Model Selection for Clustering

By

Pritha Sarkar

19th December, 2021

Contents

Chapter 1 INTRODUCTION

Cluster Analysis, or clustering, is the task of grouping a set of objects in such a way that objects in the same group (called a **cluster**) are more similar (in some sense) to each other than to those in other groups (clusters) [1]. It is a main task of exploratory data analysis, and a common technique for statistical data analysis, used in many fields, including pattern recognition, image analysis, information retrieval, bioinformatics, data compression, computer graphics and machine learning [1].

1.1 Aim and Objective

The aim of the case-study is to select the prime model to cluster the representations extracted from colorectal tissue patches.

Objectively, we go through 4 representations, namely PathologyGAN, ResNet50, InceptionV3 and VGG16 (each containing 3 features – tissue type, 5000x100 PCA vector and 5000x100 UMAP vector). Each representation is tested with two clustering algorithms (K-means Clustering and Hierarchical Clustering) to group together the 9 different tissue classes. Moreover, the cluster quality is reported according to both intrinsic and extrinsic measures.

1.2 Outline

The report provides an elaborate explanation of the problem statement, the goal to achieve and some background details, i.e., data provided for the case study in chapter 1 and 2. Chapter 3 dives into the algorithms used for clustering and the methods used to optimise the output from the clustering methods applied. Chapter 4 outlines the results of each experiment. By observing the data provided in Chapter 4, we try to establish the optimal model for cluster analysis on the given dataset in Chapter 5 through analytical reasoning. Finally, we conclude by reporting the overall finding.

Chapter 2 BACKGROUND INFORMATION

Tissue biopsy is obtained from microscope with high resolution, which resulted in 498 megabyte per slide on average. Hence, it became necessary to break them into smaller patches for the sake of analysis across many patients. Clustering on tissue patches will help tissue examiners understand the whole slide by giving statistical summary of visual features.

The particulars of the dataset provided are mentioned in the following section.

The original collected dataset consists of 5000 colorectal cancer tissue patches. The tissue patches are of 9 different types- Adipose (ADI), Background (BACK), Debris (DEB), Lymphocytes (LYM), Mucus (MUC), Smooth Muscle (MUS), Normal Colon Mucosa (NORM), Cancer-associated stroma (STR) and Colorectal adenocarcinoma epithelium (TUM).

The dataset has been represented in 4 different ways. Three of them are extracted from popular CNN classifiers, trained on ImageNet dataset and achieve 74.9%, 77.9%, 71.3% accuracy. They are ResNet50, InceptionV3 and VGG16, respectively. The remaining representation is a state-of-the-art GAN-based model for tissue images, trained on an unsupervised manner.

Two dimensionality reduction methods were also employed to reduce each representation from their original vector size to 100.

The PCA vector represents the first 100 principal components with highest variance, whereas, the UMAP vector is the 100 umap components obtained based on dataset structures.

5 H5PY files were provided for the case study. pge_dim_reduced_feature.h5 contained data of PathologyGAN representation. resnet50_dim_reduced_feature.h5 contained data from ResNet50 representation. inceptionv3_dim_reduced_feature.h5 contained data from InceptionV3 representation. Finally, vgg16_dim_reduced_feature.h5 contained data from VGG16 representation. Each file had three attributes. The first attribute pertains to the type of tissue patch, the second attribute is the PCA vector and the third attribute is the UMAP vector.

Chapter 3 METHODS

This chapter takes a look into the clustering methods applied on the tissue patch representations as well as the performance measurement techniques deployed to check the quality of each cluster.

3.1 K-Means Clustering

An unsupervised learning algorithm, k-means clustering is a method of vector quantization that aims to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the nearest *mean* (cluster centers or cluster centroid), serving as a prototype of the cluster [2].

Given a set of observations (**$\mathbf{x}_1$, $\mathbf{x}_2$, …., $\mathbf{x}_n$**), where each observation is a *d*-dimensional real vector, *k*-means clustering aims to partition the *n* observations into *k (≤ n)* sets **S = {**$S_1, S_2, \ldots, S_k$**}** so as to minimize the within-cluster sum of squares (WCSS) (i.e., variance) [2].

Formally, the objective is to find:

$$\underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{k} \sum_{\mathbf{x}\in S_i} \left\| \mathbf{x} - \boldsymbol{\mu}_i \right\|^2 = \underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{k} |S_i| \operatorname{Var} S_i$$

where $\boldsymbol{\mu}_i$ is the mean of points in S_i. This is equivalent to minimizing the pairwise squared deviations of points in the same cluster:

$$\underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{k} \frac{1}{2|S_i|} \sum_{\mathbf{x},\mathbf{y}\in S_i} \left\| \mathbf{x} - \mathbf{y} \right\|^2$$

The equivalence can be deduced from identity

$$\sum_{\mathbf{x}\in S_i} \left\| \mathbf{x} - \boldsymbol{\mu}_i \right\|^2 = \sum_{\mathbf{x}\neq\mathbf{y}\in S_i} (\mathbf{x} - \boldsymbol{\mu}_i)^T (\boldsymbol{\mu}_i - \mathbf{y}).$$

Because the total variance is constant, this is equivalent to maximizing the sum of squared deviations between points in *different* clusters (between-cluster sum of squares, BCSS) [2] [3].

The most common algorithm uses an iterative refinement technique.

Given an initial set of *k* means $m_1^{(1)}$, …., $m_k^{(1)}$ (see below), the algorithm proceeds by alternating between two steps:

Assignment step: Assign each observation into cluster with the nearest mean: that with the least squared Euclidean distance [4]. Mathematically, this means partitioning the observations according to the Voronoi diagram [5] generated by the means [2].

$$S_i^{(t)} = \left\{ x_p : \left\| x_p - m_i^{(t)} \right\|^2 \leq \left\| x_p - m_j^{(t)} \right\|^2 \ \forall j, 1 \leq j \leq k \right\},$$

where each x_p is assigned to exactly one $\boldsymbol{S}^{(t)}$, even if it could be assigned to two or more of them [2].

Update step: Recalculate means (centroids) for observations assigned to each cluster [2].

$$m_i^{(t+1)} = \frac{1}{\left|S_i^{(t)}\right|} \sum_{x_j \in S_i^{(t)}} x_j$$

The algorithm has converged when the assignments no longer change. The algorithm is not guaranteed to find the optimum [2].

The algorithm is often presented as assigning objects to the nearest cluster by distance. Using a different distance function other than (squared) Euclidean distance may prevent the algorithm from converging [2].

For our case-study, we iterated over the data-set 11 times with *k= {2, 3, ..., 12}*. sklearn.cluster.KMeans module is used to achieve this. Each iteration was checked for their quality through Silhouette Score[1] and V-measure Score and the optimal number of clusters is chosen to apply thereafter.

3.2 Hierarchical Clustering

Hierarchical clustering is a method of cluster analysis which seeks to build a hierarchy of clusters [6]. Hierarchical clustering uses agglomerative or divisive techniques to perform clustering [7].

Agglomerative, is a bottom-up approach where each representation starts in its own cluster, and pairs of clusters are merged as one moves up the hierarchy [6].

Divisive, is a top-down approach where all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy [6].

In general, the merges and splits are determined in a greedy manner and the results are presented in a dendrogram[2].

Clusters usually have multiple points in them that require a different approach for the distance matrix calculation. Linkage decides how the distance between

[1] Yellowbrick, a Machine Learning Visualization package, is used to display the Silhouette Score using plots. Please install yellowbrick before executing the scripts. Use the command: pip install yellowbrick --user

[2] A dendrogram is a diagram representing a tree.

clusters, or point to cluster distance is computed. The commonly used linkage mechanisms are – Single Linkage, Average Linkage, Complete Linkage, Ward Linkage and Centroid Linkage. The formulas for distance calculation are mentioned below.

Single Linkage

$$\min\{d(a,b): a \in A, b \in B\}$$

Average Linkage

$$\frac{1}{|A|.|B|} \sum_{a\in A} \sum_{b \in B} d(a,b)$$

Complete Linkage

$$\max\{d(a,b): a \in A, b \in B\}$$

Ward Linkage

$$d(i \cup j, k) = \frac{d(i,k)+d(j,k)}{2}$$

Centroid Linkage

$$||c_s - c_t||$$

$where\ c_s\ and\ c_t\ are\ the\ centroids\ of\ clusters\ \boldsymbol{s}\ and\ \boldsymbol{t}, respectively.$

Distance between two or more clusters can be calculated using multiple approaches, the most popular being Euclidean Distance. There are no statistical techniques to decide the number of clusters in hierarchical clustering, unlike a K Means algorithm that uses an elbow plot to determine the number of clusters. However, one common approach is to analyse the dendrogram and look for groups that combine at a higher dendrogram distance [7].

For our case-study, we decided to implement Ward Linkage and Centroid Linkage to create dendrograms and use the perceivable number of colours as the number of clusters to display the dataset by the means of a scatter plot. sklearn.cluster.Agglomerativeclustering module was used to achieve this.

3.3 Silhouette Score

Silhouette score is used to evaluate the quality of clusters created using clustering algorithms such as K-Means in terms of how well samples are clustered with other samples that are similar to each other. The Silhouette score is calculated for each sample of different clusters. In order to calculate the Silhouette score for each observation/ data point, the following distances need to be found out for each observation belonging to all the clusters:

Mean distance between the observation and all other data points in the same cluster, also known as, **mean intra-cluster distance**.

Mean distance between the observation and all other data points of the next nearest cluster, also known as, **mean nearest-cluster distance**.

Silhouette score, **S**, for each sample is calculated using the following formula:

$$\boldsymbol{S} = \frac{(b-a)}{\max(a,b)}\,, \quad where \quad a = mean\ intra - cluster\ distance$$

$$b = mean\ nearest - cluster\ distance$$

The value of Silhouette score varies from -1 to 1. If the score is 1, the cluster is dense and well- separated than other clusters. A value near 0 represents overlapping clusters with samples very close to the decision boundary of the neighbouring clusters. A negative score (-1, 0) indicate that the samples might have got assigned to the wrong clusters [8].

sklearn.metrics.silhouette_score and sklearn.metrics.silhouette_samples were deployed to calculate the silhouette score. Moreover, the plots were represented using Yellowbrick: Machine Learning Visualization package.

For our case-study, Silhouette score has been used as an intrinsic measure for evaluation of clusters.

3.4 V-Measure Score

Given the knowledge of the ground truth class assignments of the samples, it is possible to define some intuitive metric using conditional entropy analysis. Rosenberg and Hirschberg [9] define the following two desirable objectives for any cluster assignment.

Homogeneity: Each cluster contains only members of a single class.

Completeness: All members of a given class are assigned to the same cluster.

Both the metrics are bounded below by 0.0 and above by 1.0. It is mentionable that a higher score is better [10].

sklearn.metrics.v_measure_score was developed to calculate the V-measure cluster labelling given a ground truth.

For our case-study, V-measure score has been used as an extrinsic measure for evaluation of clusters.

Chapter 4 RESULTS

This chapter states the results of the experimental setup and results gained from it in detailed manner.

4.1 K-Means Clustering[3]

This sub-section concerns itself with displaying the results achieved when K-Means Clustering Analysis was deployed on the 4 different representations of the colorectal tissue patches. As mentioned earlier in the report, each representation has two different vectors- PCA and UMAP. The clustering analysis was implemented on each of the vectors. Thus, each sub-sub-section will contain 2 scatter plots- one derived from the PCA vectors and one derived from the UMAP vectors present in the files.

The tables displaying the Silhouette Score and V-Measure Score when number of clusters, $k = (2, 3, ..., 11)$ and the line charts used to reach a decision of how many clusters would serve as the near optimal number of clusters can be found in the Appendices section of the report.

4.1.1 PathologyGAN

4.1.1.A. PathologyGAN PCA

From the Line Chart displayed in Figure A.1 in the Appendices section, it can be observed that *k=10* would be able to provide a good representation of clusters.

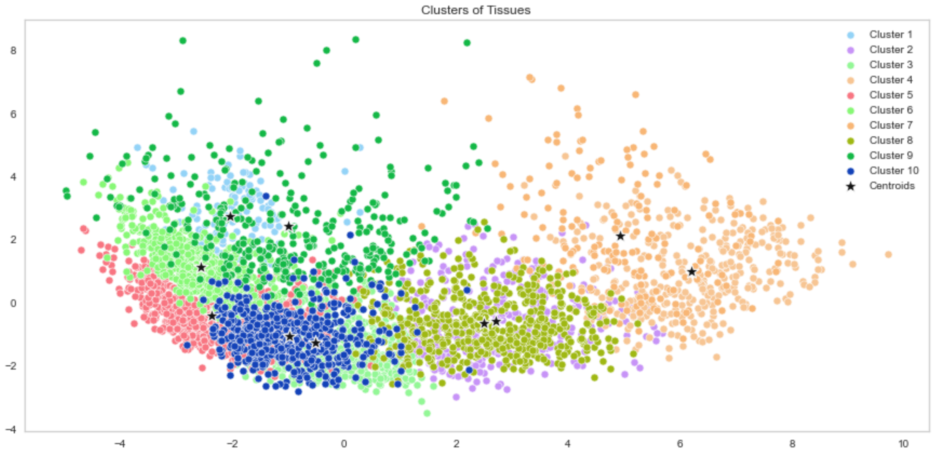


Figure 4.1: Cluster representation with their respective centroids performed on PCA vector with k=10

[3] Please refer to Jupyter Notebooks KMeans with PathologyGan, KMeans with Resnet, KMeans with Inceptionv and KMeans with VGG for codes.

4.1.1.B. PathologyGAN UMAP

From the Line Chart displayed in Figure A.2 in the Appendices section, it can be observed that *k=11* would be able to provide a good representation of clusters.

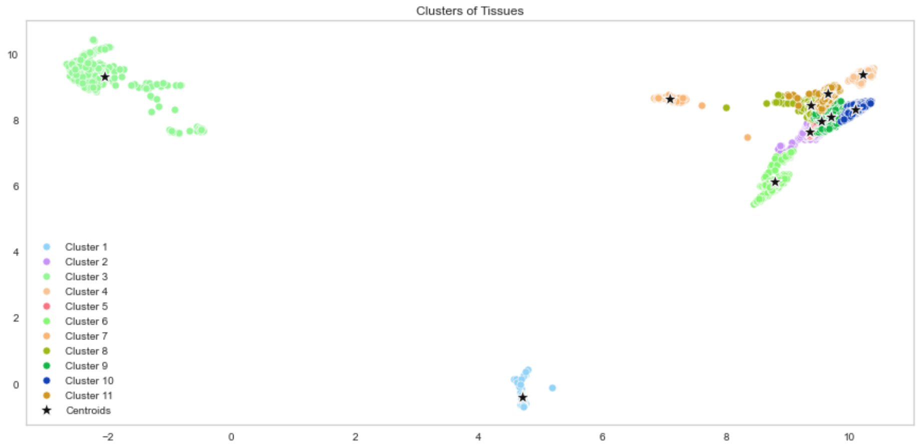


Figure 4.2: Cluster representation with their respective centroids performed on UMAP vector with k=11

4.1.2 ResNet50

4.1.2.A. ResNet50 PCA

From the Line Chart displayed in Figure B.1 in the Appendices section, it can be observed that *k=11* would be able to provide a good representation of clusters.

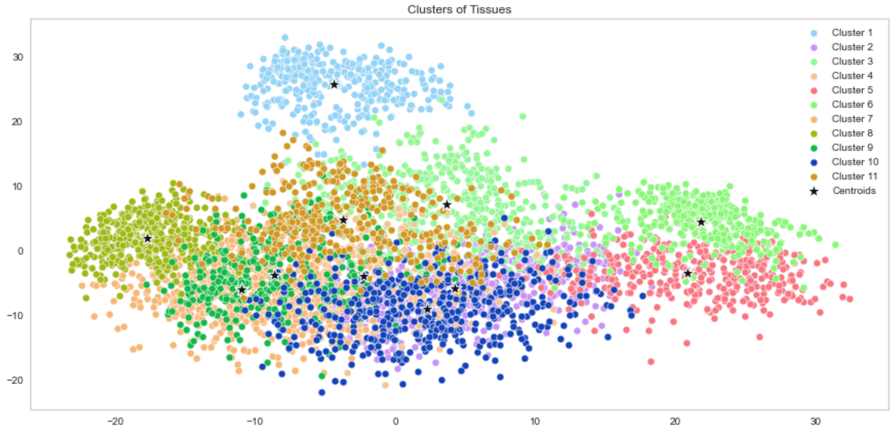


Figure 4.3: Cluster representation with their respective centroids performed on PCA vector with k=11

4.1.2.B. ResNet50 UMAP

From the Line Chart displayed in Figure B.2 in the Appendices section, it can be observed that *k=6* would be able to provide a good representation of clusters

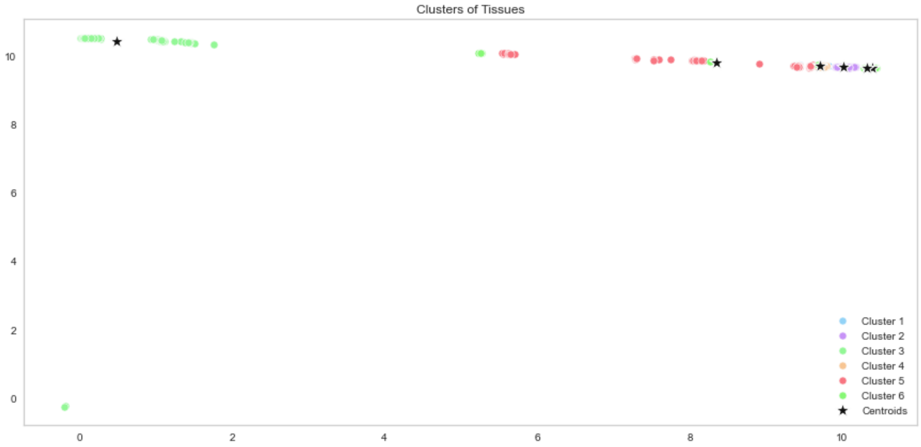


Figure 4.4: Cluster representation with their respective centroids performed on UMAP vector with k=8

4.1.3 InceptionV3

4.1.3.A. InceptionV3 PCA

From the Line Chart displayed in Figure C.1 in the Appendices section, it can be observed that *k=7* would be able to provide a good representation of clusters.

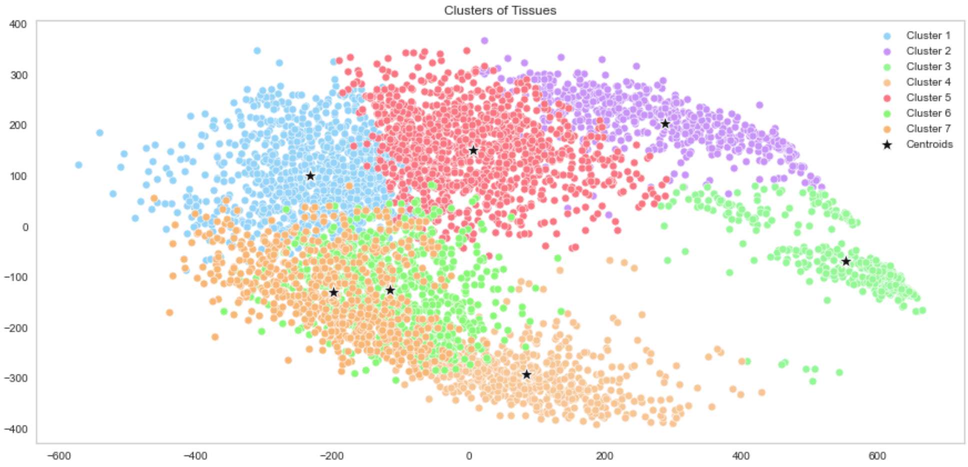


Figure 4.5: Cluster representation with their respective centroids performed on PCA vector with k=7

4.1.3.B. InceptionV3 UMAP

From the Line Chart displayed in Figure C.2 in the Appendices section, it can be observed that *k=7* would be able to provide a good representation of clusters.

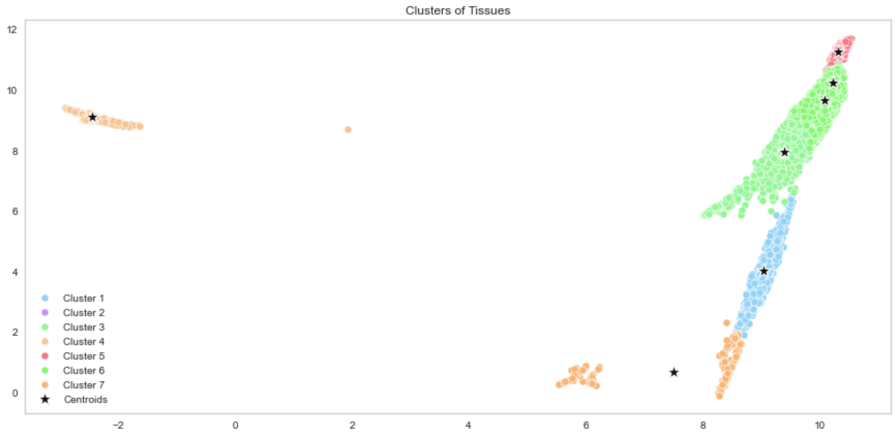


Figure 4.6: Cluster representation with their respective centroids performed on UMAP vector with k=7

4.1.4 VGG16

4.1.4.A. VGG16 PCA

From the Line Chart displayed in Figure D.1 in the Appendices section, it can be observed that *k=8* would be able to provide a good representation of clusters.

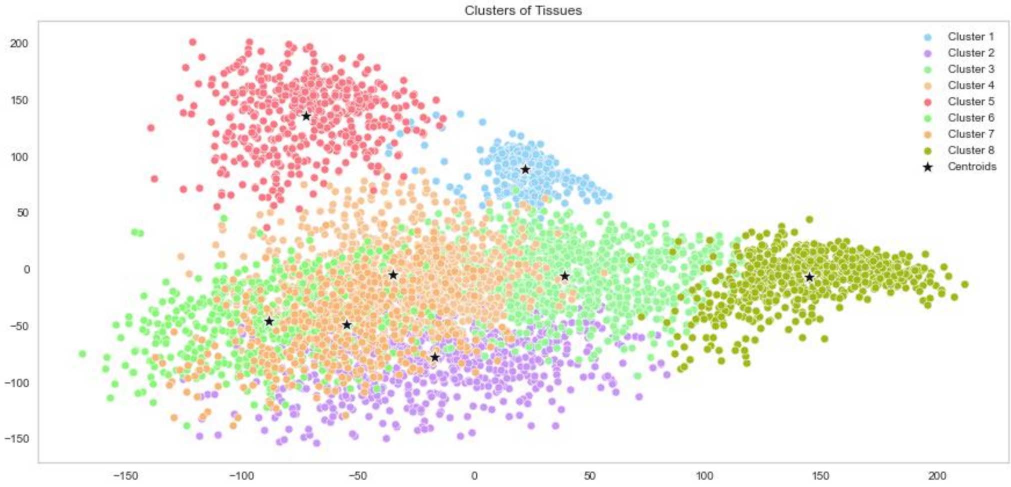


Figure 4.7: Cluster representation with their respective centroids performed on PCA vector with k=8

4.1.4.B. VGG16 UMAP

From the Line Chart displayed in Figure D.2 in the Appendices section, it can be observed that *k=7* would be able to provide a good representation of clusters.

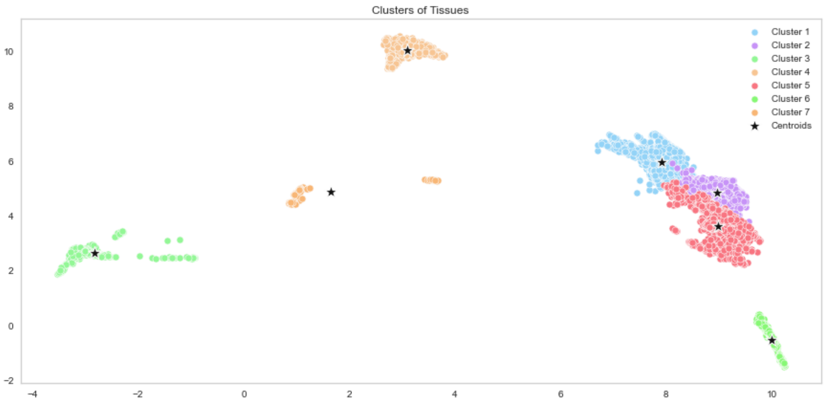


Figure 4.8: Cluster representation with their respective centroids performed on UMAP vector with k=7

4.2 Hierarchical Clustering[4]

This sub-section concerns itself with displaying the results achieved when Hierarchical Agglomerative Clustering Analysis was deployed on the 4 different representations of the colorectal tissue patches. As mentioned earlier in the report, each representation has two different vectors- PCA and UMAP. The clustering analysis was implemented on each of the vectors. Dendrograms were created for both Ward Linkage and Centroid Linkage. Subsequently, depending upon the perceivable colours on the dendrogram, clusters were created.

4.2.1 PathologyGAN

4.2.1.A. PathologyGAN PCA Ward Linkage

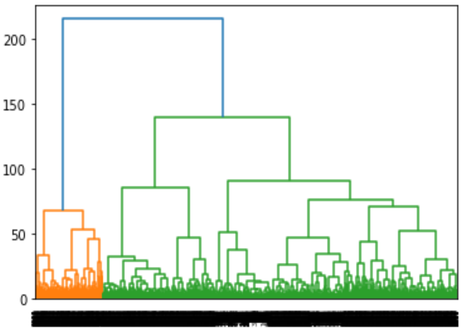


Figure 4.9: Dendrogram achieved from the PCA vector of PathologyGAN representation with ward linkage.

From the Figure 4.9, 2 colours were perceivable and thus a cluster representation with 2 clusters were created.

[4] Please refer to Jupyter Notebooks Hierarchical with PathologyGan, Hierarchical with Resnet, Hierarchical with Inceptionv and Hierarchical with VGG for codes.

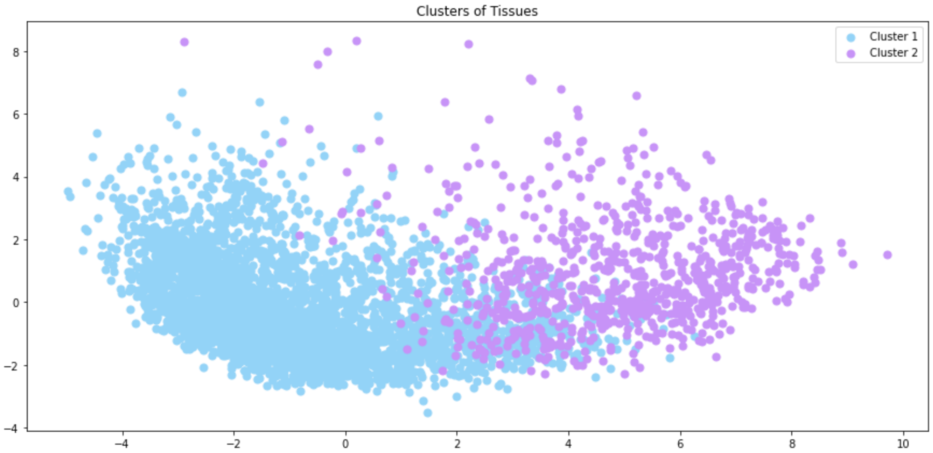


Figure 4.10: 2 clusters representation performed on PCA vector with ward linkage.

4.2.1.B. PathologyGAN PCA Centroid Linkage

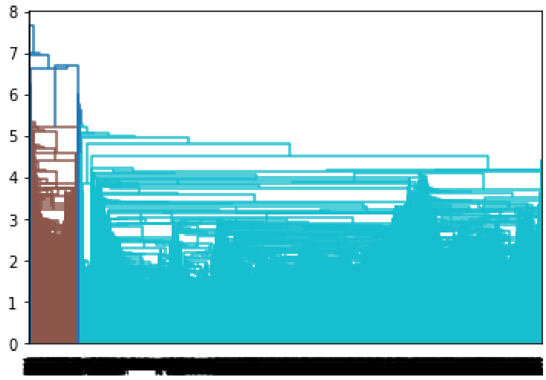


Figure 4.11: Dendrogram achieved from the PCA vector of PathologyGAN representation with centroid linkage.

From the Figure 4.11, 3 colours were perceivable and thus a cluster representation with 3 clusters were created.

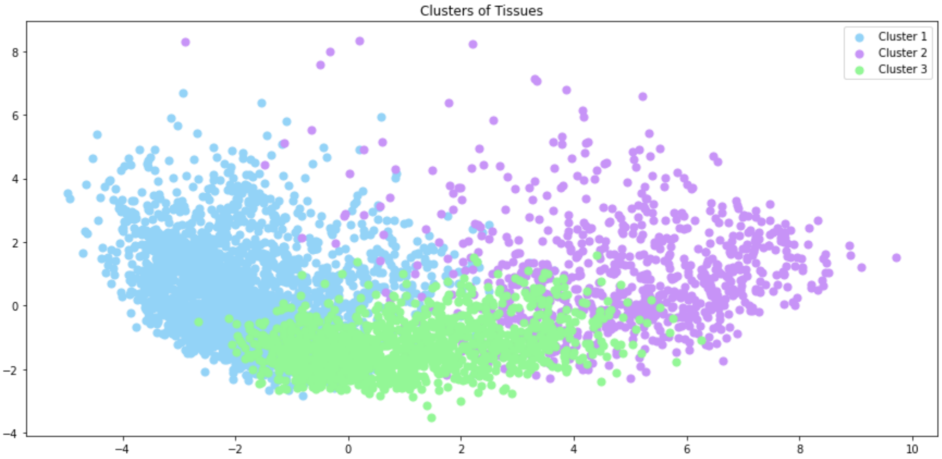


Figure 4.12: 3 clusters representation performed on PCA vector with centroid linkage.

4.2.1.C. PathologyGAN UMAP Ward Linkage

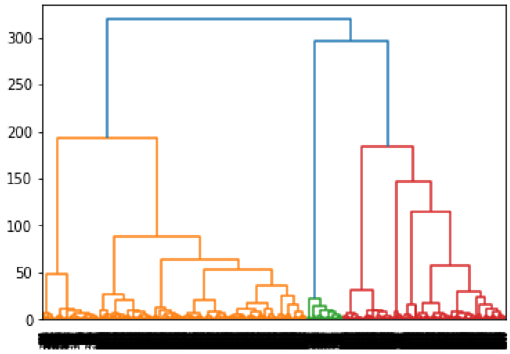


Figure 4.13: Dendrogram achieved from the UMAP vector of PathologyGAN representation with ward linkage.

From the Figure 4.13, 4 colours were perceivable and thus a cluster representation with 4 clusters were created.

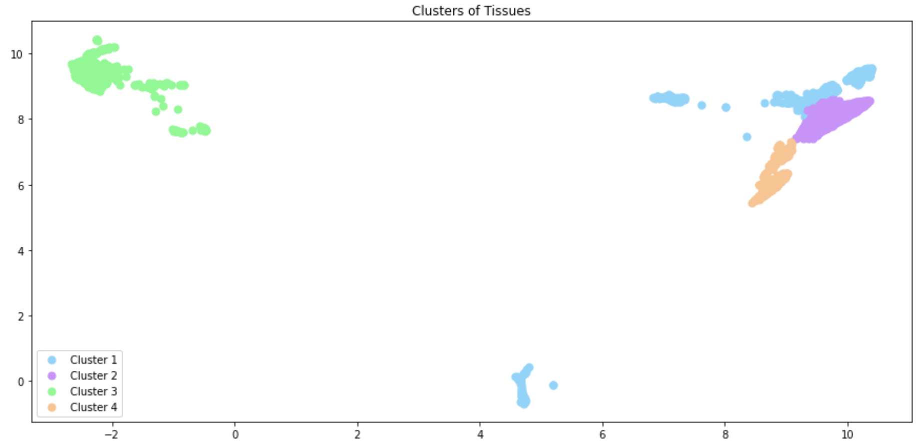
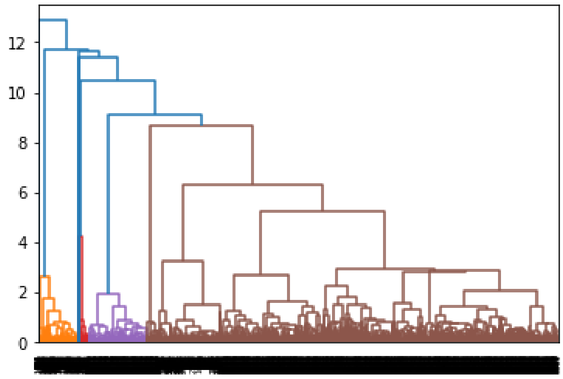


Figure 4.14: 4 clusters representation performed on UMAP vector with ward linkage.

4.2.1.D. PathologyGAN UMAP Centroid Linkage



From the Figure 4.15, 5 colours were perceivable and thus a cluster representation with 5 clusters were created.

Figure 4.15: Dendrogram achieved from the UMAP vector of PathologyGAN representation with centroid linkage.

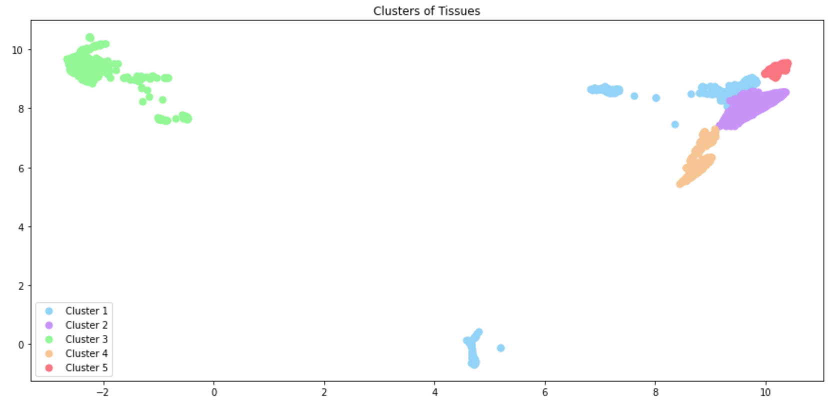


Figure 4.16: 5 clusters representation performed on UMAP vector with centroid linkage.

4.2.2 ResNet50

4.2.2.A. ResNet50 PCA Ward Linkage

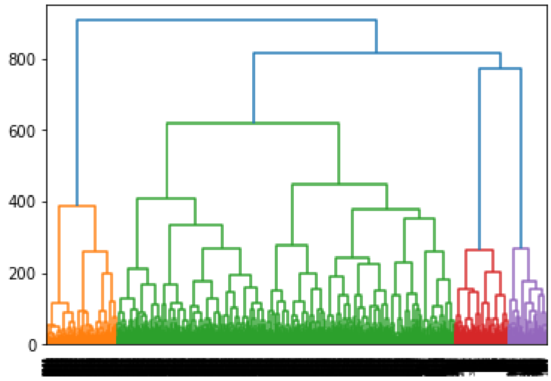


Figure 4.17: Dendrogram achieved from the PCA vector of ResNet50 representation with ward linkage.

From the Figure 4.17, 4 colours were perceivable and thus a cluster representation with 4 clusters were created

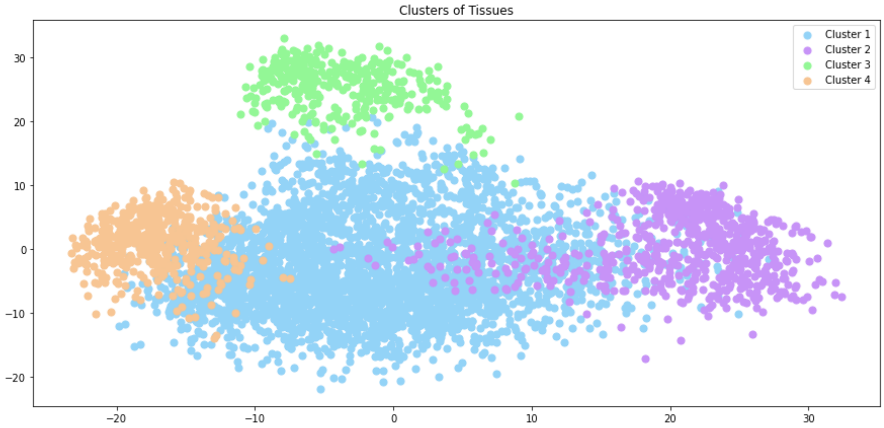


Figure 4.18: 4 clusters representation performed on PCA vector with ward linkage.

4.2.2.B. ResNet50 PCA Centroid Linkage

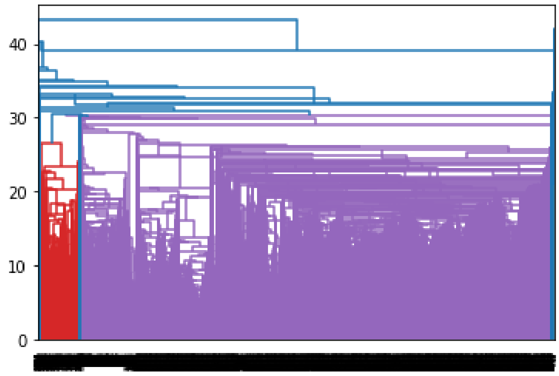


Figure 4.19: Dendrogram achieved from the PCA vector of ResNet50 representation with centroid linkage.

From the Figure 4.19, 3 colours were perceivable and thus a cluster representation with 3 clusters were created.

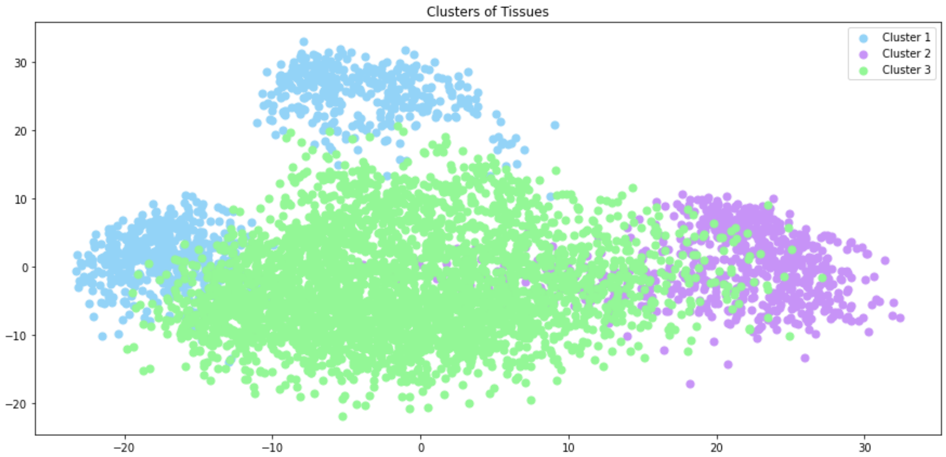


Figure 4.20: 3 clusters representation performed on PCA vector with centroid linkage.

4.2.2.C. ResNet50 UMAP Ward Linkage

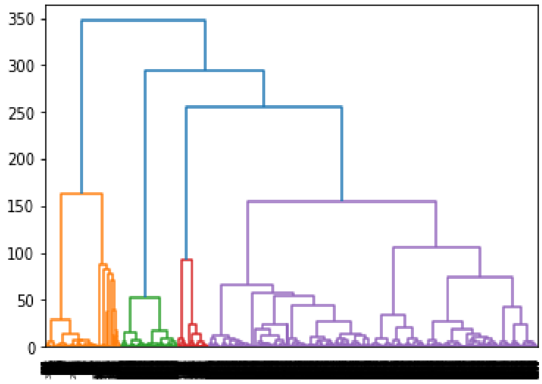


Figure 4.21: Dendrogram achieved from the UMAP vector of ResNet50 representation with ward linkage.

From the Figure 4.21, 4 colours were perceivable and thus a cluster representation with 4 clusters were created.

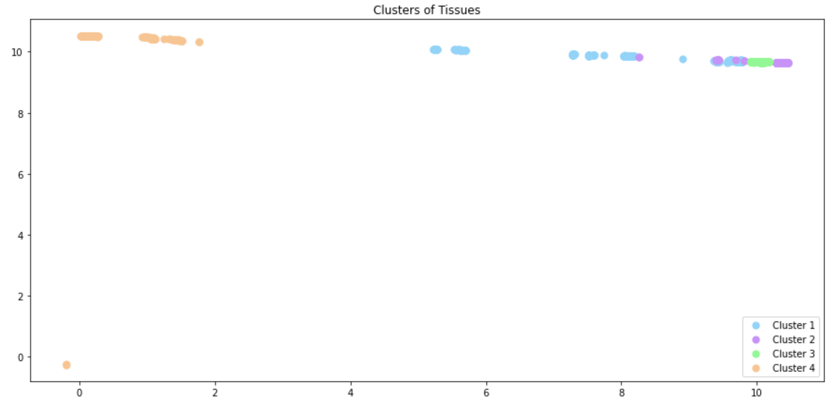
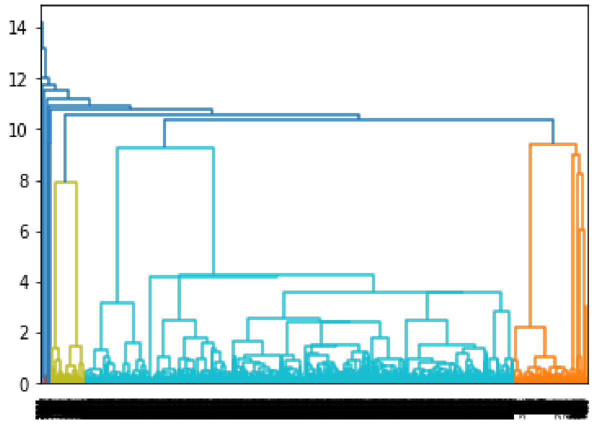


Figure 4.22: 4 clusters representation performed on UMAP vector with ward linkage.

4.2.2.D. ResNet50 UMAP Centroid Linkage



From the Figure 4.23, 4 colours were perceivable and thus a cluster representation with 4 clusters were created.

Figure 4.23: Dendrogram achieved from the UMAP vector of ResNet50 representation with centroid linkage.

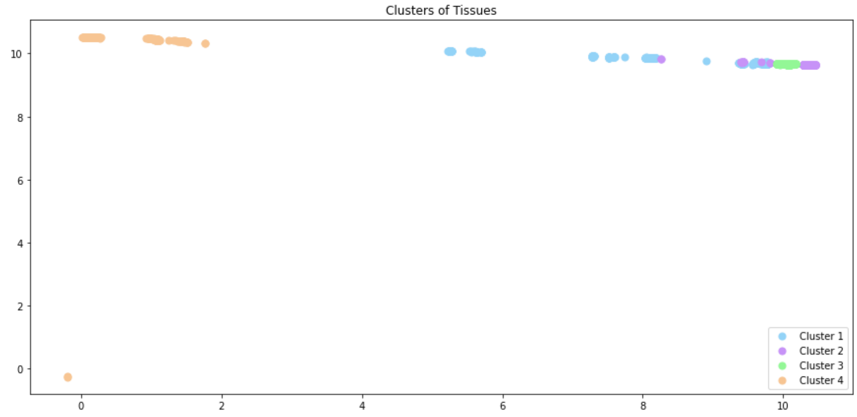
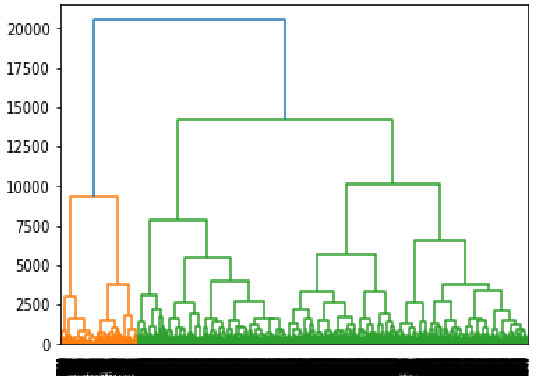


Figure 4.24: 4 clusters representation performed on UMAP vector with centroid linkage.

4.2.3 InceptionV3

4.2.3.A. InceptionV3 PCA Ward Linkage



From the Figure 4.25, 2 colours were perceivable and thus a cluster representation with 2 clusters were created.

Figure 4.25: Dendrogram achieved from the PCA vector of InceptionV3 representation with ward linkage.

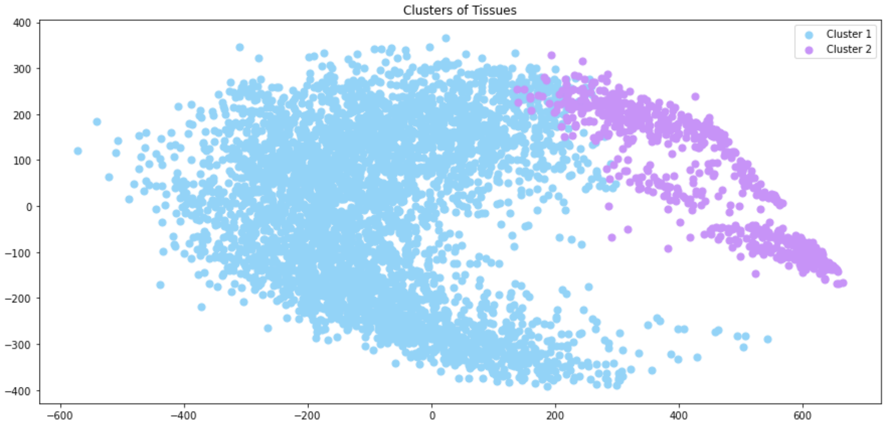
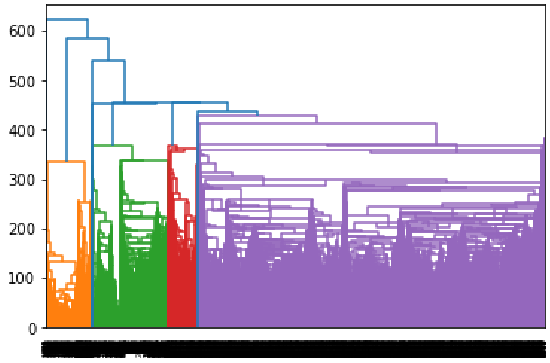


Figure 4.26: 2 clusters representation performed on PCA vector with ward linkage.

4.2.3.B. InceptionV3 PCA Centroid Linkage



From the Figure 4.27, 5 colours were perceivable and thus a cluster representation with 5 clusters were created.

Figure 4.27: Dendrogram achieved from the PCA vector of InceptionV3 representation with centroid linkage.

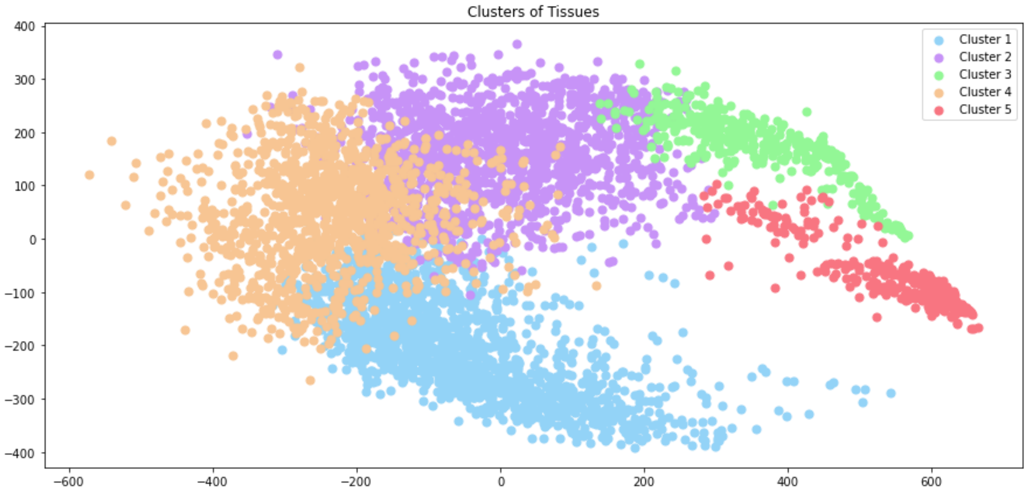


Figure 4.28: 5 clusters representation performed on PCA vector with centroid linkage.

4.2.3.C. InceptionV3 UMAP Ward Linkage

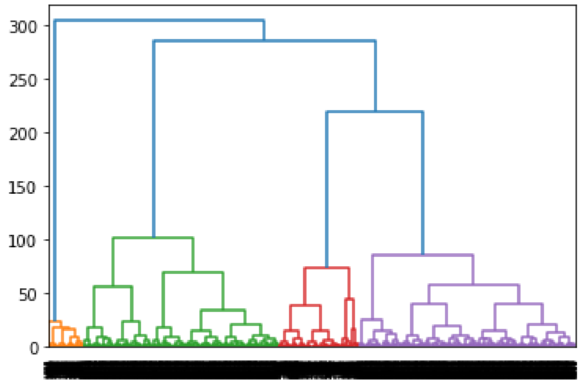


Figure 4.29: Dendrogram achieved from the UMAP vector of InceptionV3 representation with ward linkage.

From the Figure 4.29, 4 colours were perceivable and thus a cluster representation with 4 clusters were created.

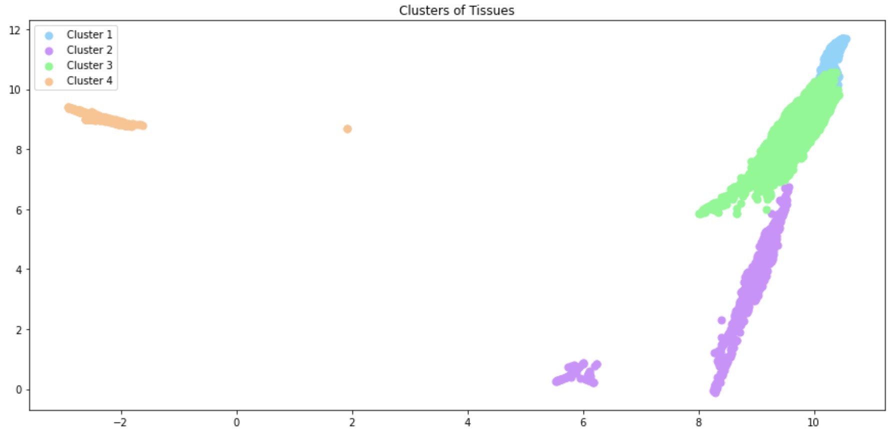
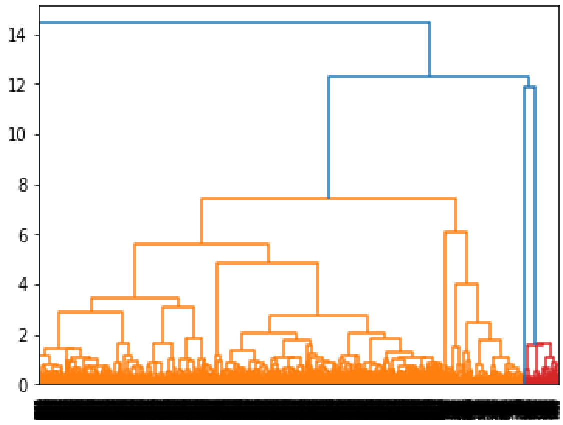


Figure 4.30: 4 clusters representation performed on UMAP vector with ward linkage.

4.2.3.D. InceptionV3 UMAP Centroid Linkage



From the Figure 4.31, 3 colours were perceivable and thus a cluster representation with 3 clusters were created.

Figure 4.31: Dendrogram achieved from the UMAP vector of InceptionV3 representation with centroid linkage.

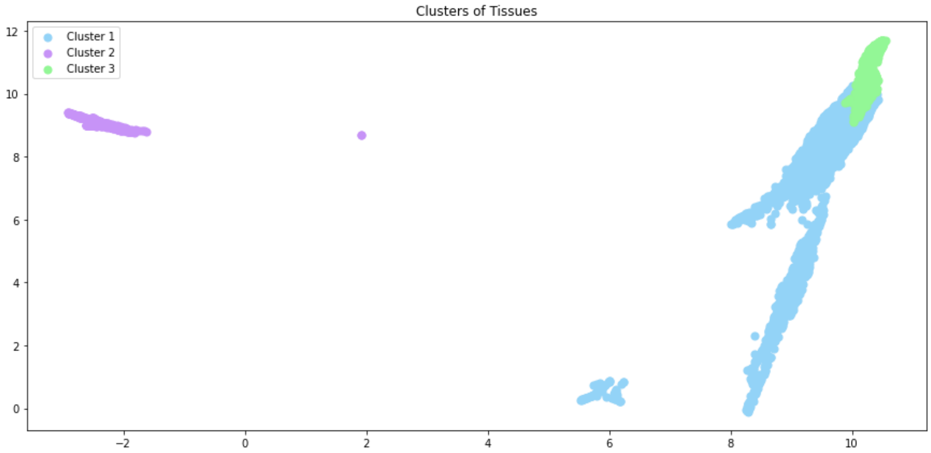
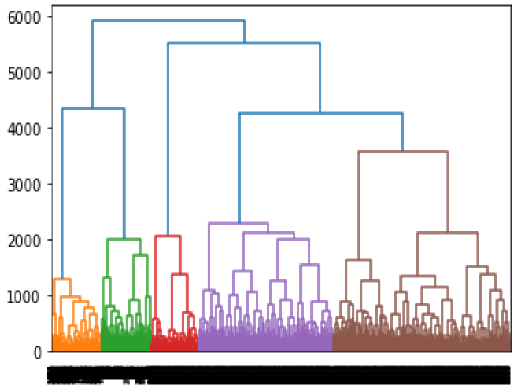


Figure 4.32: 3 clusters representation performed on UMAP vector with centroid linkage.

4.2.4 VGG16

4.2.4.A. VGG16 PCA Ward Linkage



From the Figure 4.33, 5 colours were perceivable and thus a cluster representation with 5 clusters were created.

Figure 4.33: Dendrogram achieved from the PCA vector of VGG16 representation with ward linkage.

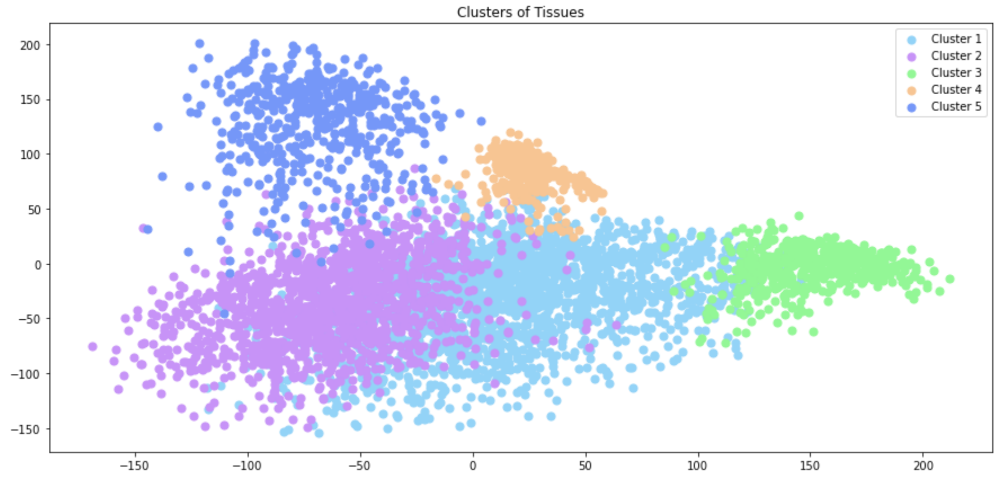


Figure 4.34: 5 clusters representation performed on PCA vector with ward linkage.

4.2.4.B. VGG16 PCA Centroid Linkage

From the Figure 4.35, only 1 colour was perceivable. However, to somewhat represent the different types of colorectal tissues a cluster representation with 2 clusters were created.

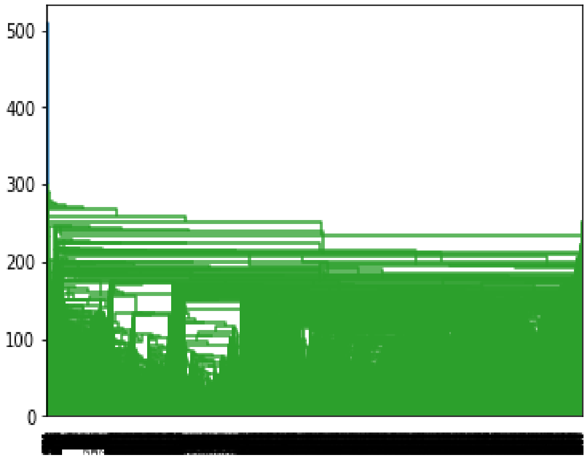


Figure 4.35: Dendrogram achieved from the PCA vector of VGG16 representation with centroid linkage.

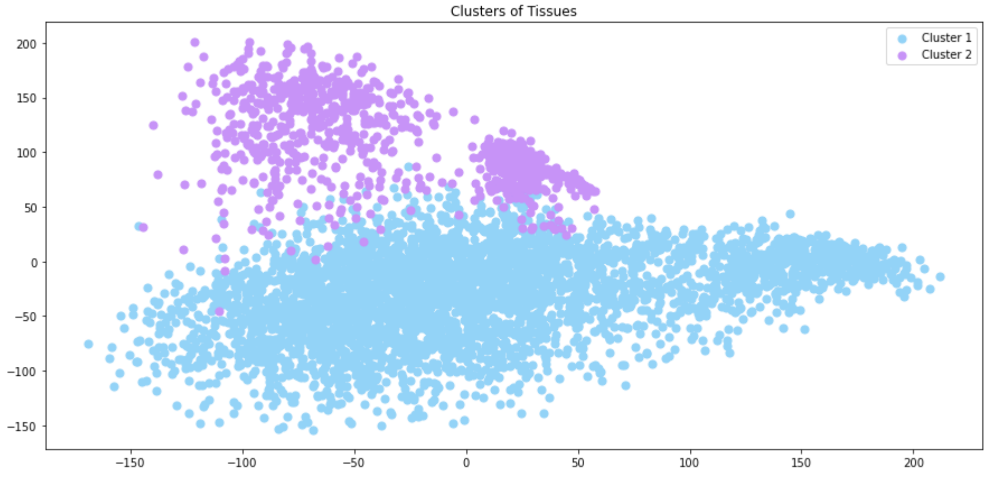


Figure 4.36: 2 clusters representation performed on PCA vector with centroid linkage.

4.2.4.C. VGG16 UMAP Ward Linkage

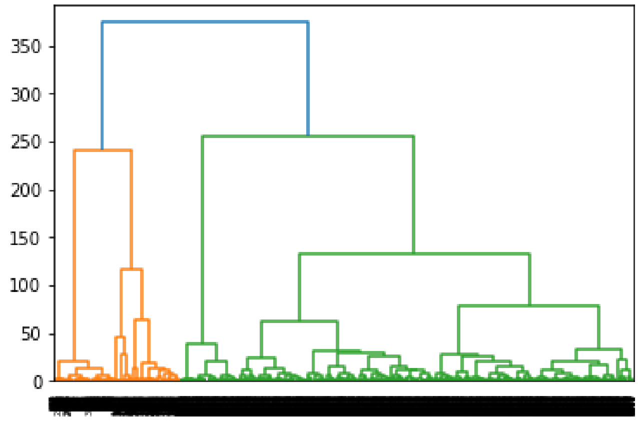


Figure 4.37: Dendrogram achieved from the UMAP vector of VGG16 representation with ward linkage.

From the Figure 4.37, 2 colours were perceivable and thus a cluster representation with 2 clusters were created.

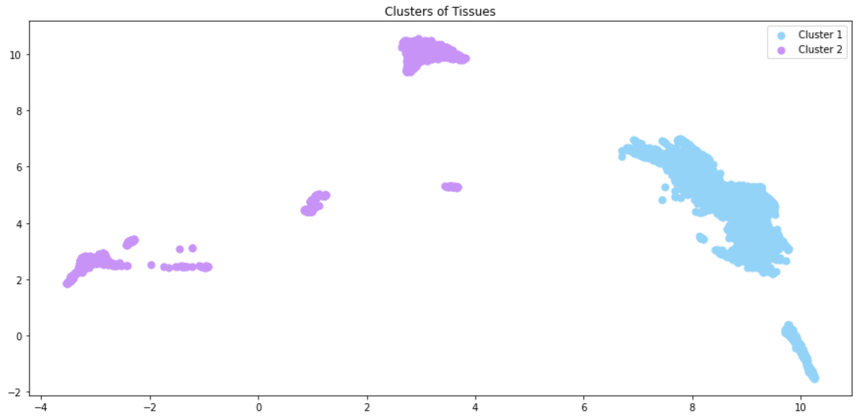
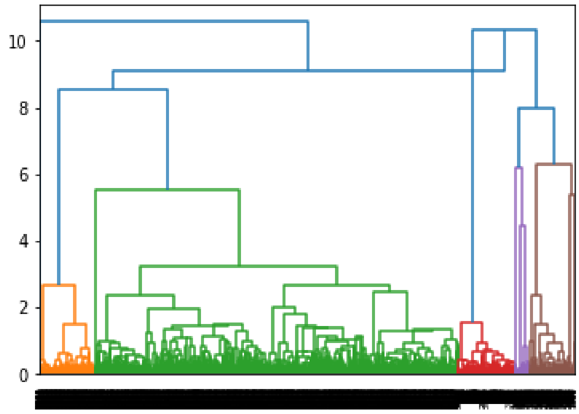


Figure 4.38: 2 clusters representation performed on UMAP vector with ward linkage.

4.2.4.D. VGG16 UMAP Centroid Linkage



From the Figure 4.39, 5 colours were perceivable and thus a cluster representation with 5 clusters were created.

Figure 4.39: Dendrogram achieved from the UMAP vector of VGG16 representation with centroid linkage.

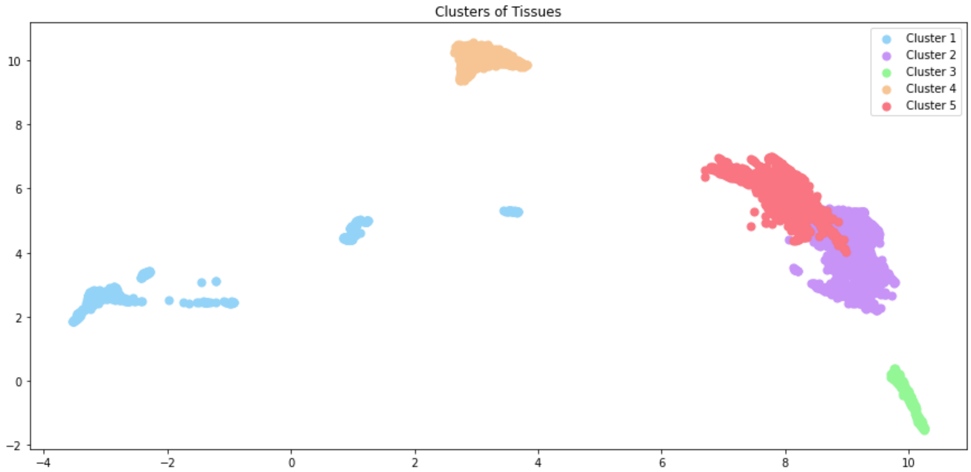


Figure 4.40: 5 clusters representation performed on UMAP vector with centroid linkage.

Chapter 5 DISCUSSIONS

In this chapter, we will explore the pros and cons encountered while applying K-Means Clustering Analysis and Hierarchical Clustering Analysis in a tabular format. Moreover, we will observe the differences in the cluster representations presented in the previous chapter.

K-Means Clustering Analysis	***Hierarchical Clustering Analysis***
Silhouette Score and V-Measure score provided a sensical way of determining the near optimal number of clusters needed for representation of data.	Dendrograms were ambiguous in nature as most of the clusters merged together pretty early during execution.
Not time consuming.	A significant amount of time was required when creating dendrograms.
Did not run into computational problem.	Trial with single linkage ran into computational problem.
Provided a variety of cluster representations compared to Hierarchical Clustering Analysis.	Was unable to provide sensical cluster representations. One of the many instances of that can be observed by comparing Figure 4.6 and Figure 4.26.
Homogeneity and completeness could be tested for.	Upon analysing for homogeneity and completeness, Hierarchical Clustering scored poorly as compared to K-Means. The score table can be found in Appendix F.

Table 5.1: Observations from implementation of K-Means Clustering Analysis and Hierarchical Clustering Analysis.

Chapter 6 CONCLUSION

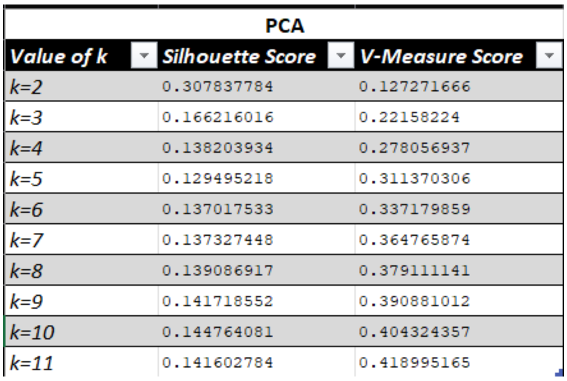
In this case-study we apply two different Cluster Analysis algorithm on colorectal tissue patches that a represented through 4 different data set. Each algorithm and their variation were applied on both the PCA and UMAP vectors of the representation.

Observing the information presented in Table 5.1, it can be said that for this particular case-study K-Means Clustering Analysis would be a better choice for tissue analysis as compared to Hierarchical Clustering Analysis.

However, since a plethora of other clustering analysis algorithms exist that are proven to work better than K-Means Clustering Analysis algorithm, the possibility of achieving optimal result on an overlapping dataset as the one in the case study must not be ruled out.

APPENDICES

A.



	PCA	
Value of k	Silhouette Score	V-Measure Score
k=2	0.307837784	0.127271666
k=3	0.166216016	0.22158224
k=4	0.138203934	0.278056937
k=5	0.129495218	0.311370306
k=6	0.137017533	0.337179859
k=7	0.137327448	0.364765874
k=8	0.139086917	0.379111141
k=9	0.141718552	0.390881012
k=10	0.144764081	0.404324357
k=11	0.141602784	0.418995165

Table A.1: Silhouette Scores and V-Measure Scores for their respective cluster analysis applied on the PCA vector of PathologyGAN representation.

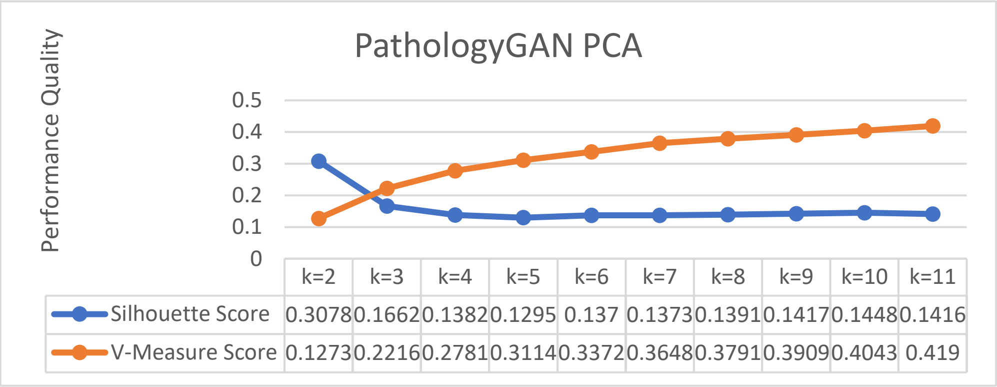
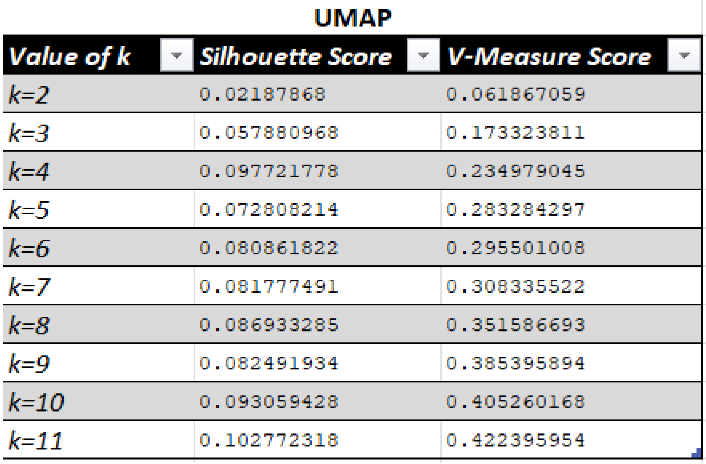


Figure A.1: Line Chart derived from Table A.1



	UMAP	
Value of k	Silhouette Score	V-Measure Score
k=2	0.02187868	0.061867059
k=3	0.057880968	0.173323811
k=4	0.097721778	0.234979045
k=5	0.072808214	0.283284297
k=6	0.080861822	0.295501008
k=7	0.081777491	0.308335522
k=8	0.086933285	0.351586693
k=9	0.082491934	0.385395894
k=10	0.093059428	0.405260168
k=11	0.102772318	0.422395954

Table A.2: Silhouette Scores and V-Measure Scores for their respective cluster analysis applied on the UMAP vector of PathologyGAN representation

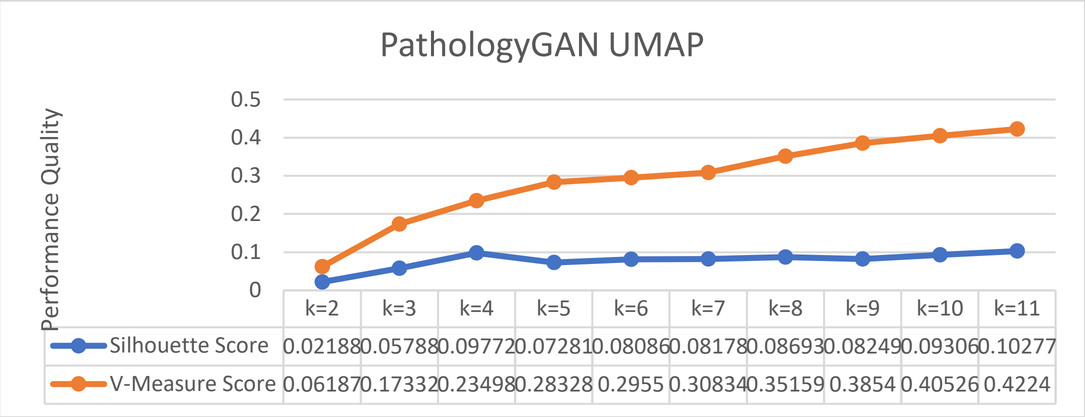
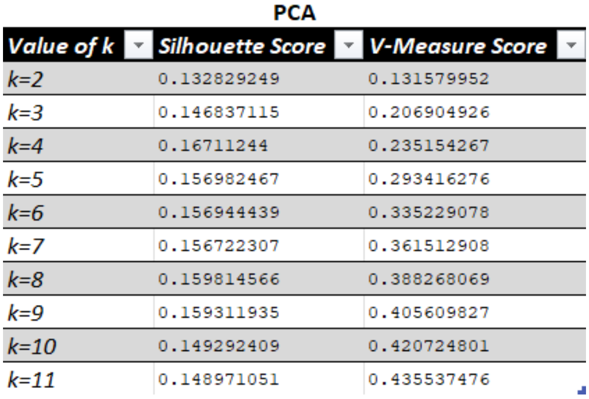


Figure A.2: Line Chart derived from Table A.2

B.



PCA

Value of k	*Silhouette Score*	*V-Measure Score*
k=2	0.132829249	0.131579952
k=3	0.146837115	0.206904926
k=4	0.16711244	0.235154267
k=5	0.156982467	0.293416276
k=6	0.156944439	0.335229078
k=7	0.156722307	0.361512908
k=8	0.159814566	0.388268069
k=9	0.159311935	0.405609827
k=10	0.149292409	0.420724801
k=11	0.148971051	0.435537476

Table B.1: Silhouette Scores and V-Measure Scores for their respective cluster analysis applied on the PCA vector of ResNet50 representation.

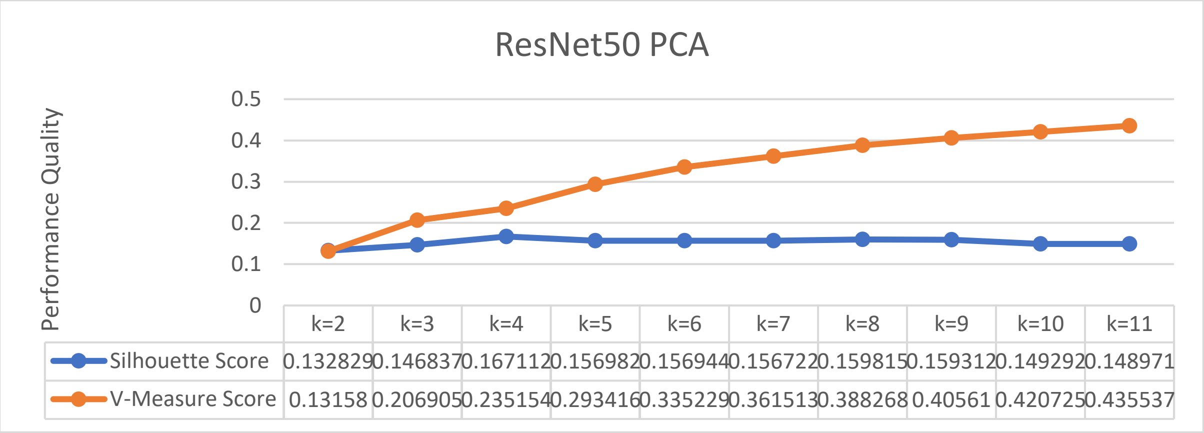
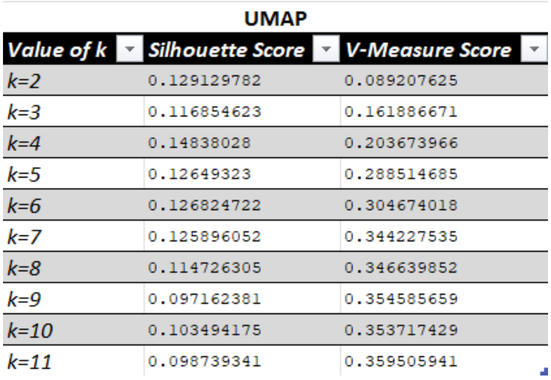


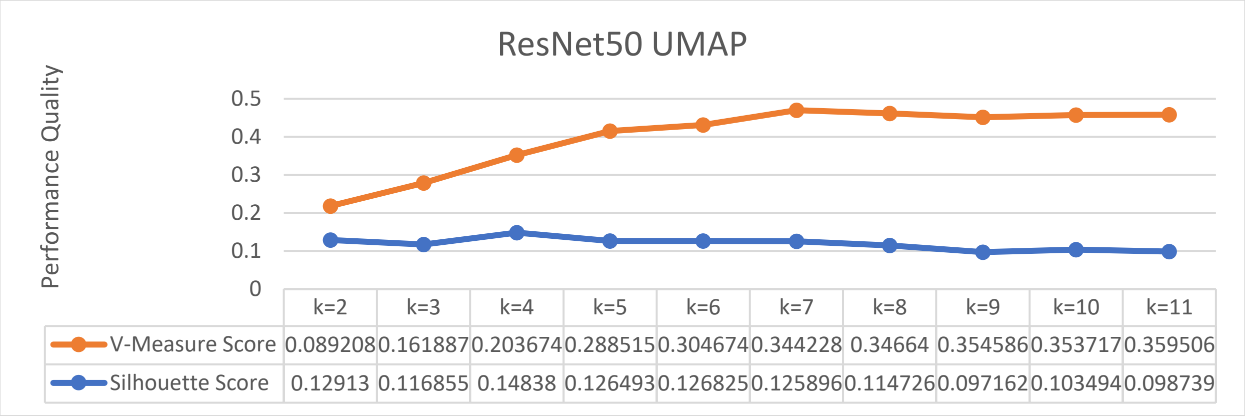
Figure B.1: Line Chart derived from Table B.1



UMAP

Value of k	Silhouette Score	V-Measure Score
k=2	0.129129782	0.089207625
k=3	0.116854623	0.161886671
k=4	0.14838028	0.203673966
k=5	0.12649323	0.288514685
k=6	0.126824722	0.304674018
k=7	0.125896052	0.344227535
k=8	0.114726305	0.346639852
k=9	0.097162381	0.354585659
k=10	0.103494175	0.353717429
k=11	0.098739341	0.359505941

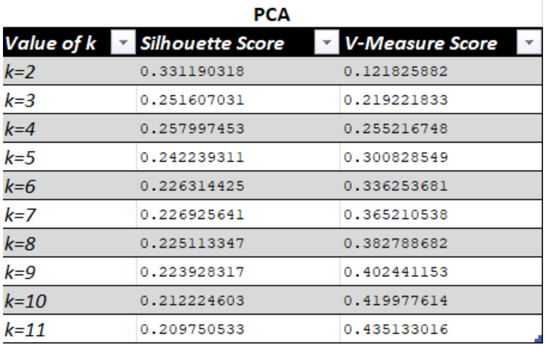
Table B.2: Silhouette Scores and V-Measure Scores for their respective cluster analysis applied on the UMAP vector of ResNet50 representation.



	k=2	k=3	k=4	k=5	k=6	k=7	k=8	k=9	k=10	k=11
V-Measure Score	0.089208	0.161887	0.203674	0.288515	0.304674	0.344228	0.34664	0.354586	0.353717	0.359506
Silhouette Score	0.12913	0.116855	0.14838	0.126493	0.126825	0.125896	0.114726	0.097162	0.103494	0.098739

Figure B.2: Line Chart derived from Table B.2

C.



PCA

Value of k	Silhouette Score	V-Measure Score
k=2	0.331190318	0.121825882
k=3	0.251607031	0.219221833
k=4	0.257997453	0.255216748
k=5	0.242239311	0.300828549
k=6	0.226314425	0.336253681
k=7	0.226925641	0.365210538
k=8	0.225113347	0.382788682
k=9	0.223928317	0.402441153
k=10	0.212224603	0.419977614
k=11	0.209750533	0.435133016

Table C.1: Silhouette Scores and V-Measure Scores for their respective cluster analysis applied on the PCA vector of InceptionV3 representation.

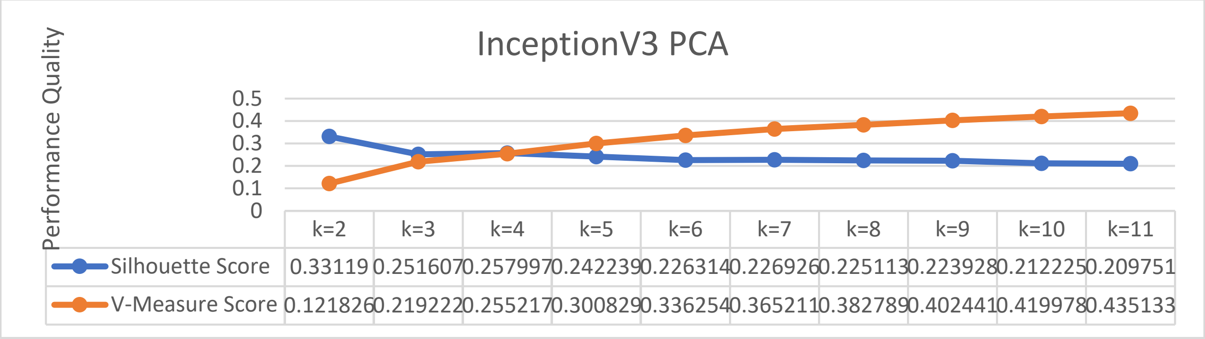
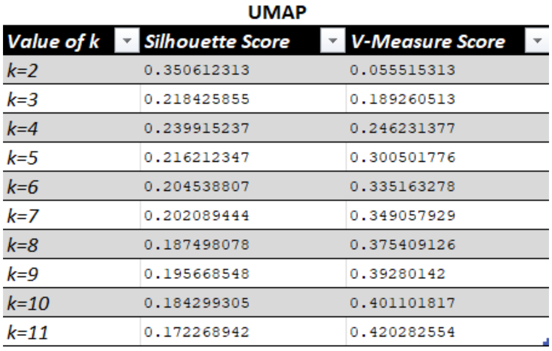


Figure C.1: Line Chart derived from Table C.1



UMAP

Value of k	Silhouette Score	V-Measure Score
k=2	0.350612313	0.055515313
k=3	0.218425855	0.189260513
k=4	0.239915237	0.246231377
k=5	0.216212347	0.300501776
k=6	0.204538807	0.335163278
k=7	0.202089444	0.349057929
k=8	0.187498078	0.375409126
k=9	0.195668548	0.39280142
k=10	0.184299305	0.401101817
k=11	0.172268942	0.420282554

Table C.2: Silhouette Scores and V-Measure Scores for their respective cluster analysis applied on the UMAP vector of InceptionV3 representation.

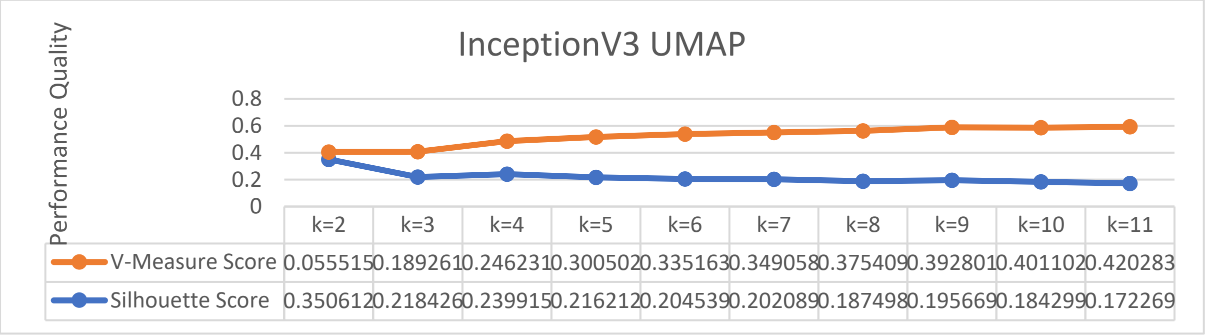
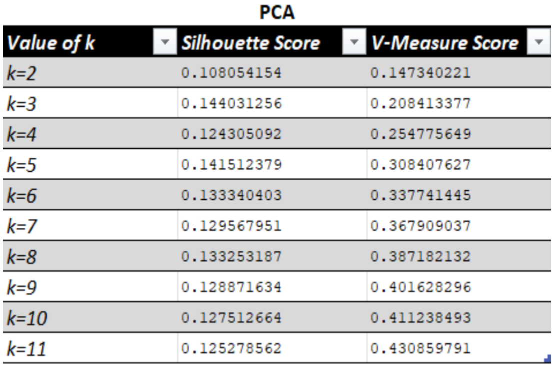


Figure C.1: Line Chart derived from Table C.2

D.



PCA

Value of k	Silhouette Score	V-Measure Score
k=2	0.108054154	0.147340221
k=3	0.144031256	0.208413377
k=4	0.124305092	0.254775649
k=5	0.141512379	0.308407627
k=6	0.133340403	0.337741445
k=7	0.129567951	0.367909037
k=8	0.133253187	0.387182132
k=9	0.128871634	0.401628296
k=10	0.127512664	0.411238493
k=11	0.125278562	0.430859791

Table D.1: Silhouette Scores and V-Measure Scores for their respective cluster analysis applied on the PCA vector of VGG16 representation.

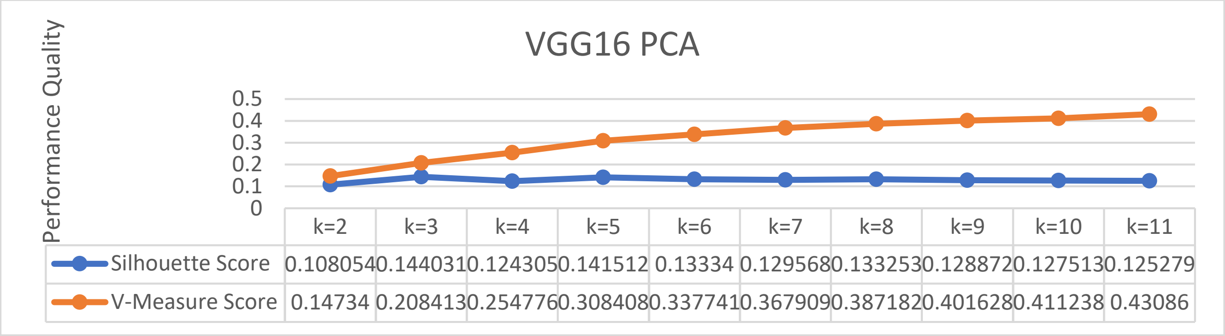
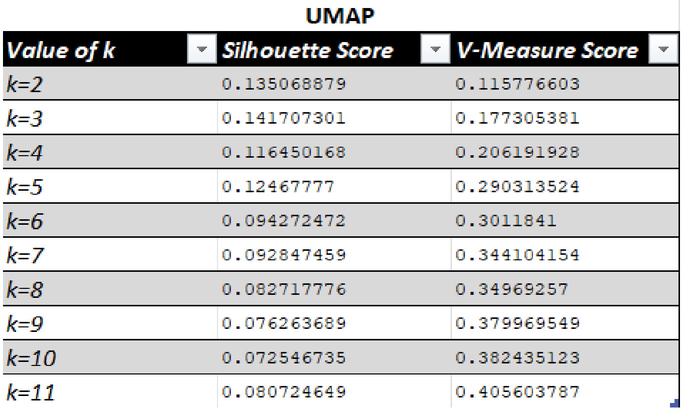


Figure D.1: Line Chart derived from Table D.1



UMAP

Value of k	*Silhouette Score*	*V-Measure Score*
k=2	0.135068879	0.115776603
k=3	0.141707301	0.177305381
k=4	0.116450168	0.206191928
k=5	0.12467777	0.290313524
k=6	0.094272472	0.3011841
k=7	0.092847459	0.344104154
k=8	0.082717776	0.34969257
k=9	0.076263689	0.379969549
k=10	0.072546735	0.382435123
k=11	0.080724649	0.405603787

Table D.2: Silhouette Scores and V-Measure Scores for their respective cluster analysis applied on the UMAP vector of VGG16 representation

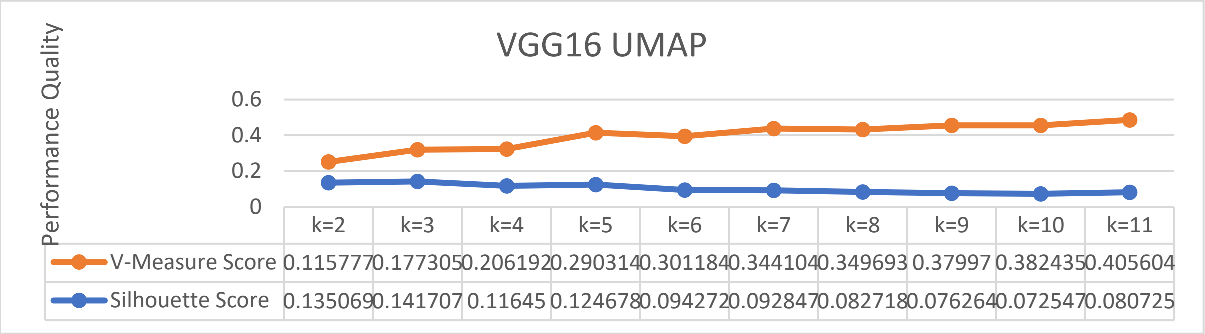


Figure D.2: Line Chart derived from Table D.2

E. The following table lists the Silhouette Score and V-Measure Score for the Hierarchical Clustering experiments ran on the 4 different datasets.

	InceptionV3				***ResNet50***				***PathologyGAN***				***VGG16***			
	PCA		UMAP		PCA		UMAP		PCA		UMAP		PCA		UMAP	
	Ward	Centroid	Ward	Centroid	Ward	Centroid	Ward	Centroid	Ward	Centroid	Ward	Centroid	Ward	Centroid	Ward	Centroid
Silhouette Score	0.3494	0.2080	0.5141	0.4479	0.1515	0.1335	0.5791	0.5791	0.3199	0.1376	0.5032	0.5531	0.1225	0.1346	0.5683	0.5124
V-Measure Score	0.1005	0.2954	0.2470	0.1853	0.2062	0.1824	0.2066	0.2066	0.0981	0.2034	0.2437	0.2823	0.2906	0.1164	0.1157	0.2909

REFERENCES

[1] Cluster Analysis. Wikipedia 2021. Wikipedia: The Free Encyclopaedia. Retrieved from https://en.wikipedia.org/wiki/Cluster_analysis

[2] K-Means Clustering. Wikipedia 2021. Wikipedia: The Free Encyclopaedia. Retrieved from https://en.wikipedia.org/wiki/K-means_clustering

[3] The (black) art of runtime evaluation: Are we comparing algorithms or implementations? by Hans-Peter Kriegel, Erich Schubert & Arthur Zimek. Published on 22nd October, 2016. Published in Knowledge and Information Systems **52**, 341-378(2017). DOI: https://doi.org/10.1007/s10115-016-1004-2

[4] Euclidean Distance. Wikipedia 2021. Wikipedia: The Free Encyclopaedia. Retrieved from https://en.wikipedia.org/wiki/Euclidean_distance

[5] Voronoi Diagram. Wikipedia 2021. Wikipedia: The Free Encyclopaedia. Retrieved from https://en.wikipedia.org/wiki/Voronoi_diagram

[6] Hierarchical Clustering. Wikipedia 2021. Wikipedia: The Free Encyclopaedia. Retrieved from https://en.wikipedia.org/wiki/Hierarchical_clustering

[7] Hierarchical clustering in Python using Dendrogram and Cophenetic Correlation by Angel Das. Published on September 12th, 2020. Towards Data Science 2021. Retrieved from https://towardsdatascience.com/hierarchical-clustering-in-python-using-dendrogram-and-cophenetic-correlation-8d41a08f7eab

[8] KMeans Silhouette Score Explained with Python Example by Ajitesh Kumar. Published on September 15th, 2020. Vitalflux.com 2021. Retrieved from https://vitalflux.com/kmeans-silhouette-score-explained-with-python-example/#Perform_Comparative_Analysis_to_Determine_Best_value_of_K_using_Silhouette_Plot

[9] V-Measure: A conditional entropy-based external cluster evaluation measure by Andrew Rosenberg and Julia Hirschberg. Published in EMNLP 1st June, 2007.

[10] 2.3.9.3. Homogeneity, completeness and V-measure by International Scholar Pooh. Published on 3rd November, 2018. WordPress 2021. Retrieved from https://esigma6.wordpress.com/2018/11/03/2-3-9-3-homogeneity-completeness-and-v-measure/