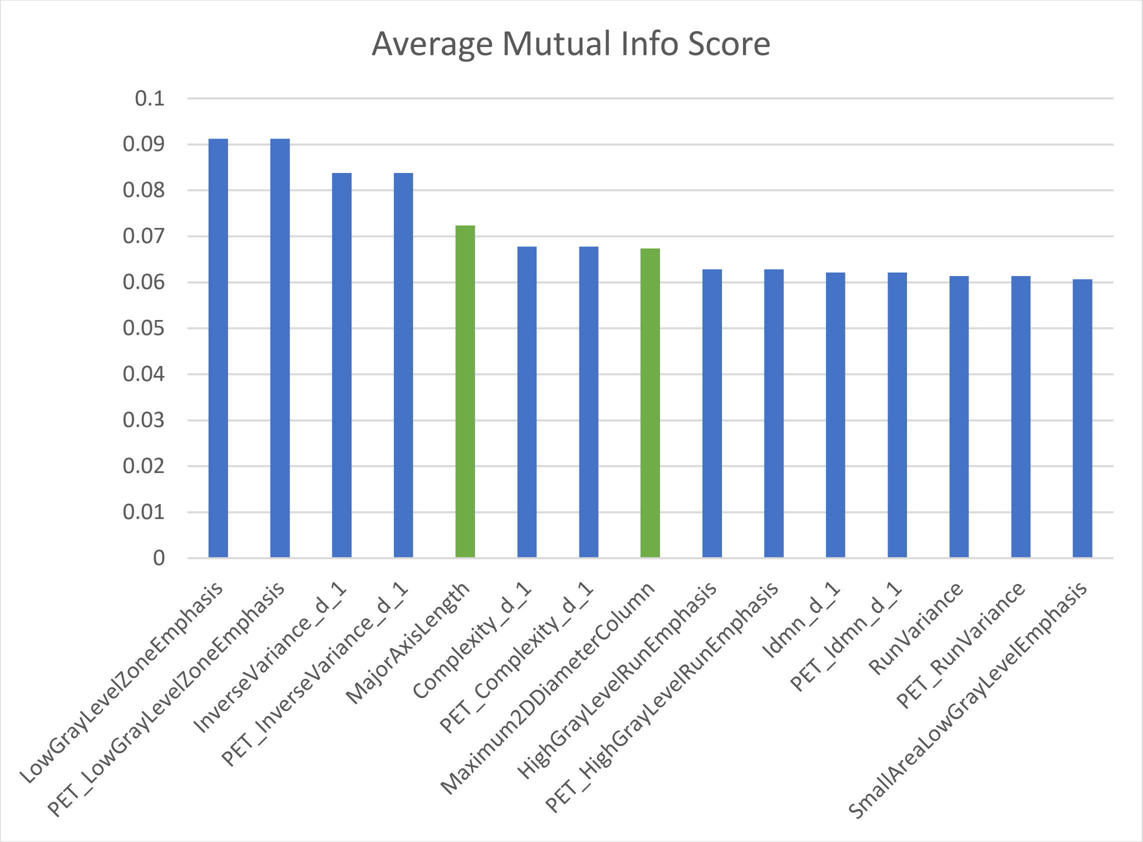


Figure 5-17 Average mutual information score among all the experiments, head and neck cancer dataset used with CT, PET scan images and the clinical data. Shape features are in green, and texture features are in blue

The variance that is used in variance threshold favoured the radiomics features also, as shown in Figure 5-18.

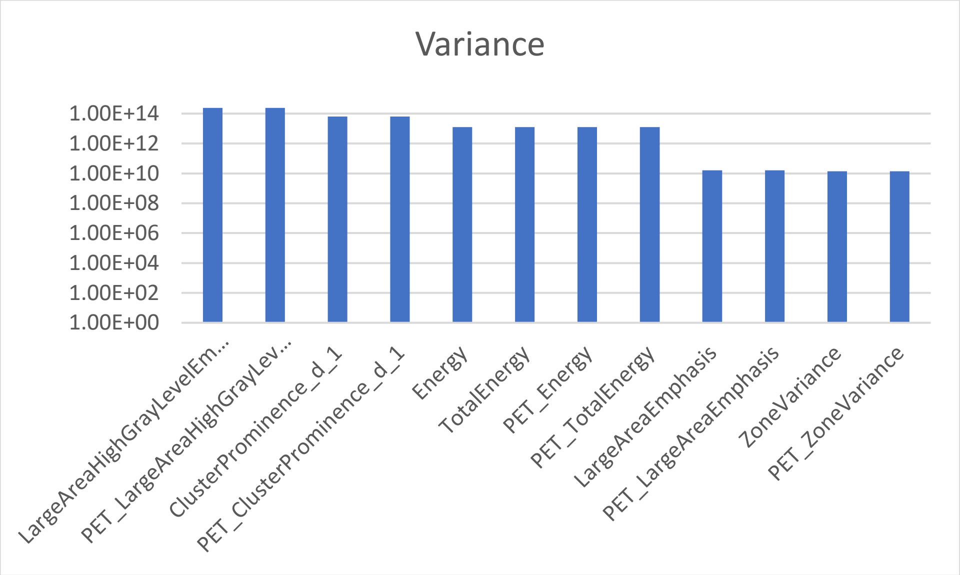


Figure 5-18 Top features by variance (logarithmic scale), head and neck dataset, CT, PET scan images and Clinical data. All top variance features are texture features

Only the multivariate filter selection, which is the ReliefF method favoured the clinical data, and at the same time, it did perform best in almost all classifiers. The top features shown in Figure 5-19. and in Figure 5-20. So, it would be a good idea to add more multivariate filter methods to the Biorad application in the future, as the univariate methods seem to select sub-optimal subsets of the features. The Fisher scores are unfortunately not available in Biorad because the method used does not provide a way of retrieving the scores.

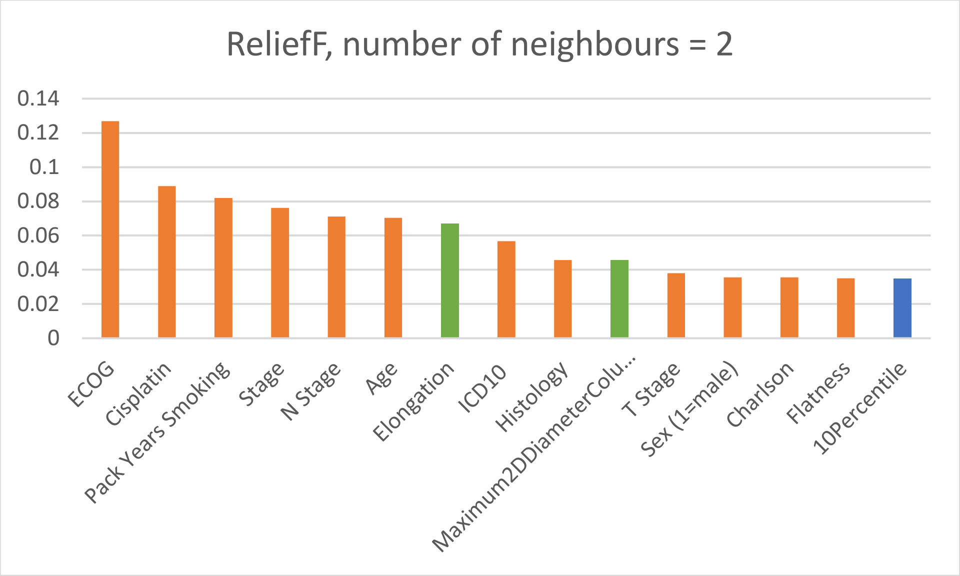


Figure 5-19 ReliefF top score features when the number of neighbours = 2. Shape features are in green, texture features are in blue and medical factors are in orange.

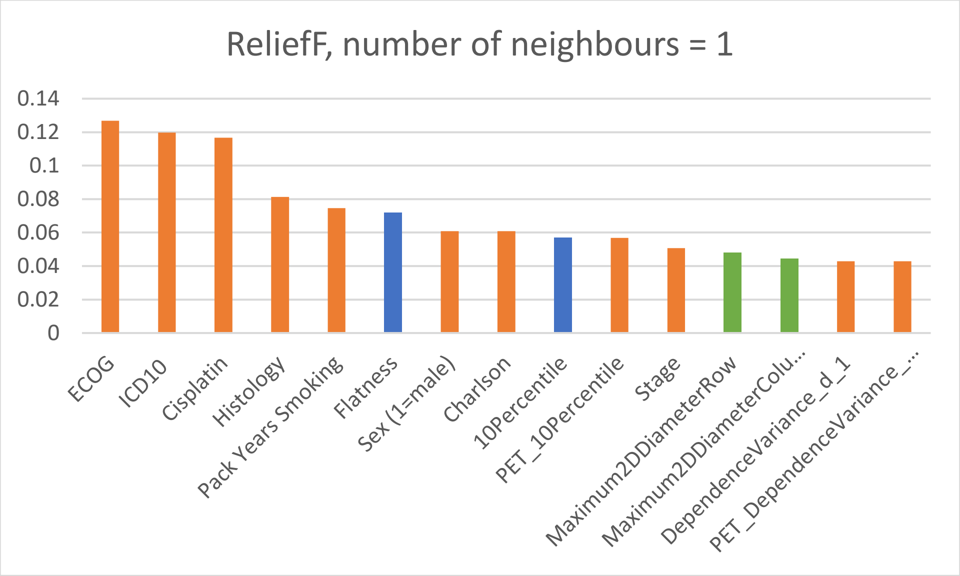


Figure 5-20 ReliefF top score features when the number of neighbours = 1. Shape features are in green, texture features are in blue and medical factors are in orange.

In five out of six experiments with RelieffF, the number of neighbours selected was 2, and in the remaining experiment it was 1. So, another experiment was conducted to estimate the best range to tune the number of neighbours in RelieffF. This time the range was (5 to 6) instead of the previous one which was (1 to 3). In the results, in all six experiments the lower limit, which is five was selected, and the scores were not improved compared to the previous test, but the opposite, as shown in Figure 5-21.

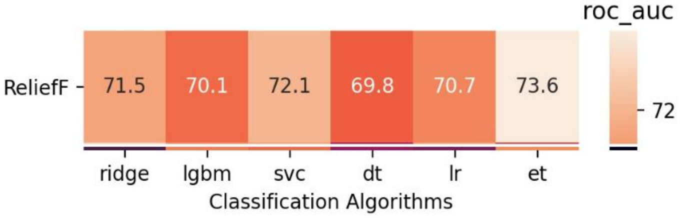


Figure 5-21 ReliefF scores for the head and cancer dataset, the number of neighbours selected was 5.

In chapter 3.7.2.1, we mentioned that the larger the number of neighbours, the better sub-set of features we get, but that was not the case in this experiment. The best value of the number of neighbours was 2, and the results tend to get worse when we increase this value; that can happen because the ReliefF algorithm converges to univariate selectors as the number of neighbours increase (Mckinney et al., 2013). In Figure 5-22, we have the top 10 features selected with the number of features = 5.

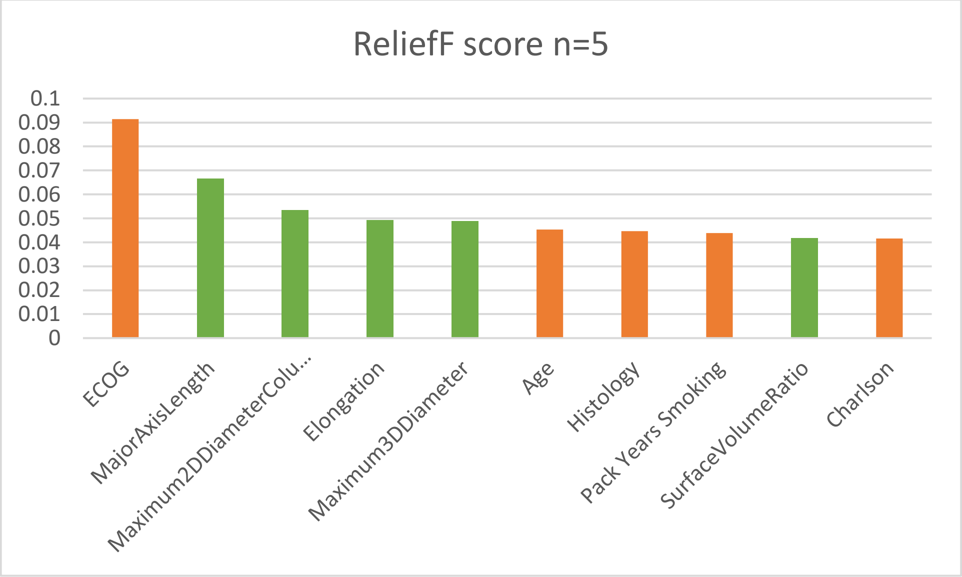


Figure 5-22 Top features scores by ReliefF when the number of neighbours = 5 Shape￼ features are in green, texture features are in blue and medical factors are in orange.

For comparison, the following results are from the same experiment but with '20' as the maximum number of iterations to select the optimal hyperparameters.

The trial took 12 minutes and 13 seconds. Results were slightly worst then the experiment done with 80 different iterations to find the optimal parameters. The heatmap of the results is shown in Figure 5-23. And the most selected features are shown in Figure 5-24.

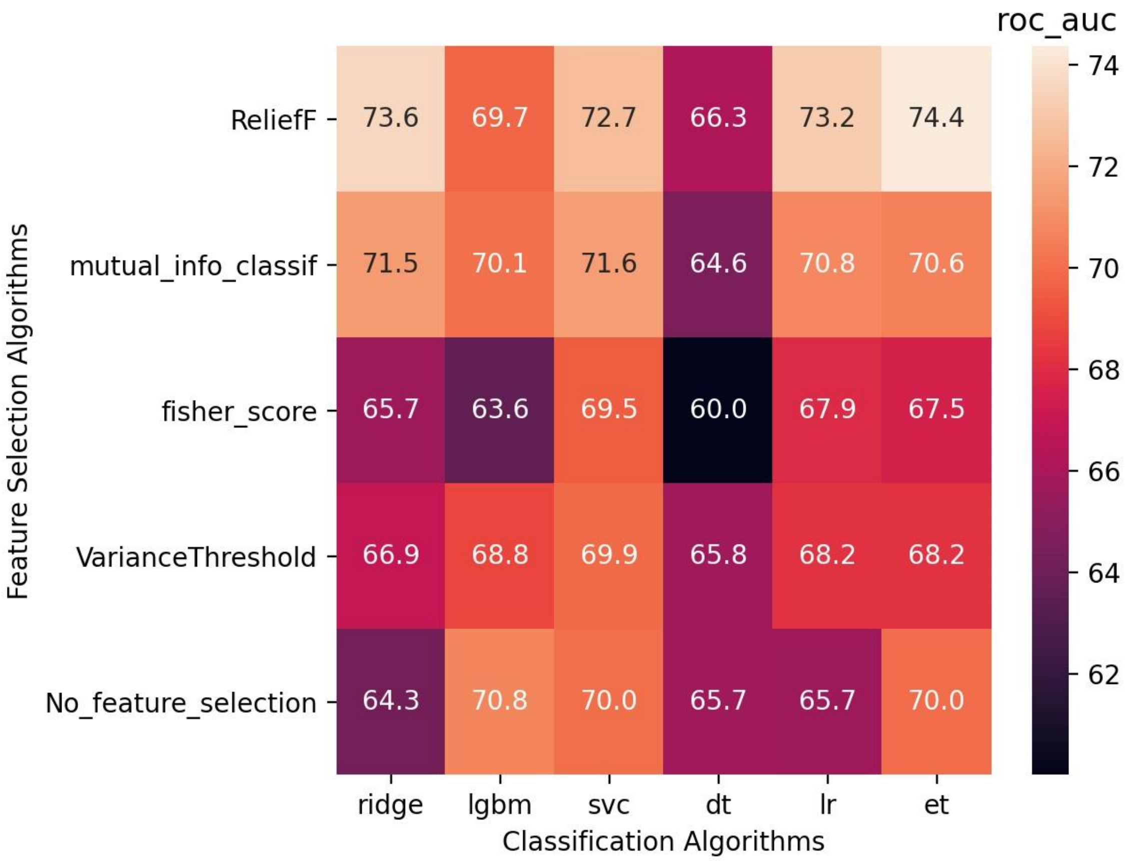


Figure 5-23: Heatmap for running the classification tool with '20' iterations, head and neck dataset was used to generate this graph

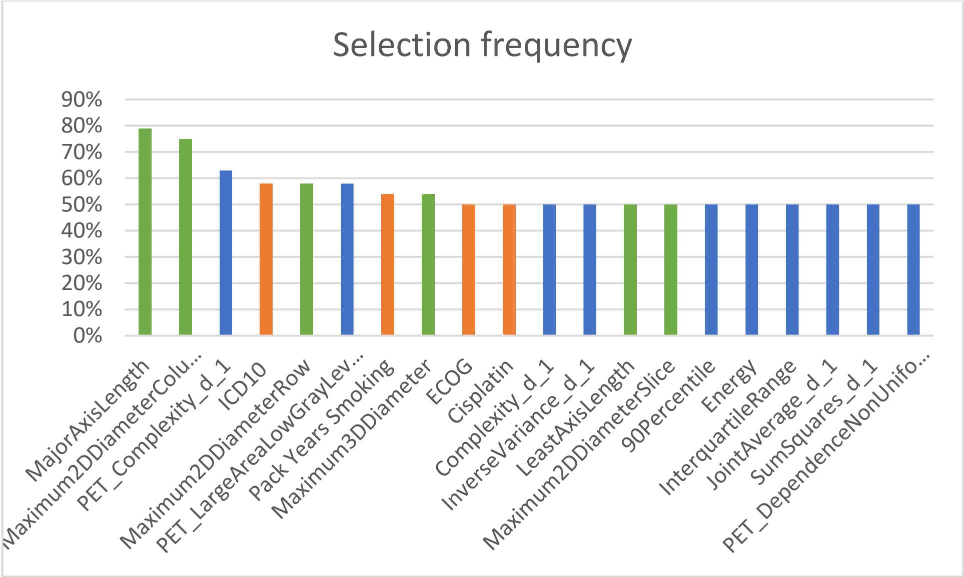


Figure 5-24 Most selected features in all classifiers and feature selectors in Biorad for the head and neck cancer dataset while using both radiomics and the clinical data, maximum iterations is 20. Shape features are in green, texture features are in blue and medical factors are in orange.

For the next test, the F1 score was used, all other configurations were similar to the third test that included all the radiomics features of the CT/PET scan images and the clinical factors with a maximum of 80 iterations to find the optimal parameters. The test took 51 minutes and 53 seconds to complete. Results are shown in Figure 5-25. The best result was given by the Ridge classifier and the ReliefF feature selector,0.661±0.022.

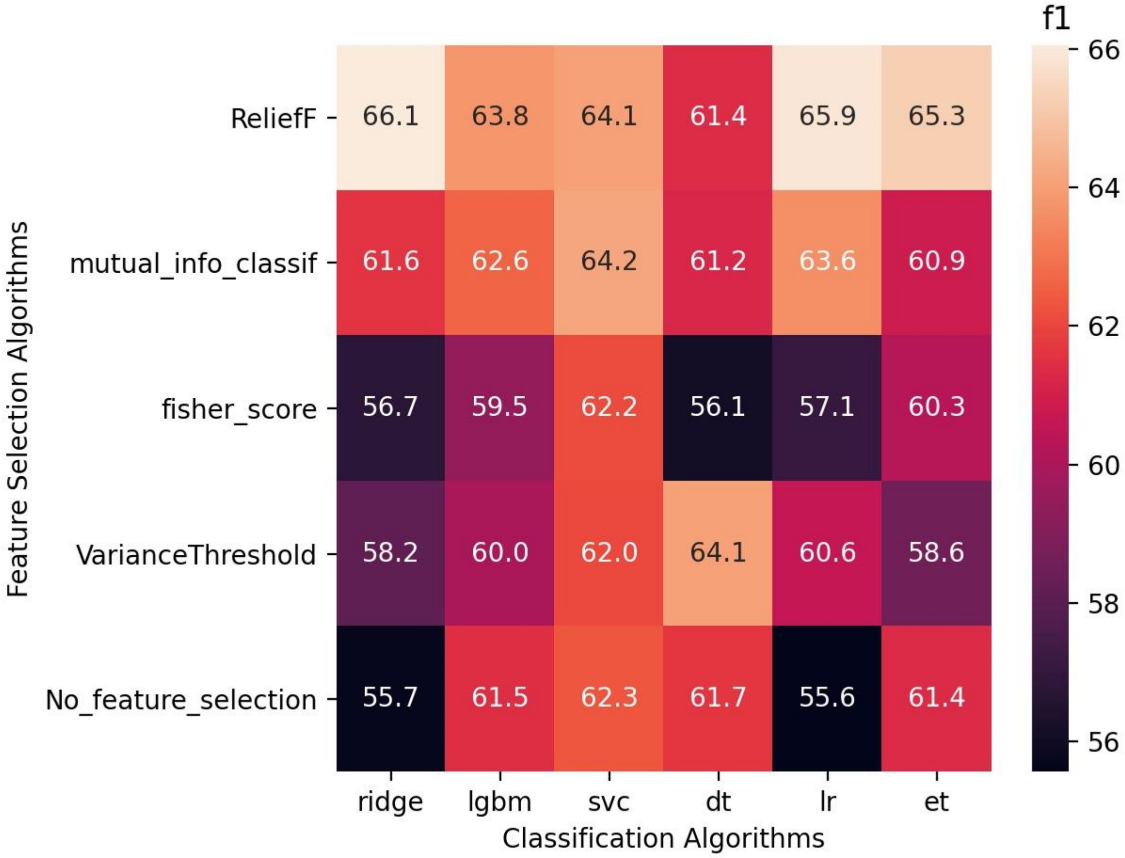


Figure 5-25 Heatmap for running the classification tool with 80 iterations, both CT scan images and PET scan images from the head and neck cancer dataset were used to generate this graph, the clinical data also were added. The F1 score used here.

Figure 5-26, gives the frequency of each feature getting selected using F1 scoring. Five out of the top ten selected features are shape features, including the top three, and three of them are clinical data. Pack Year Smoking feature is still among the top of the clinical data and was selected the same number of times as when using the roc_auc scoring.

F1 score is mostly useful when dealing with unbalanced datasets, but the head and neck cancer patients' dataset was balanced, and since we are here focusing on both positive and negative classes, the AUC would be more informative as a measure for the model performance (Aoullay, 2018), (Shung, 2020).

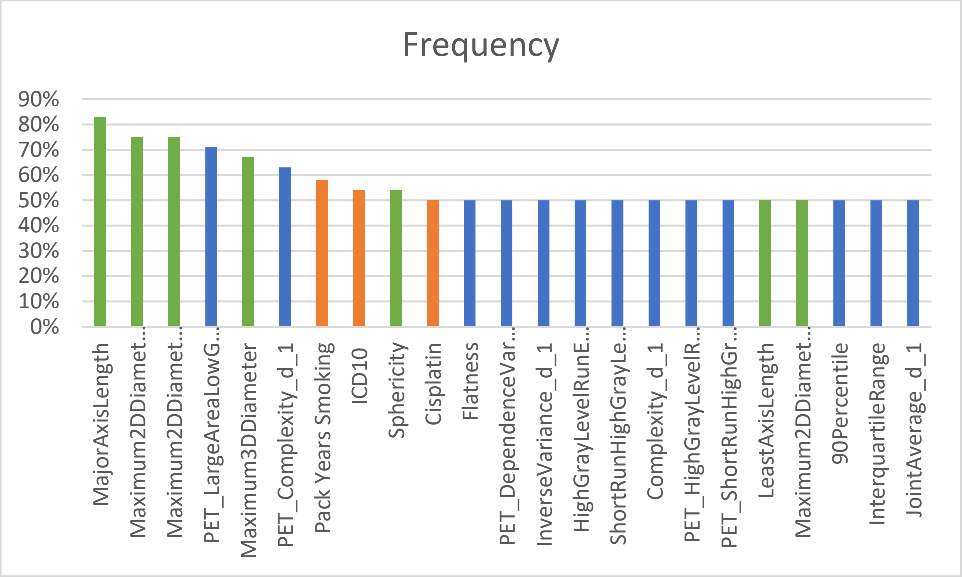


Figure 5-26 Most selected features in all classifiers and feature selectors in Biorad for the head and neck cancer dataset while using both radiomics and clinical factors, and F1 for scoring.

5.4 Other datasets

The biorad application was used and tested with several datasets:

- The scikit-learn wine dataset, as a sample test for multiclass classification.
- The Effect of MPH-treatment in Attention deficit hyperactivity (ADHD) Diagnosed Children, by Master student Inger Annett Grünbeck from NBMU.
- The rectum cancer radiomics survival rate, by Master student Aase Mellingen Langan from NTNU.

5.4.1 Scikit-learn Wine recognition Dataset

The biorad application supports the multiclass classification as well. For testing, the scikit-learn wine dataset was used. In the configurations, F1_weighted scoring was selected and 80 was the maximum number of iterations. Details of the dataset are available in Appendix A.2 Wine recognition dataset.

The heatmap is shown in Figure 5-27.

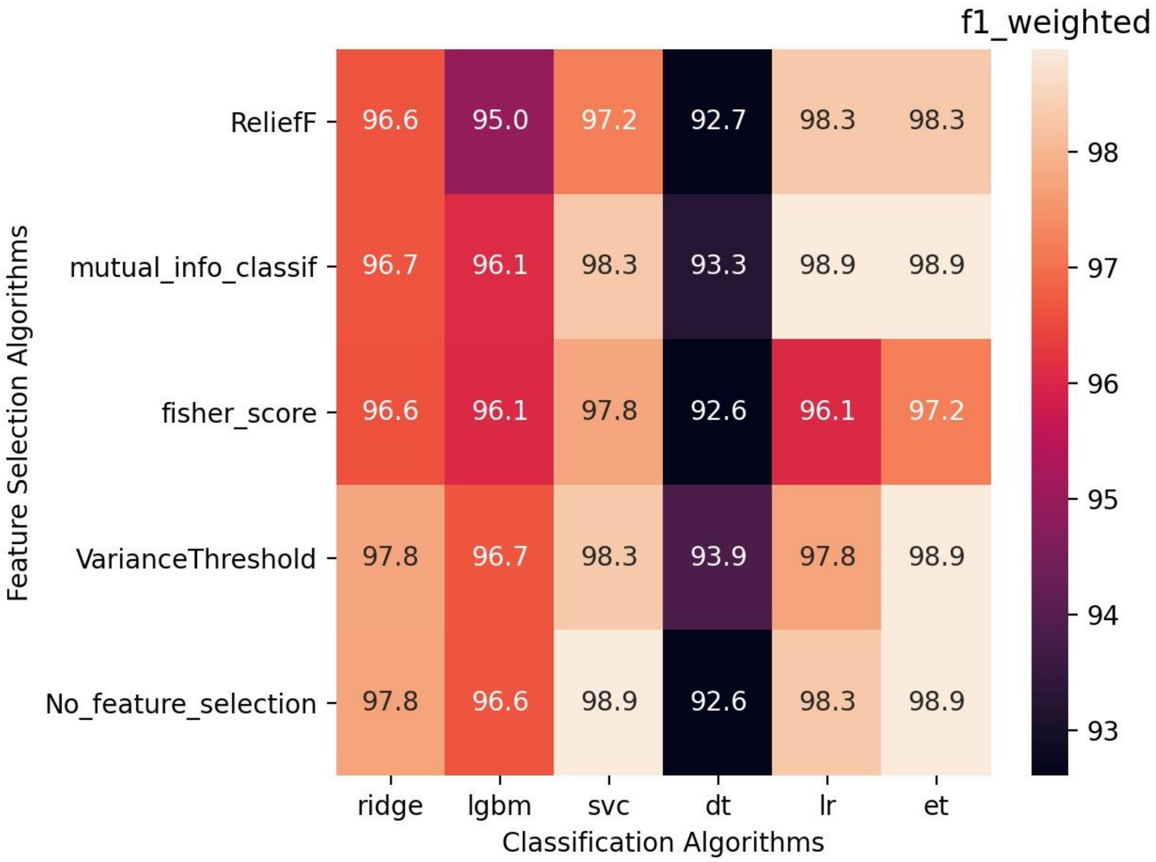


Figure 5-27 Multiclass classification of the scikit-learn wine dataset.

In the wine recognition test, the decision tree did overfit the training data with a perfect score almost every time and performed poorly in the test set with high variance between the test folds. On the other hand, all other classifiers performed well on this dataset, with a low variance between the multiple folds – (standard deviation between 0.011 to 0.030). The most selected features are shown in Figure 5-28.

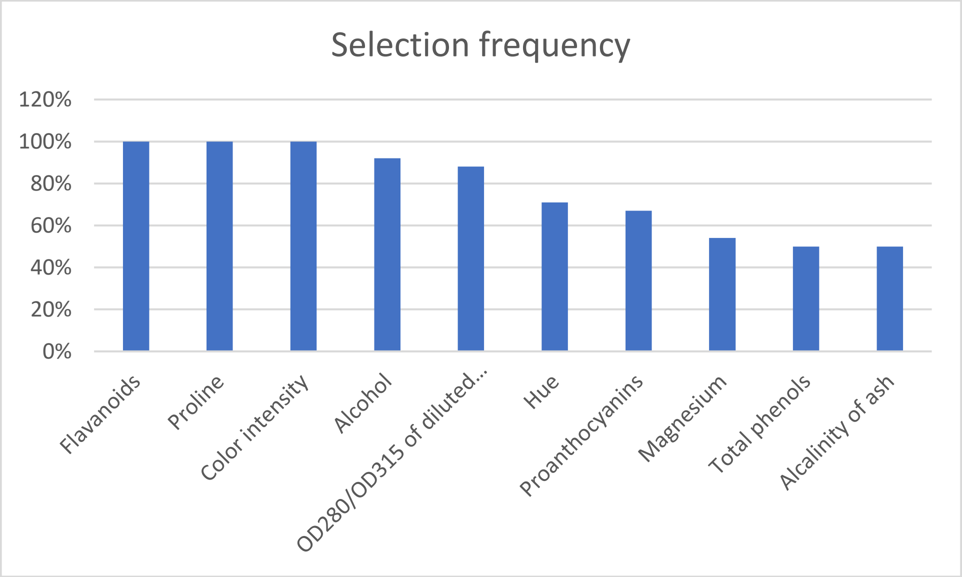


Figure 5-28 Most selected features from scikit-learn wine dataset in all classifiers and feature selectors.

5.4.2 The Effects of MPH-Treatment in ADHD-Diagnosed Children. An Explorative Analysis Using Radiomic Feature

Master student Inger Annett Grünbeck studied the effect of Methylphenidate (MPH) treatment in Attention deficit hyperactivity (ADHD) diagnosed children in her thesis (I. A. Grünbeck, 2020).

It was a binary classification problem with 42 samples, 22 Class MPH-treated and 24 placebo-treated children. T1-weighted MR images were obtained to analyse the differences between placebo and MPH treated participants in different parts of the brain. Figure 5-29 shows the right thalamus part of the brain in one patient; other areas of interest are caudate, hippocampus, pallidum and putamen.

The radiomics features from the MRI images were extracted using Biorad feature extraction module and then she analysed the data with Biorad also. The heatmaps of the radiomics analysis of different parts of the brain are shown in Figure 5-30.

Some of the results were good (above 0.7 roc_auc) but the standard deviation of the results among the different five cross-validation folds was high, most probably due to the limited number of samples (only 46 samples). For example, in the Pallidum region of the brain (Variance Threshold features selector, and the Extra Tree classifier), The ROC AUC was 0.786, but the standard deviation was 0.16305 which is too high.

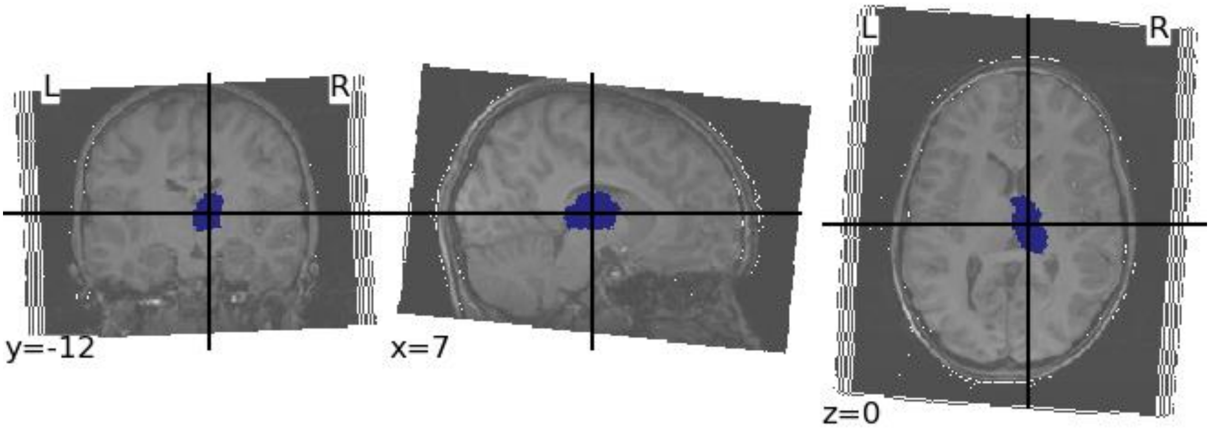
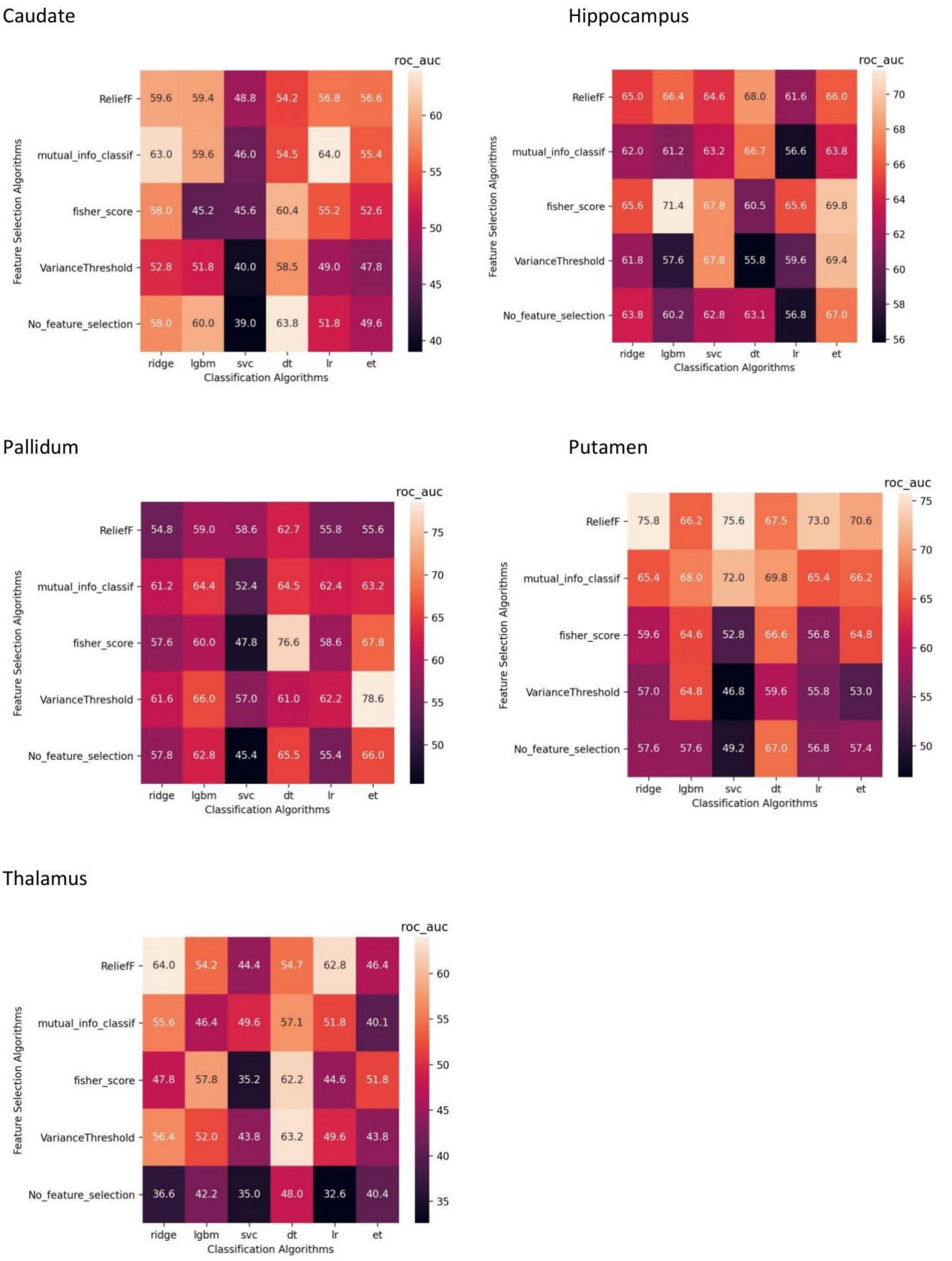


Figure 5-29 MRI image of a child, the region of interest delineated is the right thalamus.



Caudate

Hippocampus

Pallidum

Putamen

Thalamus

Figure 5-30 Radiomics analysis of the Effect of MPH-treatment in ADHD-Diagnosed Children on different parts of the brain

5.4.3 MRI-based radiomics analysis for predicting treatment outcome in rectal cancer

Master student Aase Mellingen Langan worked on the MRI-based radiomics analysis for predicting treatment outcome in rectal cancer for her research (Langan, 2020).

Dataset used in her research:

- T2-weighted (T2W) and diffusion-weighted (DW) MR images from 81 patients with rectal cancer, all had surgery. Thirty-five of these patients had preoperative treatment, referred to as the nCRT cohort.
- Seven DWIs obtained for each patient.
- Tested four combinations of samples and response varieble (RV):
    - All patients + progression free survival (PFS)
    - nCRT cohort + PFS
    - nCRT cohort + tumor regression grade (TRG)
    - nCRT cohort + ypT

Some of the main results were as follows:

Model performance:

- All patients (n = 81) predicting PFS, features from T2W images and one DWI for each patient (number of features per patient = 214), Figure 5-31.

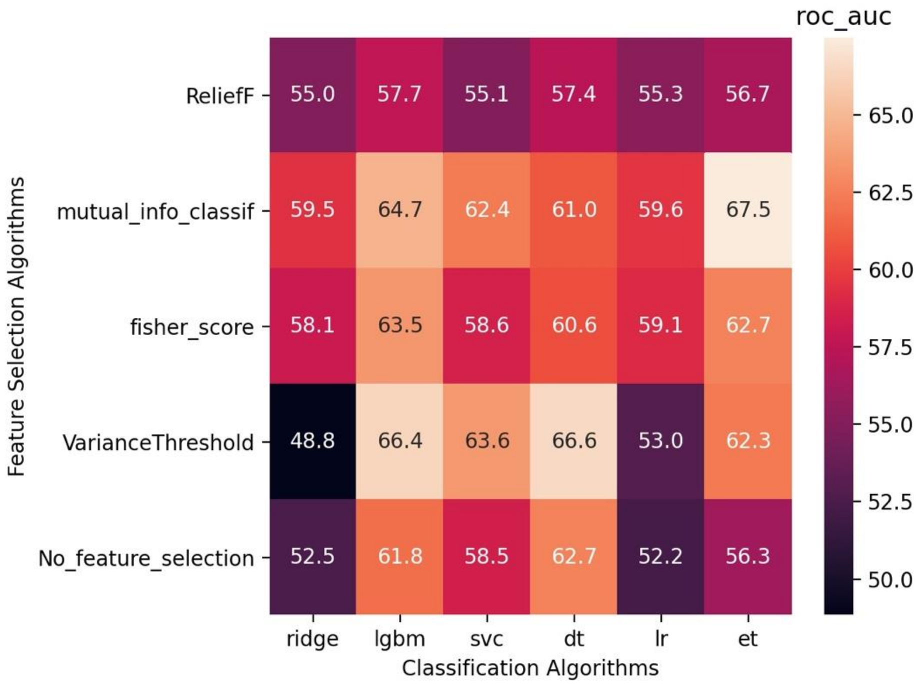


Figure 5-31 All patients (n = 81) predicting PFS, features from T2W images and one DWI for each patient (number of features per patient = 214). Test standard deviation ranges from 4.9 - 20.0 % (excluding models with no feature selection).

Mutual information and Extra trees:

- Test: 67.5 ± 15.0% (15% is the standard deviation)
- Train: 76.0 ± 3.7 %

Fisher score and Logistic regression:

- Test: 59.1 ± 9.6 %
- Train: 60.5 ± 1.4 %

- nCRT predicting TRG, features from T2W images and one DWI for each patient (number of features per patient = 214), Figure 5-32.

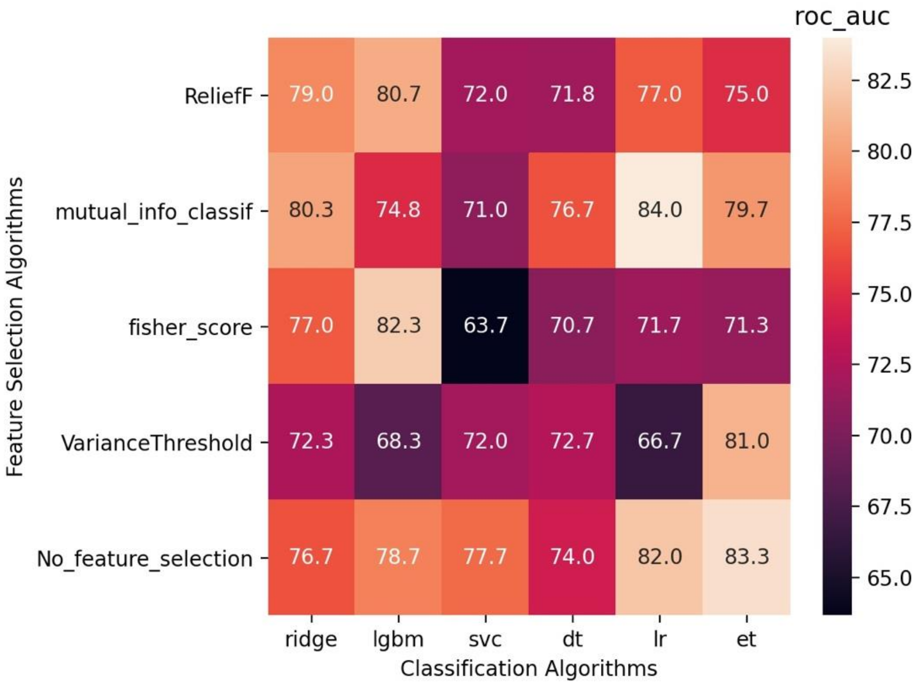


Figure 5-32 nCRT predicting TRG, features from T2W images and one DWI for each patient (number of features per patient = 214). Test standard deviation ranges from 3.4 - 24.2 %.

Mutual information and Decision tree:

- Test: 76.7 ± 4.6 %
- Train: 79.0 ± 2.6 %

Mutual information and Logistic regression:

- Test: 84.0 ± 16.9 %
- Train: 87.2 ± 5.5 %

- Selection rates from experiments performed with all patients (n = 81) predicting PFS

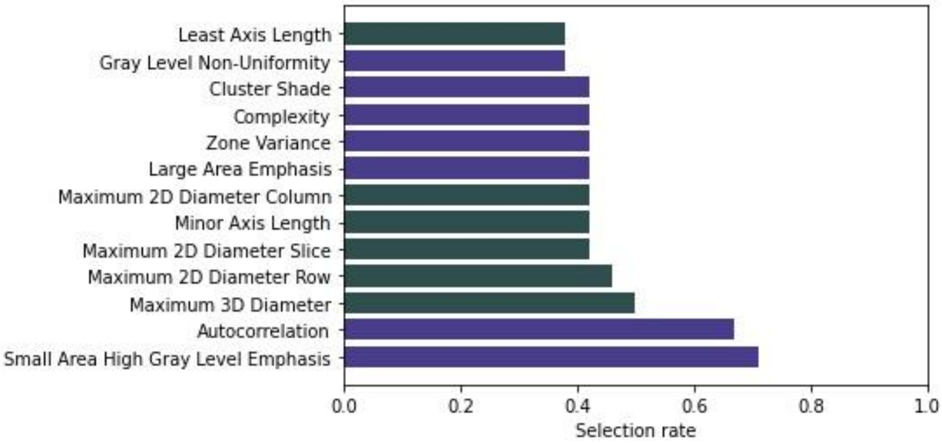


Figure 5-33 Features from T2W images only.

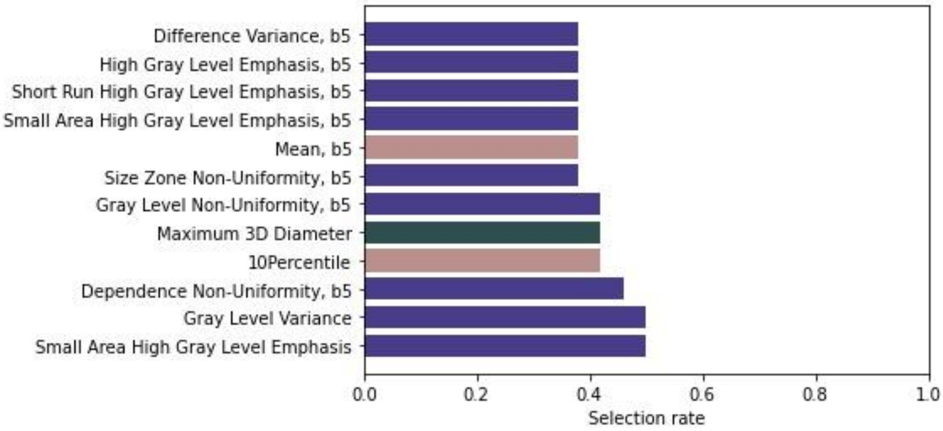


Figure 5-34 Features from T2W images and one DWI for each patient. Features with rates > 0.33 (8/24) only included.

The small area high grey level emphasis feature also most selected (rate 0.75) when predicting PFS for the nCRT cohort based on features from T2W images.

The first evaluation of reproducibility: experiments performed with changed bin width (before 25, now 35), delineation of the volume of interest (before mask1, now mask2), and resampling of voxel size (before not isotropic in the z-direction, now 1x1x1 mm^3), respectively.

- All over: poor reproducibility of results, especially with respect to the features selected. Evaluating feature correlation and getting rid of redundant features may improve this.

6 Recommendation and conclusions

The Biorad application has been used already in many applications. The feature extraction module, for example, was used by Isak Biringvad Lande in his thesis about nuclear forensics for analysing scanning electron microscope images of uranium concentrate ores (Lande, 2020). And both of Biorad application modules were used by both Grünbeck in her research regarding The Effect of MPH-treatment in ADHD-Diagnosed Children, and Langan in her research about MRI-based radiomics analysis for predicting treatment outcome in rectal cancer.

The results varied, but in general, the smaller number of samples, the less reliable the results. In both Grünbeck and Langan, the standard deviation of the scores was very high.

On the other hand, in the main test of the application that used the head and neck dataset, the results were promising, the scores were higher than a similar experiment with the same dataset done by Langberg, and the most informative features were very similar in both experiments. However, we can notice that the shape features were the most selected, which means that the models are mostly predicting the response based on the gross volume of the tumour, not on the textural features. Further work should be done to examine the correlations between the selected textural features and the volume of the tumour (Welch et al., 2019).

The best results in the tests of the application were given by ReliefF, which is a multivariant feature selection method. On the other hand, other feature selection methods seemed to fail to select the optimal subset of the features, and the worst result was given by the Variance Threshold feature selection method, which seemed to favour the texture features only and omit the most informative features from the clinical factors and shape features.

The programming work was in parallel with the other students' usage of the application. So, the feedback was beneficial to add features that end-users are interested in. However, some of the suggestions were received too late to be implemented and tested properly, and it would be valuable if they were added in the future to the application. For example, it would be very beneficial to have the feature selection rate for each feature selection method. It is possible to get all the data from the output CSV file, but it takes too much time to do this task manually. Another good addition to the application would be adding more flexibility, such that the users can be able to choose which algorithm they want to run, and which parameters they want to tune from the configuration JSON file. Also generating informative graphs, like the error bars from the data, is a lengthy process. It will save much time from researchers if the Biorad application is able to generate such graphs.

It would also be useful to add the more feature selectors even those that require long processing time but give the users the option to use them or not like the MultiSURF and the Recursive Feature Elimination (RFE) and Chi-Square which performed well in Langberg's thesis (Langberg, 2019). More classifiers such as the random forest and the k-nearest neighbours, which performed well in a study for the survival rate of head and neck cancer could also be implemented (Parmar et al., 2015).

7 Bibliography

Aoullay, A. (2018, September 4). *What's WRONG with Metrics?* Medium. https://towardsdatascience.com/choosing-the-right-metric-is-a-huge-issue-99ccbe73de61

Artifacts and Partial-Volume Effects – UTCT – University of Texas. (n.d.). Retrieved 30 May 2020, from https://www.ctlab.geo.utexas.edu/about-ct/artifacts-and-partial-volume-effects/

Bergstra, J., & Bengio, Y. (2012). *Random Search for Hyper-Parameter Optimization*. 25.

Biological basis of radiomcs | eLife. (n.d.). Retrieved 10 May 2020, from https://elifesciences.org/articles/23421

Bogowicz, M., Vuong, D., Huellner, M. W., Pavic, M., Andratschke, N., Gabrys, H. S., Guckenberger, M., & Tanadini-Lang, S. (2019). CT radiomics and PET radiomics: Ready for clinical implementation? *The Quarterly Journal of Nuclear Medicine and Molecular Imaging: Official Publication of the Italian Association of Nuclear Medicine (AIMN) [and] the International Association of Radiopharmacology (IAR), [and] Section of the Society Of...*, *63*(4), 355–370. https://doi.org/10.23736/S1824-4785.19.03192-3

Brown, G., Pocock, A., Zhao, M.-J., & Lujan, M. (2012). *Conditional Likelihood Maximisation: A Unifying Framework for Information Theoretic Feature Selection*. 40.

Brownlee, J. (2014, March 11). Feature Selection to Improve Accuracy and Decrease Training Time. *Machine Learning Mastery*. https://machinelearningmastery.com/feature-selection-to-improve-accuracy-and-decrease-training-time/

Chaddad, A., Toews, M., Desrosiers, C., & Niazi, T. (2019). Deep Radiomic Analysis Based on Modeling Information Flow in Convolutional Neural Networks. *IEEE Access*, *7*, 97242–97252. https://doi.org/10.1109/ACCESS.2019.2930238

Comparing randomized search and grid search—Scikit-learn. (n.d.). Retrieved 8 May 2020, from https://scikit-learn.org/stable/auto_examples/model_selection/plot_randomized_search.html

Computational Radiomics System. (n.d.). Retrieved 8 May 2020, from https://cancerres.aacrjournals.org/content/77/21/e104

CT scan—Mayo Clinic. (n.d.). Retrieved 30 May 2020, from https://www.mayoclinic.org/tests-procedures/ct-scan/about/pac-20393675

Dash, M., & Liu, H. (1997). Feature Selection for Classification. *Intelligent Data Analysis*, *1*, 131–156.

Dataset—Scikit-learn. (n.d.). Retrieved 9 June 2020, from https://scikit-learn.org/stable/datasets/index.html

Decision Trees—Scikit-learn. (n.d.). Retrieved 3 June 2020, from https://scikit-learn.org/stable/modules/tree.html

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. *Machine Learning*, *63*(1), 3–42. https://doi.org/10.1007/s10994-006-6226-1

Gillies, R. J., Kinahan, P. E., & Hricak, H. (2015). Radiomics: Images Are More than Pictures, They Are Data. *Radiology*, *278*(2), 563–577. https://doi.org/10.1148/radiol.2015151169

GraphPad QuickCalcs: T test calculator. (n.d.). Retrieved 14 June 2020, from https://www.graphpad.com/quickcalcs/ttest1/

Griethuysen, J. J. M. van, Fedorov, A., Parmar, C., Hosny, A., Aucoin, N., Narayan, V., Beets-Tan, R. G. H., Fillion-Robin, J.-C., Pieper, S., & Aerts, H. J. W. L. (2017). Computational Radiomics System to Decode the Radiographic Phenotype. *Cancer Research*, *77*(21), e104–e107. https://doi.org/10.1158/0008-5472.CAN-17-0339

Grünbeck, I. A. (2020). *The Effects of MPH-Treatment in ADHD-Diagnosed Children. An Explorative Analysis Using Radiomic Feature*.

Gu, Q., Li, Z., & Han, J. (n.d.). *Generalized Fisher Score for Feature Selection*. 8.

Hua, J., Xiong, Z., Lowey, J., Suh, E., & Dougherty, E. R. (2005). Optimal number of features as a function of sample size for various classification rules. *Bioinformatics*, *21*(8), 1509–1515. https://doi.org/10.1093/bioinformatics/bti171

ImageJ. (n.d.). Retrieved 4 June 2020, from https://imagej.nih.gov/ij/

Lande, I. (2020). *Nuclear forensics for analysing scanning electron microscope images of uranium concentrate ores*.

Langan, A. M. (2020). *MRI-based radiomics analysis for predicting treatment outcome in rectal cancer*.

Langberg, G. S. R. E. (2019). *Searching for Biomarkers of Disease-Free Survival in Head and Neck Cancers Using PET/CT Radiomics*.

LightGBM documentation. (n.d.). Retrieved 2 June 2020, from https://lightgbm.readthedocs.io/en/latest/index.html

Lowekamp, B. C., Chen, D. T., Ibanez, L., & Blezek, D. (2013). The Design of SimpleITK. *Frontiers in Neuroinformatics*, *7*. https://doi.org/10.3389/fninf.2013.00045

Mandot, P. (2018, December 1). *What is LightGBM, How to implement it? How to fine tune the parameters?* Medium. https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-60347819b7fc

Mckinney, B., White, B., Grill, D., Li, P., Kennedy, R., Poland, G., & Oberg, A. (2013). ReliefSeq: A Gene-Wise Adaptive-K Nearest-Neighbor Feature Selection Tool for Finding Gene-Gene Interactions and Main Effects in mRNA-Seq Gene Expression Data. *PloS One*, *8*, e81527. https://doi.org/10.1371/journal.pone.0081527

Midtfjord, A. D. (2018). *Prediction of treatment outcome of head and throat cancer using radiomics of PET/CT images*.

MRI - Mayo Clinic. (n.d.). Retrieved 30 May 2020, from https://www.mayoclinic.org/tests-procedures/mri/about/pac-20384768

Parmar, C., Grossmann, P., Rietveld, D., Rietbergen, M. M., Lambin, P., & Aerts, H. (2015). *Radiomic Machine Learning Classifiers for Prognostic Biomarkers of Head & Neck Cancer*. http://dx.doi.org/10.3389/fonc.2015.00272

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn. *Journal of Machine Learning Research*, *12*(85), 2825–2830.

PET - Mayo Clinic. (n.d.). Retrieved 30 May 2020, from https://www.mayoclinic.org/tests-procedures/pet-scan/about/pac-20385078

Pyradiomics documentation. (n.d.). Retrieved 30 May 2020, from https://pyradiomics.readthedocs.io/en/latest/usage.html

Radiomic Features—Pyradiomics. (n.d.). https://pyradiomics.readthedocs.io/en/latest/features.html

Radiomics: The Process and the Challenges. (n.d.). Retrieved 1 June 2020, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3563280/

RandomizedSearchCV - scikit-learn. (n.d.). Retrieved 10 June 2020, from https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html

Raschka, S., & Mirjalili, V. (2017). *Python Machine Learning*. Packt Publishing Ltd.

Rizzo, S., Botta, F., Raimondi, S., Origgi, D., Fanciullo, C., Morganti, A. G., & Bellomi, M. (2018). Radiomics: The facts and the challenges of image analysis. *European Radiology Experimental*, *2*. https://doi.org/10.1186/s41747-018-0068-z

Sá, J. P. M. de. (2007). *Applied Statistics Using SPSS, STATISTICA, MATLAB and R* (2nd ed.). Springer-Verlag. https://doi.org/10.1007/978-3-540-71972-4

Scikit-learn. (n.d.). Retrieved 30 May 2020, from https://scikit-learn.org/stable/index.html

Shaikh, R. (2018, October 28). *Feature Selection Techniques in Machine Learning with Python*. Medium. https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e

Shung, K. P. (2020, April 10). *Accuracy, Precision, Recall or F1?* Medium. https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

SimpleITK documentation. (n.d.). Retrieved 8 May 2020, from https://simpleitk.readthedocs.io/en/next/Documentation/docs/source/fundamentalConcepts.html

Siu, C. (2020). *Chappers/scikit-feature* [Python]. https://github.com/chappers/scikit-feature (Original work published 2017)

sklearn.linear_model.LogisticRegression—Scikit-learn 0.23.1 documentation. (n.d.). Retrieved 3 June 2020, from https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

Sklearn.svm.SVC — scikit-learn 0.23.1 documentation. (n.d.). Retrieved 2 June 2020, from https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

StandardScaler—Scikit-learn. (n.d.). Retrieved 10 June 2020, from https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

Univariate selection—Diving into data. (n.d.). Retrieved 3 June 2020, from https://blog.datadive.net/selecting-good-features-part-i-univariate-selection/

Urbanowicz, R. J., Meeker, M., LaCava, W., Olson, R. S., & Moore, J. H. (2018). Relief-Based Feature Selection: Introduction and Review. *ArXiv:1711.08421 [Cs, Stat]*. http://arxiv.org/abs/1711.08421

Urbanowicz, R. J., Olson, R. S., Schmitt, P., Meeker, M., & Moore, J. H. (2018). Benchmarking relief-based feature selection methods for bioinformatics data mining. *Journal of Biomedical Informatics*, *85*, 168–188. https://doi.org/10.1016/j.jbi.2018.07.015

Urbanowicz, R. S. O., Pete Schmitt, and Ryan J. (n.d.). *skrebate: Relief-based feature selection algorithms* (Version 0.6) [Python]. Retrieved 3 June 2020, from https://github.com/EpistasisLab/scikit-rebate

Using skrebate—Scikit-rebate. (n.d.). Retrieved 3 June 2020, from https://epistasislab.github.io/scikit-rebate/using/

VarianceThreshold—Scikit-learn. (n.d.). Retrieved 3 June 2020, from https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.VarianceThreshold.html

Welch, M. L., McIntosh, C., Haibe-Kains, B., Milosevic, M. F., Wee, L., Dekker, A., Huang, S. H., Purdie, T. G., O'Sullivan, B., Aerts, H. J. W. L., & Jaffray, D. A. (2019). Vulnerabilities of radiomic signature development: The need for safeguards. *Radiotherapy and Oncology: Journal of the European Society for Therapeutic Radiology and Oncology*, *130*, 2–9. https://doi.org/10.1016/j.radonc.2018.10.027

Appendix A:

Appendix A.1 Head and neck cancer patients' dataset

The total number of patients in the head and neck cancer dataset is 198, pre-treatment and tumour characteristics referred to as clinical factors of the patient cohort (Langberg, 2019).

- Number of samples: 198
- Number of features: 15
- Number of classes: 2
- Class distribution: Disease free survival (DSF) = True (90), DSF = False (108)

Factor Description	Values	Distribution
Age (years)		60 (40,80) *
Gender		
	Male	25%
	Female	75%
Tumour stage		
	T1/T2	48%
	T3/T4	52%
Packs per year		22 (0,128) *
Naxogin (days)		39 (0,45) *
Cisplatin(treatments)		
	0	22%
	1-3	10%
	4-6	68%
Stage		
	0	0.50%
	I	1%
	II	8%
	III	19%
	IV	69%
Degree of spread		
	N0	61%
	N1	24%
	N2	12%

Appendix A:

	N3	4%
Tumour site		
	Oral cavity	9%
	Oropharynx	73%
	Hypopharynx	8%
	Larynx	10%
Tumour volume (square cm)		
		147 (0.8, 285) *
HPV status		
	Positive	42%
	Negative	9%
	Unknown	49%
ICD-10		
	C01	17%
	C02	4%
	C03	0.50%
	C04	1.50%
	C05	2%
	C06	0.50%
	C09	37%
	C10	18%
	C12	3%
	C13	5%
	C32	11%
Histology		
	0	70%
	1	26%
	2	5%
	3	0.50%
ECOG performance status		
	0	65%
	1	33%
	2	2%
Charlson Comorbidity Index		
	0	66%
	1	23%
	2	8%
	3	2%

Appendix A:

	4	2%
	5	0.50%

* median (minimum, maximum)

Appendix A.2 Wine recognition dataset

Wine recognition dataset it is one of the standard datasets available in scikit-learn (*Dataset - Scikit-Learn*, n.d.).

- Number of samples: 178
- Number of features: 13
- Number of classes: 3
- Class distribution: class_0 (59), class_1 (71), class_2 (48)

Summary statistics:

Feature	**Minimum**	**Maximum**	**Mean**	**Standard Deviation**
Alcohol:	11	14.8	13	0.8
Malic Acid:	0.74	5.8	2.34	1.12
Ash:	1.36	3.23	2.36	0.27
Alcalinity of Ash:	10.6	30	19.5	3.3
Magnesium:	70	162	99.7	14.3
Total Phenols:	0.98	3.88	2.29	0.63
Flavanoids:	0.34	5.08	2.03	1
Nonflavanoid Phenols:	0.13	0.66	0.36	0.12
Proanthocyanins:	0.41	3.58	1.59	0.57
Colour Intensity:	1.3	13	5.1	2.3
Hue:	0.48	1.71	0.96	0.23
OD280/OD315 of diluted wines:	1.27	4	2.61	0.71
Proline:	278	1680	746	315

Appendix A.3 Breast cancer Wisconsin (diagnostic) dataset

Breast cancer Wisconsin (diagnostic) dataset it is one of the standard datasets available in scikit-learn (*Dataset - Scikit-Learn*, n.d.).

- Number of samples: 569
- Number of features: 30
- Number of classes: 2
- Class distribution: 212 - Malignant, 357 - Benign

Summary statistics:

Feature	Minimum	Maximum
radius (mean):	6.981	28.11
texture (mean):	9.71	39.28
perimeter (mean):	43.79	188.5
area (mean):	143.5	2501
smoothness (mean):	0.053	0.163
compactness (mean):	0.019	0.345
concavity (mean):	0	0.427
concave points (mean):	0	0.201
symmetry (mean):	0.106	0.304
fractal dimension (mean):	0.05	0.097
radius (standard error):	0.112	2.873
texture (standard error):	0.36	4.885
perimeter (standard error):	0.757	21.98
area (standard error):	6.802	542.2
smoothness (standard error):	0.002	0.031
compactness (standard error):	0.002	0.135

Appendix A:

concavity (standard error):	0	0.396
concave points (standard error):	0	0.053
symmetry (standard error):	0.008	0.079
fractal dimension (standard error):	0.001	0.03
radius (worst):	7.93	36.04
texture (worst):	12.02	49.54
perimeter (worst):	50.41	251.2
area (worst):	185.2	4254
smoothness (worst):	0.071	0.223
compactness (worst):	0.027	1.058
concavity (worst):	0	1.252
concave points (worst):	0	0.291
symmetry (worst):	0.156	0.664
fractal dimension (worst):	0.055	0.208

Appendix B: Biorad installations and use instructions

Requirements:

- Install Anaconda version 3.7 or above from: https://www.anaconda.com/distribution/

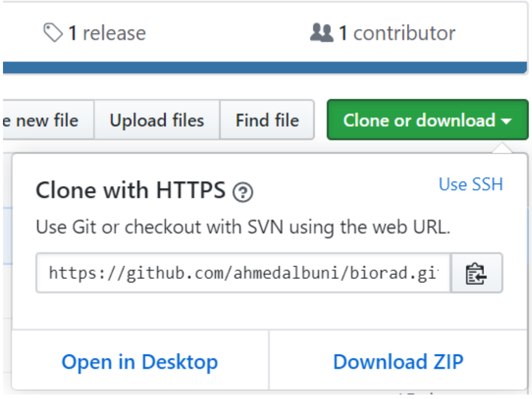
Mac users will need to install Homebrew, instruction can be found here: https://brew.sh/. Then they need to install libomp in the terminal window, which is a non-python dependency. Libomp provides OpenMP bindings to llvm, which is used by parallel numba code and the clang compiler.

installation command: install libomp

Biorad project:

The Biorad project is available on GitHub in the following location: https://github.com/ahmedalbuni/biorad

- Download or clone the code to the local machine:



Clone with HTTPS

Use Git or checkout with SVN using the web URL.

```
https://github.com/ahmedalbuni/biorad.gi
```

- Open Anaconda prompt on Windows or the command line in macOS, navigate to the directory (inside the biorad folder) where you placed the code on your local machine, and type the following command to install the project requirements:
  pip install -r requirements.txt

  If the user is not familiar with the command line window, the user can change the current working directory by using this command ***cd c:\newpath*** Go through this quick tutorial for more information: https://www.digitalcitizen.life/command-prompt-how-use-basic-commands

After installing the project requirements, the user should be able to run both the classifications and the features extraction tools.

For features extraction, use the command prompt and navigate to the following folder:

Appendix B: Biorad installations and use instructions

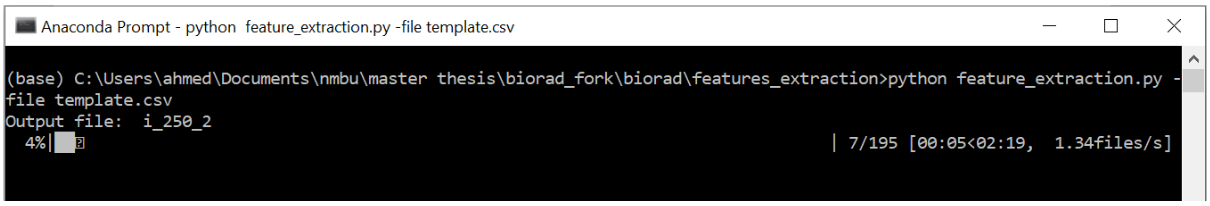
biorad\features_extraction

Modify the template.csv file

image_dir	mask_dir	output_file_name	bin_width	shape	first_order	glszm	glrlm	ngtdm	gldm	glcm
C:\tmp\250\	C:\tmp\250\m\	i_250_2	25		1	1	1		1	1
C:\tmp\500\	C:\tmp\500\m\	i_500_2	25	1	1	1	1		1	1

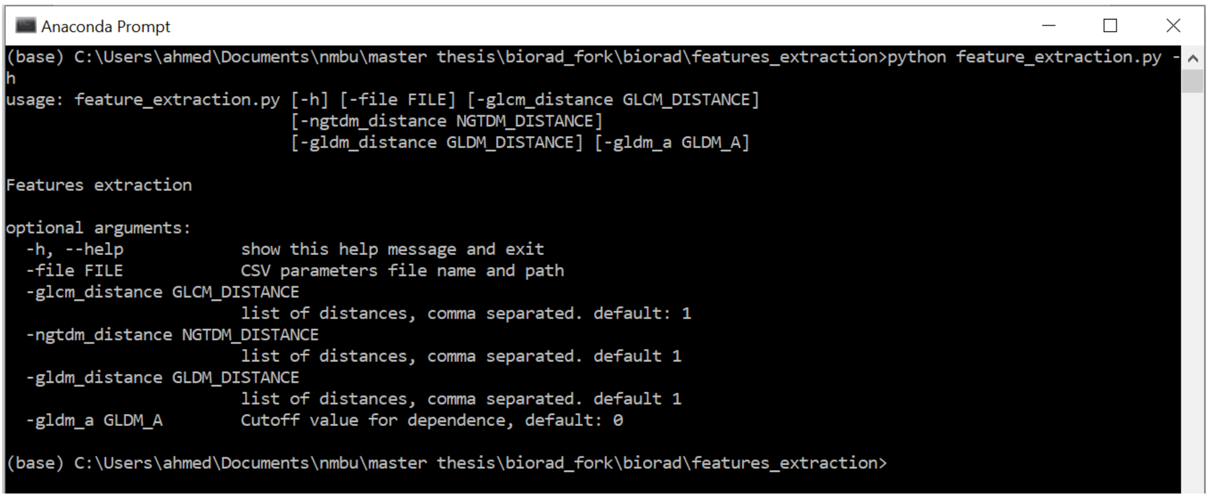
- Modify image_dir to the list of directories of the images, and the mask_dir to the locations of the masks. The names of the masks should match precisely the image names. If the mask is not provided, a dummy mask that covers the whole image will be automatically generated, but the shape features will not be applicable in that case.
- The output file where the results are stored; if the user did not specify the full path, it will be stored at the current working directory.
- The bin_width, the default value is 25; each bin represents specific greyscale intensity values; the user can modify this value based on the needs.
- At the end of the CSV file, there is a list of radiomics features categories, the user should write '1' for the category features to be extracted.

- Write the following command in the command prompt to run the tool:
  python feature_extraction.py -file template.csv


Anaconda Prompt - python feature_extraction.py -file template.csv

```
(base) C:\Users\ahmed\Documents\nmbu\master thesis\biorad_fork\biorad\features_extraction>python feature_extraction.py -
file template.csv
Output file:  i_250_2
  4%|█▏                                                                          | 7/195 [00:05<02:19,  1.34files/s]
```

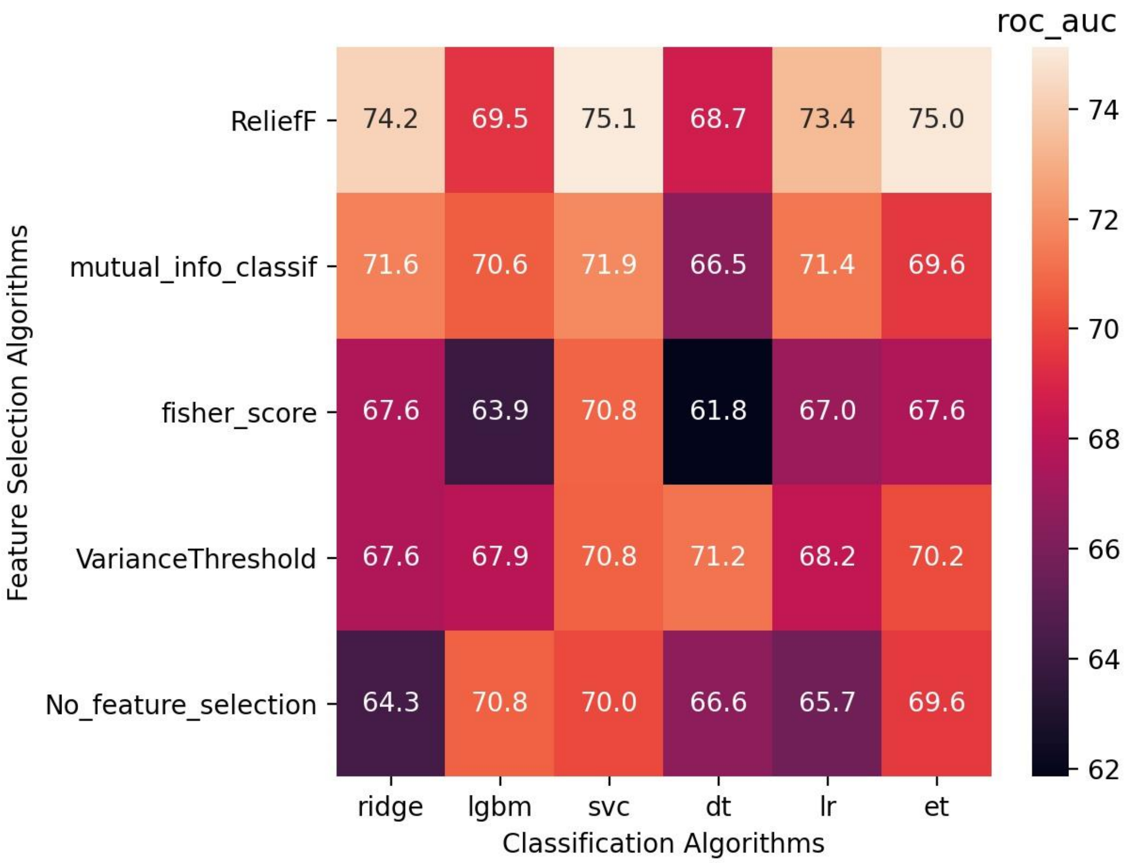
- Additional parameters can be provided for advanced settings:



```
(base) C:\Users\ahmed\Documents\nmbu\master thesis\biorad_fork\biorad\features_extraction>python feature_extraction.py -
h
usage: feature_extraction.py [-h] [-file FILE] [-glcm_distance GLCM_DISTANCE]
                             [-ngtdm_distance NGTDM_DISTANCE]
                             [-gldm_distance GLDM_DISTANCE] [-gldm_a GLDM_A]

Features extraction

optional arguments:
  -h, --help            show this help message and exit
  -file FILE            CSV parameters file name and path
  -glcm_distance GLCM_DISTANCE
                        list of distances, comma separated. default: 1
  -ngtdm_distance NGTDM_DISTANCE
                        list of distances, comma separated. default 1
  -gldm_distance GLDM_DISTANCE
                        list of distances, comma separated. default 1
  -gldm_a GLDM_A        Cutoff value for dependence, default: 0

(base) C:\Users\ahmed\Documents\nmbu\master thesis\biorad_fork\biorad\features_extraction>
```

Features selection and classifications:

This tool tests random combinations of hyperparameters specified in a JSON file, and provide the user with the following heatmap, which can help in selecting the optimal features selector and classifier for the problem:



To run the tool, the user will need to provide the dataset in a CSV file, where the response variable is the last field. All data should be numerical, with no missing information. This tool supports both binary and multiclass classification problem, but the correct scoring should be selected. For binary classification the following are supported:

- roc_auc
- accuracy
- f1
- precision
- recall

And for multiclass classification:

- accuracy
- f1_micro
- f1_macro
- f1_weighted
- precision_micro
- precision_macro

- precision_weighted
- recall_micro
- recall_macro
- recall_weighted

Also, users will need a JSON file with the configurations, config.json, and under the biorad directory, a sample file is given. Users can modify it to select the range of selected features, regularisation parameters range and others.

To modify a JSON file the user can use any text editor, that can be done by right click on the file, open with, then select notepad.

In the JSON file, the user will need to modify the path of the dataset file as follows:

"features_file": "c:\\tmp\\hn_ct_c.csv", remember to use the escape character "\" in the path, which means you should replace all single backslash characters with double backslashes, and do not forget the file extension ".csv"

Also, the user needs to update the output directory, where the results are stored.

- In command prompt navigate to the biorad directory
- Run the following command:
  python main.py -file config.json

In addition to the heatmap, the tool will provide CSV files with the details of all the random experiments. The location of the CSV file is provided in the configuration JSON file.

Appendix C: Pyradiomics features

For more information about this table, refer to the pyradiomics website (*Pyradiomics Documentation*, n.d.).

- *X* is the set of *Np* Voxels in the ROI area defined by the mask
- *P(i)* is the histogram of *Ng* unique intensity values
- *V is the Volume* of the mesh in mm^3
- *A* is the Surface of the mesh in mm^3
- *HX* = $-\sum_{j=1}^{N_g} p_x(i)\log_2(p_x(i)+\epsilon)$
- *HY* = $-\sum_{j=1}^{N_g} p_y(i)\log_2\left(p_y(i)+\epsilon\right)$
- *−HXY+HXY1*= $\sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p(i,j)\log_2(p(i,j)) - \sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p(i,j)\log_2(p_x(i)p_y(j))$

Feature	**Formula**
First-order features	
1. Energy	$\sum_{i=1}^{N_p}(X(i)+c)^2$
2. Total energy	$V_{voxel}\cdot\sum_{i=1}^{N_p}(X(i)+c)^2$
3. Entropy	$\sum_{i=1}^{N_g} p(i)log_2\,(p(i)+\epsilon)$
4. Minimum	$Min(X)$
5. 10th percentile	
6. 90th percentile	
7. Maximum	$Max(X)$
8. Mean	$\frac{1}{N_p}\sum_{i=1}^{N_p}X(i)$
9. Median	
10. Interquartile Range	$P^{75}-P^{25}$
11. Range	$Max(X)-Min(X)$
12. Mean Absolute Deviation (MAD)	$\frac{1}{N_p}\sum_{i=1}^{N_p}\lvert X(i)-\bar{X}\rvert$
13. Robust Mean Absolute Deviation (rMAD)	$\frac{1}{N_{10-90}}\sum_{i=1}^{N_p}\lvert X_{10-90}(i)-\bar{X}_{10-90}\rvert$
14. Root Mean Squared (RMS)	$\sqrt{\frac{1}{N_p}\sum_{i=1}^{N_p}(\mathrm{X}(i)+c)^2}$

15. Standard Deviation	$\sqrt{\frac{1}{N_p}\sum_{i=1}^{N_p}(\mathrm{X}(i)-\overline{X})^2}$
16. Skewness	$\frac{\mu_3}{\sigma^3}=\frac{\frac{1}{N_p}\sum_{i=1}^{N_p}(\mathrm{X}(i)-\overline{X})^3}{\left(\sqrt{\frac{1}{N_p}\sum_{i=1}^{N_p}(\mathrm{X}(i)-\overline{X})^2}\right)^3}$
17. Kurtosis	$\frac{\mu_4}{\sigma^4}=\frac{\frac{1}{N_p}\sum_{i=1}^{N_p}(\mathrm{X}(i)-\overline{X})^4}{\left(\frac{1}{N_p}\sum_{i=1}^{N_p}(\mathrm{X}(i)-\overline{X})^2\right)^2}$
18. Variance	$\frac{1}{N_p}\sum_{i=1}^{N_p}(\mathrm{X}(i)-\overline{X})^2$
19. Uniformity	$\sum_{i=1}^{N_g}p(i)^2$
Shape Features (3D)	
1. Mesh Volume	$V_i=\frac{Oa_i\cdot(Ob_i\times Oc_i)}{6}$ $V=\sum_{i=1}^{N_f}V_i$
2. Voxel Volume	$\sum_{k=1}^{N_v}V_k$
3. Surface Area	$A_i=\frac{1}{2}\lvert\mathrm{a}_i\mathrm{b}_i\times\mathrm{a}_i\mathrm{c}_i\rvert$ $A=\sum_{i=1}^{N_f}A_i$
4. Surface Area to Volume ratio	$\frac{A}{V}$
5. Sphericity	$\frac{\sqrt[3]{36\pi V^2}}{A}$
6. Compactness 1	$\frac{V}{\sqrt{\pi A^3}}$
7. Compactness 2	$36\pi\frac{V^2}{A^3}$
8. Spherical Disproportion	$\frac{A}{\sqrt[3]{36\pi V^2}}$
9. Maximum 3D diameter	
10. Maximum 2D diameter (Slice)	
11. Maximum 2D diameter (Column)	

12. Maximum 2D diameter (Row)	
13. Major Axis Length	$4\sqrt{\lambda_{major}}$
14. Minor Axis Length	$4\sqrt{\lambda_{minor}}$
15. Least Axis Length	$4\sqrt{\lambda_{least}}$
16. Elongation	$\sqrt{\frac{\lambda_{minor}}{\lambda_{major}}}$
17. Flatness	$\sqrt{\frac{\lambda_{least}}{\lambda_{major}}}$
Shape Features (2D)	
1. Mesh Surface	$\frac{1}{2}\text{Oa}_i \times \text{Ob}_i$ $\sum_{i=1}^{N_f} A_i$
2. Pixel Surface	$\sum_{k=1}^{N_v} A_k$
3. Perimeter	$P_i = \sqrt{(\text{a}_i - \text{b}_i)^2}$ $P = \sum_{i=1}^{N_f} P_i$
4. Perimeter to Surface ratio	$\frac{P}{A}$
5. Sphericity	$\frac{2\sqrt{\pi A}}{P}$
6. Spherical Disproportion	$\frac{P}{2\sqrt{\pi A}}$
7. Maximum 2D diameter	
8. Major Axis Length	$4\sqrt{\lambda_{major}}$
9. Minor Axis Length	$4\sqrt{\lambda_{minor}}$
10. Elongation	$\sqrt{\frac{\lambda_{minor}}{\lambda_{major}}}$
Grey Level Co-occurrence Matrix (GLCM) Features	
1. Autocorrelation	$\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j)ij$
2. Joint Average	$\sum_{i=1}^{N_g} \sum_{j=1}^{N_g} p(i,j)i$

3. Cluster Prominence	$\sum_{i=1}^{N_g}\sum_{j=1}^{N_g}(i+j-\mu_x-\mu_y)^4 p(i,j)$
4. Cluster Shade	$\sum_{i=1}^{N_g}\sum_{j=1}^{N_g}(i+j-\mu_x-\mu_y)^3 p(i,j)$
5. Cluster Tendency	$\sum_{i=1}^{N_g}\sum_{j=1}^{N_g}(i+j-\mu_x-\mu_y)^2 p(i,j)$
6. Contrast	$\sum_{i=1}^{N_g}\sum_{j=1}^{N_g}(i-j)^2 p(i,j)$
7. Correlation	$\frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p(i,j)ij-\mu_x\mu_y}{\sigma_x(i)\sigma_y(j)}$
8. Difference Average	$\sum_{k=0}^{N_g-1} kp_{x-y}(k)$
9. Difference Entropy	$\sum_{k=0}^{N_g-1} p_{x-y}(k)\log_2(p_{x-y}(k)+\epsilon)$
10. Difference Variance	$\sum_{k=0}^{N_g-1} (k-DA)^2 p_{x-y}(k)$
11. Joint Energy	$\sum_{i=1}^{N_g}\sum_{j=1}^{N_g}(p(i,j))^2$
12. Joint Entropy	$-\sum_{i=1}^{N_g}\sum_{j=1}^{N_g} p(i,j)\log_2(p(i,j)+\epsilon)$
13. Informational Measure of Correlation (IMC) 1	$\frac{HXY-HXY1}{\max\{HX,HY\}}$
14. Informational Measure of Correlation (IMC) 2	$\sqrt{1-e^{-2(HXY2-HXY)}}$

15. Inverse Difference Moment (IDM)	$\sum_{k=0}^{N_g-1}\frac{p_{x-y}(k)}{1+k^2}$
16. Maximal Correlation Coefficient (MCC)	$= \sqrt{\text{second largest eigenvalue of Q}}$ $Q(i,j) = \sum_{k=0}^{N_g}\frac{p(i,k)p(j,k)}{p_x(i)p_y(k)}$
17. Inverse Difference Moment Normalized (IDMN)	$\sum_{k=0}^{N_g-1}\frac{p_{x-y}(k)}{1+\left(\frac{k^2}{N_g^2}\right)}$
18. Inverse Difference (ID)	$\sum_{k=0}^{N_g-1}\frac{p_{x-y}(k)}{1+k}$
19. Inverse Difference Normalized (IDN)	$\sum_{k=0}^{N_g-1}\frac{p_{x-y}(k)}{1+\left(\frac{k}{N_g}\right)}$
20. Inverse Variance	$\sum_{k=1}^{N_g-1}\frac{p_{x-y}(k)}{k^2}$
21. Maximum Probability	$\max(p(i,j))$
22. Sum Average	$\sum_{k=2}^{2N_g}p_{x+y}(k)k$
23. Sum Entropy	$\sum_{k=2}^{2N_g}p_{x+y}(k)\log_2(p_{x+y}(k)+\epsilon)$
24. Sum of Squares	$\sum_{i=1}^{N_g}\sum_{j=1}^{N_g}(i-\mu_x)^2p(i,j)$
Grey Level Run Length Matrix (GLRLM) Features	
1. Short Run Emphasis (SRE)	$\frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}\frac{\mathrm{P}(i,j\mid\theta)}{j^2}}{N_r(\theta)}$
2. Long Run Emphasis (LRE)	$\frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}\mathrm{P}(i,j\mid\theta)j^2}{N_r(\theta)}$
3. Grey Level Non-Uniformity (GLN)	$\frac{\sum_{i=1}^{N_g}\left(\sum_{j=1}^{N_r}\mathrm{P}(i,j\mid\theta)\right)^2}{N_r(\theta)}$

4. Grey Level Non-Uniformity Normalized (GLNN)	$\frac{\sum_{i=1}^{N_g}\left(\sum_{j=1}^{N_r}\mathrm{P}(i,j\mid\theta)\right)^2}{N_r(\theta)^2}$
5. Run Length Non-Uniformity (RLN)	$\frac{\sum_{j=1}^{N_r}\left(\sum_{i=1}^{N_g}\mathrm{P}(i,j\mid\theta)\right)^2}{N_r(\theta)}$
6. Run Length Non-Uniformity Normalized (RLNN)	$\frac{\sum_{j=1}^{N_r}\left(\sum_{i=1}^{N_g}\mathrm{P}(i,j\mid\theta)\right)^2}{N_r(\theta)^2}$
7. Run Percentage (RP)	$\frac{N_r(\theta)}{N_p}$
8. Grey Level Variance (GLV)	$\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}p(i,j\mid\theta)(i-\mu)^2$
9. Run Variance (RV)	$\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}p(i,j\mid\theta)(j-\mu)^2$
10. Run Entropy (RE)	$-\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}p(i,j\mid\theta)\log_2(p(i,j\mid\theta)+\epsilon)$
11. Low Grey Level Run Emphasis (LGLRE)	$\frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}\frac{\mathrm{P}(i,j\mid\theta)}{i^2}}{N_r(\theta)}$
12. High Grey Level Run Emphasis (HGLRE)	$\frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}\mathrm{P}(i,j\mid\theta)i^2}{N_r(\theta)}$
13. Short Run Low Grey Level Emphasis (SRLGLE)	$\frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}\frac{\mathrm{P}(i,j\mid\theta)}{i^2j^2}}{N_r(\theta)}$
14. Short Run High Grey Level Emphasis (SRHGLE)	$\frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}\frac{\mathrm{P}(i,j\mid\theta)i^2}{j^2}}{N_r(\theta)}$
15. Long Run Low Grey Level Emphasis (LRLGLE)	$\frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_r}\frac{\mathrm{P}(i,j\mid\theta)j^2}{i^2}}{N_r(\theta)}$

16. Long Run High Grey Level Emphasis (LRHGLE)	$\frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_r} \mathrm{P}(i,j\mid\theta)i^2j^2}{N_r(\theta)}$
Grey Level Size Zone Matrix (GLSZM) Features	
1. Small Area Emphasis (SAE)	$\frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_s} \frac{\mathrm{P}(i,j)}{j^2}}{N_z}$
2. Large Area Emphasis (LAE)	$\frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_s} \mathrm{P}(i,j)j^2}{N_z}$
3. Grey Level Non-Uniformity (GLN)	$\frac{\sum_{i=1}^{N_g} \left(\sum_{j=1}^{N_s} \mathrm{P}(i,j)\right)^2}{N_z}$
4. Grey Level Non-Uniformity Normalized (GLNN)	$\frac{\sum_{i=1}^{N_g} \left(\sum_{j=1}^{N_s} \mathrm{P}(i,j)\right)^2}{N_z^2}$
5. Size-Zone Non-Uniformity (SZN)	$\frac{\sum_{j=1}^{N_s} \left(\sum_{i=1}^{N_g} \mathrm{P}(i,j)\right)^2}{N_z}$
6. Size-Zone Non-Uniformity Normalized (SZNN)	$\frac{\sum_{j=1}^{N_s} \left(\sum_{i=1}^{N_g} \mathrm{P}(i,j)\right)^2}{N_z^2}$
7. Zone Percentage (ZP)	$\frac{N_z}{N_p}$
8. Grey Level Variance (GLV)	$\sum_{i=1}^{N_g} \sum_{j=1}^{N_s} p(i,j)(i-\mu)^2$
9. Zone Variance (ZV)	$\sum_{i=1}^{N_g} \sum_{j=1}^{N_s} p(i,j)(j-\mu)^2$
10. Zone Entropy (ZE)	$-\sum_{i=1}^{N_g} \sum_{j=1}^{N_s} p(i,j)\log_2(p(i,j)+\epsilon)$
11. Low Grey Level Zone Emphasis (LGLZE)	$\frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_s} \frac{\mathrm{P}(i,j)}{i^2}}{N_z}$
12. High Grey Level Zone Emphasis (HGLZE)	$\frac{\sum_{i=1}^{N_g} \sum_{j=1}^{N_s} \mathrm{P}(i,j)i^2}{N_z}$

13. Small Area Low Grey Level Emphasis (SALGLE)	$\frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}\frac{\mathrm{P}(i,j)}{i^2j^2}}{N_z}$
14. Small Area High Grey Level Emphasis (SAHGLE)	$\frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}\frac{\mathrm{P}(i,j)i^2}{j^2}}{N_z}$
15. Large Area Low Grey Level Emphasis (LALGLE)	$\frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}\frac{\mathrm{P}(i,j)j^2}{i^2}}{N_z}$
16. Large Area High Grey Level Emphasis (LAHGLE)	$\frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_s}\mathrm{P}(i,j)i^2j^2}{N_z}$
Neighbouring Grey Tone Difference Matrix (NGTDM) Features	
1. Coarseness	$\frac{1}{\sum_{i=1}^{N_g}p_is_i}$
2. Contrast	$\left(\frac{1}{N_{g,p}(N_{g,p}-1)}\sum_{i=1}^{N_g}\sum_{j=1}^{N_g}p_ip_j(i-j)^2\right)\left(\frac{1}{N_{v,p}}\sum_{i=1}^{N_g}s_i\right)$
3. Busyness	$\frac{\sum_{i=1}^{N_g}p_is_i}{\sum_{i=1}^{N_g}\sum_{j=1}^{N_g}\lvert ip_i-jp_j\rvert}$
4. Complexity	$\frac{1}{N_{v,p}}\sum_{i=1}^{N_g}\sum_{j=1}^{N_g}\lvert i-j\rvert\frac{p_is_i+p_js_j}{p_i+p_j}$
5. Strength	$\frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_g}(p_i+p_j)(i-j)^2}{\sum_{i=1}^{N_g}s_i}$
Grey Level Dependence Matrix (GLDM) Features	
1. Small Dependence Emphasis (SDE)	$\frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_d}\frac{\mathrm{P}(i,j)}{i^2}}{N_z}$
2. Large Dependence Emphasis (LDE)	$\frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_d}\mathrm{P}(i,j)j^2}{N_z}$

3. Grey Level Non-Uniformity (GLN)	$\frac{\sum_{i=1}^{N_g}\left(\sum_{j=1}^{N_d}\mathrm{P}(i,j)\right)^2}{N_z}$
4. Dependence Non-Uniformity (DN)	$\frac{\sum_{j=1}^{N_d}\left(\sum_{i=1}^{N_g}\mathrm{P}(i,j)\right)^2}{N_z}$
5. Dependence Non-Uniformity Normalized (DNN)	$\frac{\sum_{j=1}^{N_d}\left(\sum_{i=1}^{N_g}\mathrm{P}(i,j)\right)^2}{N_z^2}$
6. Grey Level Variance (GLV)	$\sum_{i=1}^{N_g}\sum_{j=1}^{N_d}p(i,j)(i-\mu)^2\text{ , where}\mu=\sum_{i=1}^{N_g}\sum_{j=1}^{N_d}ip(i,j)$
7. Dependence Variance (DV)	$\sum_{i=1}^{N_g}\sum_{j=1}^{N_d}p(i,j)(j-\mu)^2\text{ , where}\mu=\sum_{i=1}^{N_g}\sum_{j=1}^{N_d}jp(i,j)$
8. Dependence Entropy (DE)	$-\sum_{i=1}^{N_g}\sum_{j=1}^{N_d}p(i,j)\log_2(p(i,j)+\epsilon)$
9. Low Grey Level Emphasis (LGLE)	$\frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_d}\frac{\mathrm{P}(i,j)}{i^2}}{N_z}$
10. High Grey Level Emphasis (HGLE)	$\frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_d}\mathrm{P}(i,j)i^2}{N_z}$
11. Small Dependence Low Grey Level Emphasis (SDLGLE)	$\frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_d}\frac{\mathrm{P}(i,j)}{i^2j^2}}{N_z}$
12. Small Dependence High Grey Level Emphasis (SDHGLE)	
13. Large Dependence Low Grey Level Emphasis (LDLGLE)	$\frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_d}\frac{\mathrm{P}(i,j)j^2}{i^2}}{N_z}$
14. Large Dependence High Grey Level Emphasis (LDHGLE)	$\frac{\sum_{i=1}^{N_g}\sum_{j=1}^{N_d}\mathrm{P}(i,j)i^2j^2}{N_z}$

Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway