Assignment 2, Task 2: Semi-supervised learning

Tim Eckhart (s3952509)[1], Kyriakos Antoniou (s5715881)[1], Robert Power (s5332419)[1], and Abi Raveenthiran (s4010132)[1]

Group [06]
Pattern Recognition (WMAI021-05) 2023-2024.1A
[1] Artificial Intelligence Department, University of Groningen
{*t.eckhardt,k.antoniou,r.power,a.raveenthiran*}@student.rug.nl

August 29, 2025

Abstract

The classification of numerical data and image data was investigated using several different known data processing methods; this included feature extraction methods, clustering methods, and classification methods. The numerical dataset consisted of RNA-sequences that correspond to types of tumors. The image data consisted of photo's that correspond to specific types of animals. Feature extraction methods included Principal component analysis and Mutual information, SIFT, and ORB. Clustering methods included Bisecting k-Means, k-Means, and DBSCAN. Classification methods included K-nearest neighbours, Random forests, Gaussian naive Bayes, Logistic regression, and Support-vector machine. For the numerical pipeline, it was found that PCA in combination with Random forests is the best combination of feature extractor and classifier. For the image pipeline, it was found that SIFT, together with logistic regression, is the best combination of feature extractor and classifier. For the numerical dataset, it was found that Bisecting k-Means can cluster PCA-reduced data much better compared to the non-reduced data. For the image dataset, DBSCAN could not cluster either the regular image dataset nor the extracted features from these images into more than 1 cluster. It can be concluded that, especially for the numerical data, a well-working processing pipeline was established. For the image data, more work is needed to improve the performance of classification, using, for example, an ensemble of different classifiers.

1 Introduction

This report presents the development of pattern recognition (PR) pipelines for two unique data types, namely image data and numerical data. Pattern recognition pipelines have practical use cases in many fields including, but certainly not limited to, bioinformatics [4] and computer vision [20]. The objective of this task is to select different methods for each pipeline component, analyze the results of these methods, and determine which combination of methods results in the most effective pipeline for each data type. A review of existing literature concerning the selected methods is presented in this section. Section 1.1 discusses the image data PR pipeline and section 1.2 discusses the numerical data PR pipeline.

1.1 Image Pipeline

In this section, the concepts relevant to the image pipeline are discussed. Section 1.1.1 introduces the dataset used and presents an analysis of the data, section 1.1.2 introduces the selected feature extraction methods, section 1.1.3 introduces the chosen prediction models, and section 1.1.4 presents the clustering method used.

1.1.1 Image Dataset

For the image PR pipeline, the Wild-Anim dataset [14]. The original Wild-Anim dataset, D_A, consists of 5000 images with five classes:

- Class 1: Bear
- Class 2: Elephant
- Class 3: Leopard
- Class 4: Lion
- Class 5: Wolf

The dataset is perfectly balanced, with 1000 images for each class. For this task, a sampling process was implemented to attain a more manageable dataset. A random sample of 50 images that met the resolution requirements, the image had to be at least 150x150 pixels across both dimensions, was taken from each class. The resulting sampled dataset, D_S, is perfectly balanced. The class distribution of the sampled dataset is illustrated in Figure 1, of 250 images with 50 images in each class. Example images of each class are presented in Figures (2, 3, 4, 5, and 6).

1.1.2 Feature Extraction

For the feature extraction on the image data, two methods were tested. These methods were SIFT and ORB.
Scale Invariant Feature Transform (SIFT) [12] is a method that is able to extract 'key points' from image data. These key points can be used for the matching and recognition of certain objects in images. One distinctive feature of SIFT is that it can find these key points while being invariant to scale and rotation. For example, if two images both contain a particular object, but in one of the images, the object is displayed smaller and darker, in addition to being rotated, SIFT should still be able to find the similar key points in both of these images. The way in which SIFT is able to match keypoints, is because it describes these keypoints using arrays of 128 integers. The descriptors are in essence 128-dimensional datapoints, of which the exact value is determined by the local pixels of the keypoint.
Oriented FAST and Rotated BRIEF (ORB) [18] is another method which makes use of keypoints to extract features from image data. ORB is based on two other feature extraction methods: Features from Accelerated Segment Test (FAST) [17] and Binary Robust Independent Elementary Features (BRIEF) [2], which are both keypoint-based methods as well. These two method's main advantage over SIFT is that they use less resource, at the cost of the ability to detect keypoints as reliably. ORB aims to combine these two methods, keeping the improved resource usage of BRIEF and FAST, while achieving a performance which is comparable to SIFT using some additional improvements. Still, ORB is not just a faster implementation of SIFT, for instance, it's descriptors only consist of 32 integers, as opposed to the 128 integers that SIFT descriptors have.

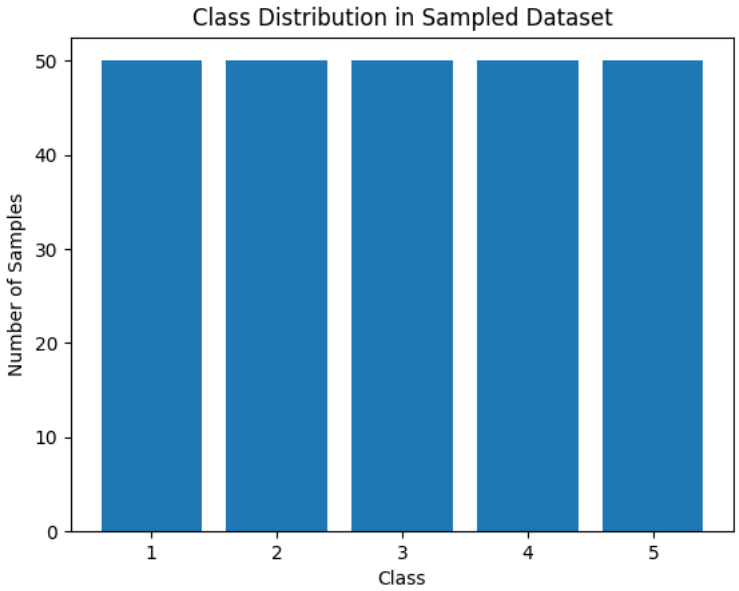

Figure 1: Class Distribution in Sampled Dataset

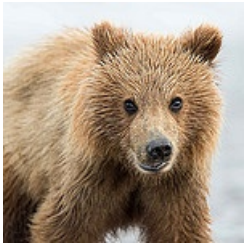

Figure 2: Class 1 (Bear) Example Image

To be able to use the keypoints that are produced by the above mentioned kinds of feature extraction methods, an additional step has to be applied, this step is a part of the (visual) bag-of-words model [3]. The bag-of-words model is an end-to-end model for the classification of images. The first step involves extracting keypoints using methods such as SIFT. The next step involves quantifying these keypoints by applying a clustering method such as k-Means [13], to cluster keypoints with similar descriptors together[1]. Based on how many keypoints of a specific image are assigned to each cluster, a feature vector (often referred to as a 'bag of keypoints') can be created that describes the image. This feature vector can then directly be used with conventional classification methods, which is the last step of this model.

[1]To emphasize, k-Means is used here as a component of the feature extraction, the feature clustering that is a part of the pipeline is separate from this. See section 1.1.4.

Figure 3: Class 2 (Elephant) Example Image

Figure 4: Class 3 (Leopard) Example Image

1.1.3 Prediction Models

For the prediction section of the image processing pipeline, two types of classification models were considered: Logistic regression [11] and the Support vector machine(SVM) [9].
Logistic regression involves a linear function e.g.:

$$p(x_1, x_2) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 * x_1 + \beta_2 * x_2)}} \tag{1}$$

This function takes the features x_1, ... x_i (where i is the amount of features), together with it's coefficients β, to make a binary prediction about the given feature values. To find the values of the coefficients that best fit a given data set, iterative approaches have to be used. Because the output of the linear function is continuous, logistic regression uses a decision boundary to determine when the outcome is negative (0) or positive (1). As the output of the function is always between 0 and 1, the decision boundary is also always between 0 and 1.
Because the image data set contains more than 2 classes, a specific variant of logistic regression was considered, this variant is called multinomial logistic regression. Multinomial logistic regression also uses a linear function for its predictions, however the output of this function contains a prediction value for each of the possible discrete outcomes that it can predict. The outcome with the highest prediction value is chosen as the final prediction.

SVM is a supervised machine learning approach that is capable of generating nonlinear mappings that separate data into different classes. SVM attempts to locate a hyperplane (decision boundary) that minimizes classification error (percentage of incorrect classifications of the given dataset), and maximizes the distance between the margin of the hyperplane . The margin is the distance to the closest point (support vector) to the hyperplane. The image dataset used is a multiclass set, so a "one-versus-one" (OvO) [16] approach is implemented. In the OvO approach, a binary SVM is trained for every possible pair of classes. In the case of 5 classes (1, 2, 3, 4, 5), 10 binary classifiers

Figure 5: Class 4 (Lion) Example Image

Figure 6: Class 5 (Wolf) Example Image

are trained. A voting mechanism is then employed to determine the final predicted classes from the collective predictions of the 10 binary SVM classifiers.

1.1.4 Clustering method

For the clustering section of the image processing pipeline, a method called DBSCAN (Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise) [5] was used. DBSCAN is different from centroid-based clustering methods such as k-Means, in that it does not use centroids at all. Rather, DBSCAN finds clusters based on local densities of datapoints. What this allows for is that DBSCAN can detect data clusters that have more complex shapes than just a hypersphere-like shape. This also means that DBSCAN does not find a set amount of clusters. It can even be the case that it finds none at all if a dataset is too scattered (datapoints are too far away from each other). DBSCAN has one especially important hyperparameter, which controls the maximum distance between datapoints for them to still belong to the same cluster. The optimal value of this hyperparameter can vary significantly between datasets.

1.2 Numerical Pipeline

1.2.1 Data

The dataset used for this pipeline consists of RNA-Seq of gene expressions of patients with different types of tumors. The dataset consists of 800 rows and 20531 columns, meaning that there are 800 samples and each sample has 20531 features. The dataset also comes with labels for every sample,

these labels consist of 4 letter abbreviations of the corresponding tumor. There is a total of 5 unique classes in this dataset.

1.2.2 Feature selection methods

The pipeline tested two feature selection methods, Principal Component Analysis (PCA) [15] and Mutual Information (MI) [10]. Initially, PCA is a dimensionality reduction method that transforms large data sets to smaller ones that still contain a large amount of information. This technique removes variables from the data set, trading accuracy for simplicity. Below are the steps for dimensionality reduction using PCA.

- Firstly, the data are standardized; each feature has the mean subtracted from it and then divided by the standard deviation. This gives our data a mean of 0 and a standard deviation of 1.
- On the second step, the covariance matrix is calculated. It calculates the relationship between all pairs of features in the dataset. The formula is $\text{Cov}(X,Y) = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - \bar{X})(Y_i - \bar{Y})$ Where X and Y are two features, N is the number of data points and $\bar{X}$ and $\bar{Y}$ are the means of the respective features.
- On step three the eigenvectors and eigenvalues are calculated from the covariance matrix and are then sorted in descending order. Then depending on the N principal components selected the top N components are selected.
- On step four the selected components are used to project the principal components onto the data. This is done by multiplying the original data matrix by the matrix of selected eigenvectors

PCA is able to handle high-dimensional data, it reduces the complexity and noise from the data and finds the most important features and relationships. However, the loss of information through dimensionality reduction leads to oversimplification and distortion of the data, making it harder to identify outliers and anomalies.
Looking at mutual information, is a statistical measure that quantifies the dependence between two random variables. When applied to feature selection and dimensionality reduction it finds the features most relevant in the dataset. Below are the steps taken to reduce the dimensionality of a dataset using mutual information.

- Firstly, the data are preprocessed, the data are cleaned, missing values are handled, categorical variables are encoded, and standardization or normalization of numerical features can take place. The preprocessing is done by the user as seen fit for the data.
- Secondly, the mutual information score is calculated using the below formula:

$$I(X;Y) = \sum_{x \in X}\sum_{y \in Y} p(x,y)\log\left(\frac{p(x)\cdot p(y)}{p(x,y)}\right)$$

  Where $I(X;Y)$ is the Mutual Information between X and Y, $p(x,y)$ is the joint probability distribution of X and Y, and $p(x)$ and $p(y)$ are the marginal probability distributions of X and Y, respectively.
- At step 3, the features are ranked with the highest MI score, and the top N features are selected. These features are then used to create a new reduced dataset.

Mutual information can capture non-linear relationships between features, allowing it to work with more complex and non-linear dependencies. Furthermore, mutual information has its roots in information theory, which allows an interpretation of the results in terms of information shared between features. However, the mutual information method is very computationally expensive, especially when dealing with many features. It also is sensitive to the size of the dataset; in small datasets, it is less reliable and prone to overfitting. Mutual information measures the relationship between two variables at a time; thus it is not able to directly capture more complex multivariate dependencies.

1.2.3 Classification methods

In the pipeline the following classification algorithms were compared, Random Forests [8], Gaussian Naive Bayes and K-Nearest Neighbors (KNN) [6]. Random Forest is a classification algorithm that makes predictions with the use of an ensemble of decision trees. The following steps explain the algorithm.

- Bootstrap sampling, a bootstrap sample is drawn from the training data.
- A decision tree is grown to this sample by selecting a number of features and picking the best feature amongst them to split the sample into two daughter nodes. This is done recursively until the minimum node size is reached.
- Repeat these two steps B times to get a forest of decision trees.
- Majority voting, the forest makes a prediction by having all trees make a prediction on the input data. The class that is predicted the most will be the final prediction.

Due to the use of multiple decision trees, Random Forest becomes less prone to overfitting. The use of multiple decision trees also reduces the bias in the classification.

Gaussian Naive Bayes is one of many variants of the Naive Bayes classification algorithms that are based on the Bayes' Theorem [1], in which conditional probabilities from prior knowledge are used to make predictions on new data. Gaussian Naive Bayes in particular is based on continuous variables assumed to a Gaussian distribution [7]. It is also named 'Naive' as it assumes that the features are independent which in nature rarely occurs. The steps below explain how the algorithm works.

- First prior probabilities are calculated for each class without having any information on the features of the data.
- Then the mean and standard deviation of the features are computed.
- These values are then used in the Gaussian probability density formula to calculate the likelihood of a feature value given a class.

$$P(X|c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} e^{\frac{-(x-\mu_c)^2}{2\sigma_c^2}} \tag{2}$$

  Where:

  - $P(X|c)$ is the likelihood of observing the feature value X given class c.
  - σ_c is the standard deviation of the feature given class c.

  - μ_c is the mean of the feature given class c.

- New data can then be predicted using the Bayes Theorem' and the values of the probability density formula above.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{3}$$

The KNN algorithm is an algorithm that makes use of a distance metric to compute the similarity between data points. The steps of the algorithm are as follows.

- Calculate the distance (e.g. Euclidean distance) between the new data point and all data points in the training set.
- Order the data points of the training set by distance to the new data point.
- Take the k closest data points and count the occurences of each class.
- The class with the highest frequency is the predicted class for the new data point.

1.2.4 Clustering Method

The clustering method that was chosen is Bisecting K-means [19]. It is a hierarchical and partitional clustering approach, allowing it to find clusters of any shape and size. However, clusters of comparable sizes are usually created from its usual partitioning methods. Below are the steps Bisecting K-means takes to split the data into k clusters.

- Its first step is to create one cluster with all the data points and take as input the k amount of clusters to find.
- In the second step, it either finds the cluster with the highest amount of data points or tries to minimize the within-cluster sum of squares or a combination of them. The chosen cluster is then set as the cluster to be split.
- At its third step, it splits the selected cluster using a K-means approach with K=2 and then recalculates the centroids of the two newly created clusters.
- At step four, it now re-assigns every data point to its nearest cluster based on the updated centroids
- Lastly, it checks if it has created k amount of clusters; it repeats from step 2 until either the set amount of clusters has been created, or a common stopping criterion is met, which could be a maximum number of iterations, a minimum cluster size, or a change in the cluster assignments that falls below a certain threshold.

2 Methods

This section describes how the two finalized pattern recognition pipelines were built. Additionally, it is also described how the best components for each of the steps was found.

2.1 Numerical pipeline

2.1.1 Feature selection

The pipeline compared the feature selection methods PCA and MI. The two methods were compared with different values for n, where n is the number of features that are selected. The performance of the models was tested with the Random Forests and Gaussian Naive Bayes classifier. Figure **??** shows the model accuracy for PCA and MI with both classifiers. The methods show similar performance however, PCA does perform better with a smaller number of features. PCA was also a lot less computationally expensive compared to MI. For the final pipeline PCA was therefore used as the feature selection method. Before applying PCA the final pipeline also splits the data in 5 for k-fold cross-validation.

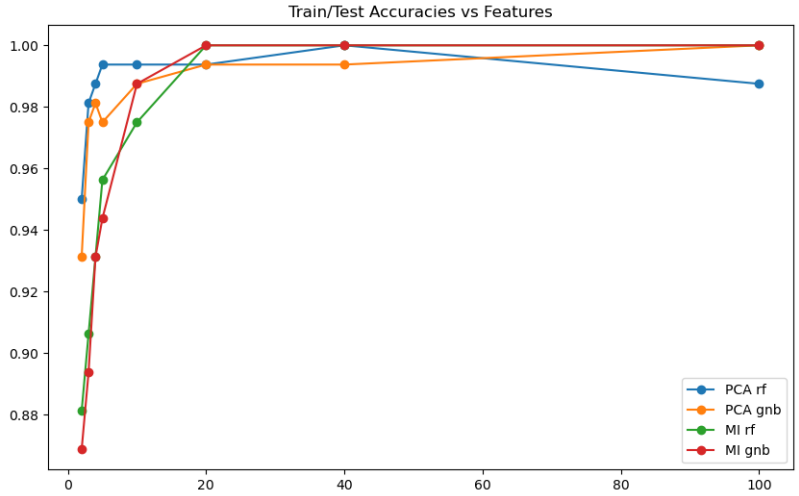

Figure 7: Accuracies of PCA and MI reduced dataset with RF and GNB against n features

2.1.2 Classification

For the classification of the pipeline, Random Forest and Gaussian Naive Bayes were compared to the KNN algorithm [6]. The classification methods were compared on the full dataset and the best-reduced dataset.

The confusion matrices show that Random Forest, Gaussian Naive Bayes and KNN perform similarly. However, Gaussian Naive Bayes does perform a lot worse without PCA reduction. For the final pipeline Random Forest is therefore used.

2.1.3 Clustering

For the clustering method Bisecting K-Means was used. The hyperparameters used are as following.

- Amount of clusters 5. This is the number of clusters the Bisecting K-Means method produces and has as a halting mechanism.
- For the initial cluster centers, k-means++ was used as input as it speeds up convergence.

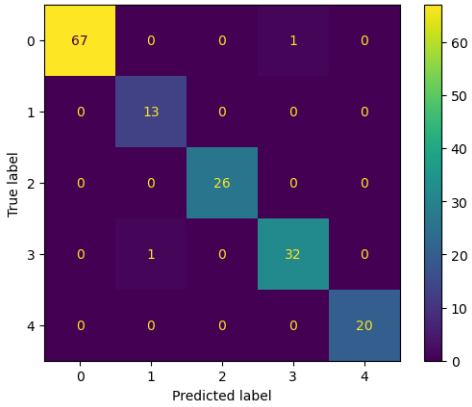

Figure 8: Confusion Matrix for the Random Forest method with PCA (3 features) reduction

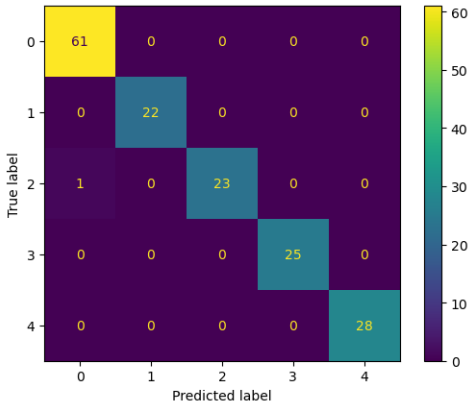

Figure 9: Confusion Matrix for the Random Forest method without PCA features) reduction

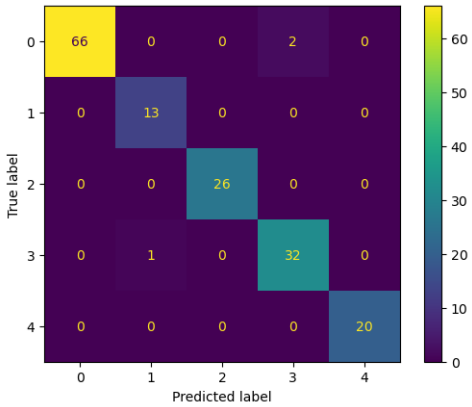

Figure 10: Confusion Matrix for the Gaussian Naive Bayes method with PCA (3 features) reduction

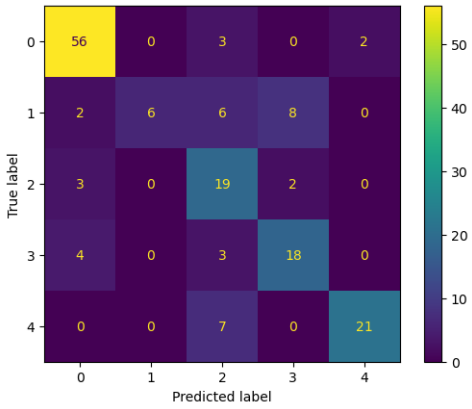

Figure 11: Confusion Matrix for the Gaussian Naive Bayes method without PCA reduction

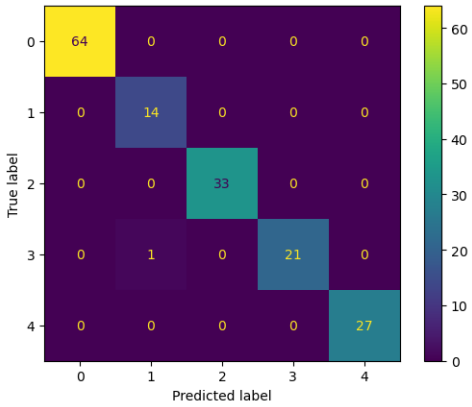

Figure 12: Confusion Matrix for the KNN method with PCA(3 features) reduction

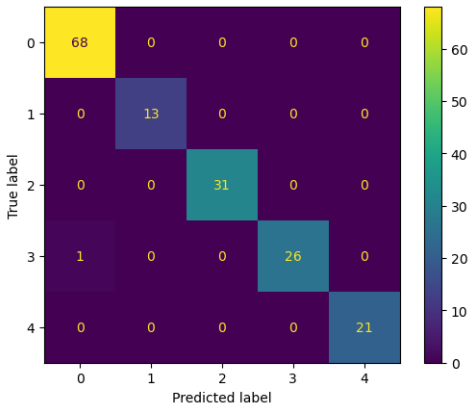

Figure 13: Confusion Matrix for the KNN method without PCA reduction

- Random state 16 was selected. An integer number was selected to make the randomness for centroid initialization in inner K-Means deterministic.
- The maximum number of iterations of the k-means algorithm for a single run was set to its default value of 300. This gave the algorithm plenty of iterations to compute cluster centroids. This also functions as a halting mechanism.
- The relative tolerance regarding the Frobenius norm of the difference in the cluster centers of two consecutive iterations to declare convergence which is used in the inner k-means algorithm at each bisection to pick the best possible clusters, was set to the default value of 1e-4. This is also used as a halting mechanism.
- The inner k-means algorithm used is Lloyd's algorithm, where it partitions the clusters into well-shaped and uniformly sized convex cells.
- The bisecting strategy used was the one with the biggest inertia. This method calculates the sum of squared errors (SSE) for each cluster and selects the one with the largest SSE for bisection.

2.1.4 Grid search

From the previous subsections in Methods, PCA was chosen over Mutual Information and Random Forest was chosen over Gaussian Naive Bayes. For PCA the selected features was set to 3. The accuracy was close to 100% (98.1%), although the accuracy could be slightly higher by selecting more features, keeping the number of features to 3 allows for visualization of the data if needed. To find the optimal hyper-parameters for the Random Forest classifier grid search was performed. The hyper-parameters sets that were used are as following:

- Number of trees in the forest - [90,100,110].
- Splitting criterion - [Gini impurity, entropy, log loss]
- maximum depth of a tree - [4,5,6, None]

Grid search found that the optimal hyper-parameters were 110 number of trees, log loss as splitting criterion and having no maximum depth for the trees. The results of this are shown in section 3.1

2.2 Image pipeline

2.2.1 Feature extraction

The feature extraction part of the pipeline made use of SIFT, which was paired with k-Means to create the bags of keypoints (size = 100) (also see section 1.1.2). Feature vectors obtained were then normalized (from 0 to 1), as a final step.**(something about how it takes stuff directly from preprocesing here)** The initial k-Means clusters were obtained from the training data, the same data on which the classifier was also trained. Any further data to be processed by the feature extraction section, made use of the same training data clusters. Figure 14 shows the locations of extracted keypoints on a pre-processed sample image from each of the five classes in the dataset.
The performances of SIFT and ORB were compared using four simple bag-of-words model configurations (see table 1). For each of the configurations, a set of different bag of keypoints sizes was tested, this was done through the *n_clusters* parameter of k-Means. No hyperparameter optimization was

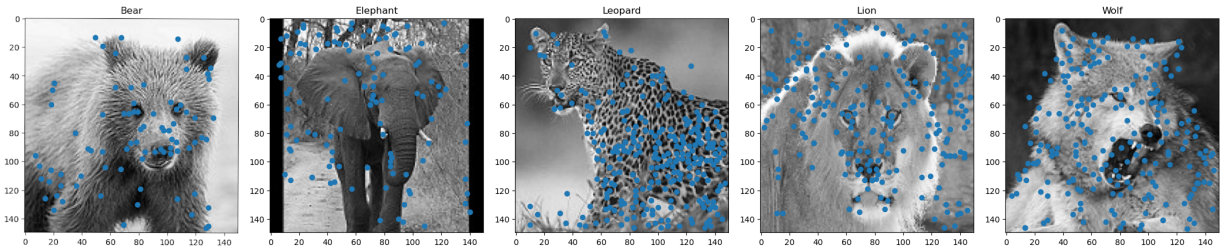

Figure 14: Keypoint locations (Blue) (using SIFT) on example images of each of the five image classes

Feature extraction methods selection setups			
Setup number	Keypoint extractor	Clustering method	Classifier
1	SIFT	k-Means	Logistic regression
2	ORB	k-Means	Logistic regression
3	SIFT	k-Means	SVM
3	ORB	k-Means	SVM

Table 1:

done on the classifiers for these feature extractor experiments. The performances were tested on a 80/20 split of the already split training set. From these experiments, the keypoint extractor with the best maximum performance was chosen, which turned out to be SIFT. The size of the bag of keypoints was also based on the maximum performance on SIFT. Refer to section **??** for further details.

2.2.2 Classification

The classification part of the pipeline made use of logistic regression, specifically multinomial logistic regression. (Hyperparameter stuff goes here) (relation to sift/orb experiment as well)

2.2.3 Clustering

The clustering part of the pipeline made use of DBSCAN. The preprocessed training data and the features extracted from that training data, were both separately clustered using DBSCAN. To paint a clearer picture of how DBSCAN clustered these two versions of the training data, the data was clustered using several different values for the maximum distance hyperparameter of DBSCAN.

3 Results

3.1 Numerical pipeline

On Train/Test Accuracies vs Variance **??** it can be seen that the accuracy reaches very high results from a variance of 0.2 or above and is stable between the 95% and 100% accuracy this can also be further confirmed with the k-fold train test accuracy vs features graph with the picked pipeline the accuracy of the kfold was 99.375% and the accuracy of the train test was 98.125%

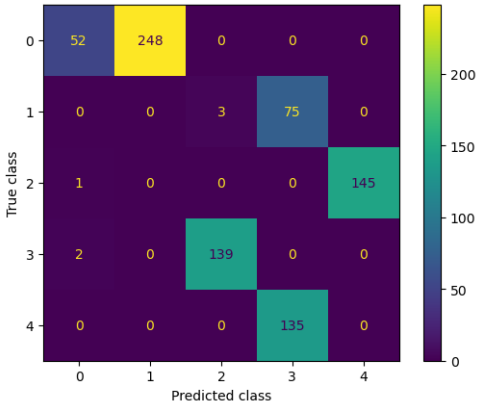

Figure 20: True vs Predicted clustering on the full dataset

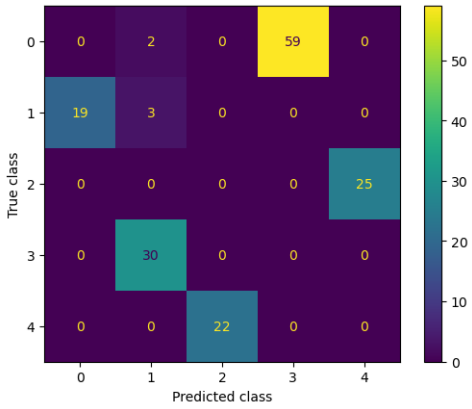

Figure 21: True vs Predicted clustering on the pca reduced dataset

2.1.4fig:kfold$_t rain_t est_a ccuracies_v s_f eatures$
Onthetwomostinformativefeaturesscatterplot??clustersofdatapointsseemtohaveformed.
OntheV ariancevsFeatures??itcanbeseenthattheamountof featuresincomparisontothevarianceisgrowingexponentia
Boththedatasetsimplification methodsaswellastheclassifiermethodsareplottedtogetherwiththeirrespectiveaccuracie

The results of the Bisecting K-Means clustering method shown in figures 20 and 21 show that the classes are not predicted correctly. The labels given are random, it's more important for the method to group all the samples of a class together. The confusion matrices show that the method does group the samples correctly for the most part.

3.2 Image pipeline

In this section, the results related to the image pipeline are discussed. Each subsection discusses the results of a different experiment.

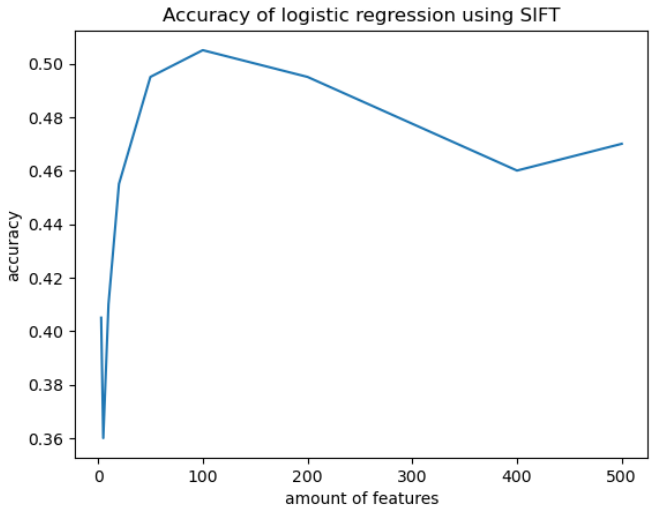

Figure 22: Accuracy for different amounts of bag-of-words features with SIFT and logistic regression

3.2.1 Feature extractor comparison

This section describes the results of the experiment described in section 2.2.1. Comparing plots: 22, 23, 24, and 25. It stands out that SIFT in combination with logistic regression, gives an accuracy of over 0.50 with a bag-of-words size of 100 features, which is the highest found accuracy from these results.
The results also show that SIFT, using an optimal bag-of-words size, gives a higher accuracy than ORB, using either classifier.

3.2.2 Clustering images vs extracted features

In this section the results from the clustering using DBSCAN of preprocessed image data and features extracted from this data is compared. The plots 26 and 27 show the results of these experiments. Both plots show similar results, with increasing maximum distance the amount of outliers (non-clustered datapoints) goes down, and the amount of datapoints in cluster 0 goes up. A line for a hypothetical cluster 1 was added in the plots to demonstrate that at no point there was a second cluster (or more) for either of the experiments.

3.2.3 Hyperparameter optimization

Hyperparameter tuning was done on the logistic regression model selected for the final PR pipeline. A grid search [16] was done on the penalty and regularization parameters. 4 possible values for penalty was tested with 'elasticnet' and 3 possible values for the regularization parameter $c \in [1, 2, 3]$. The best parameter from the gri search was implemented in the final PR pipeline for image data.

4 Discussions and/or conclusions

Discussion: state your interpretation of your findings (what do the presented results in the results section mean), perhaps comparing or contrasting them with the literature. Reflect on your actual

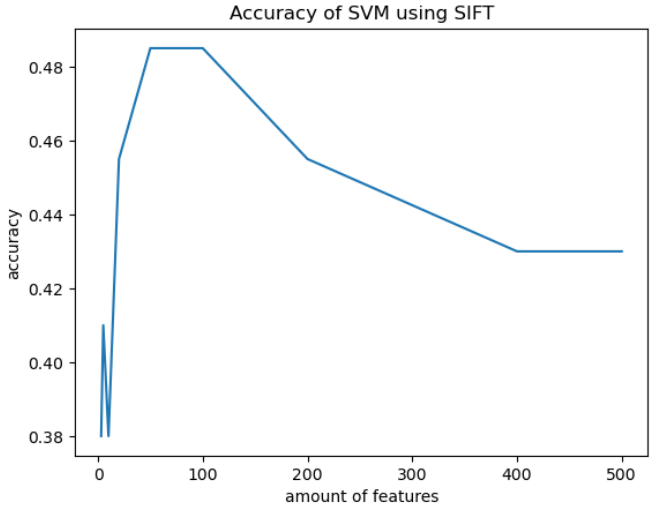

Figure 23: Accuracy for different amounts of bag-of-words features with SIFT and an SVM

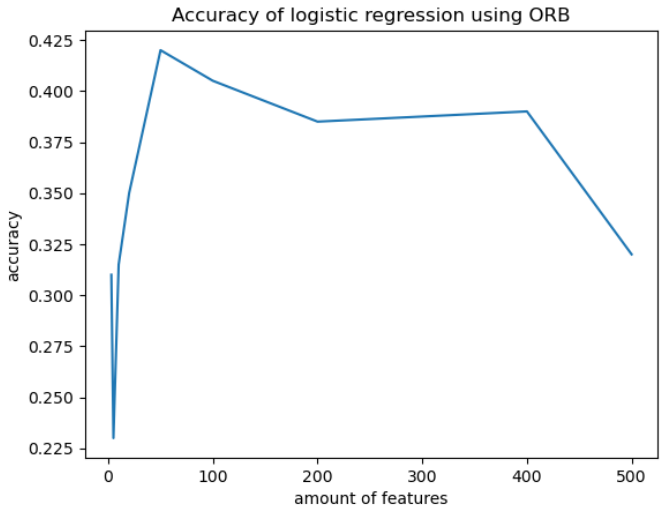

Figure 24: Accuracy for different amounts of bag-of-words features with ORB and logistic regression

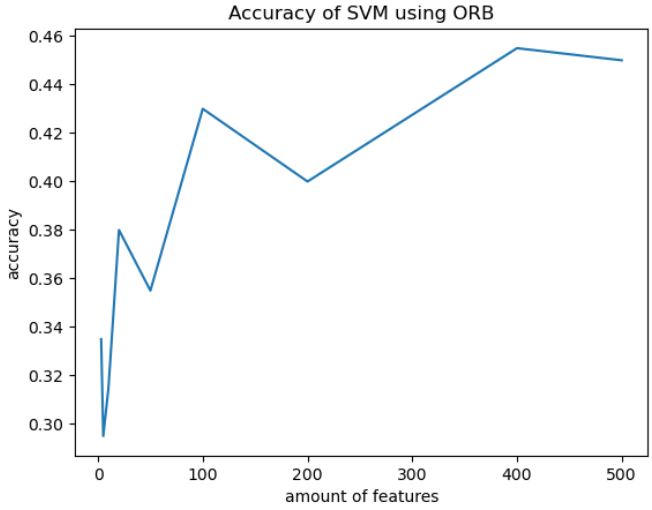

Figure 25: Accuracy for different amounts of bag-of-words features with ORB and an SVM

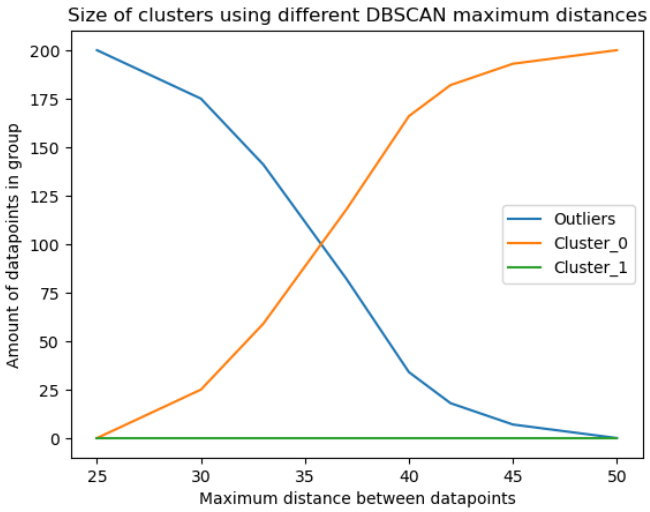

Figure 26: Sizes of DBSCAN clusters using preprocessed image data

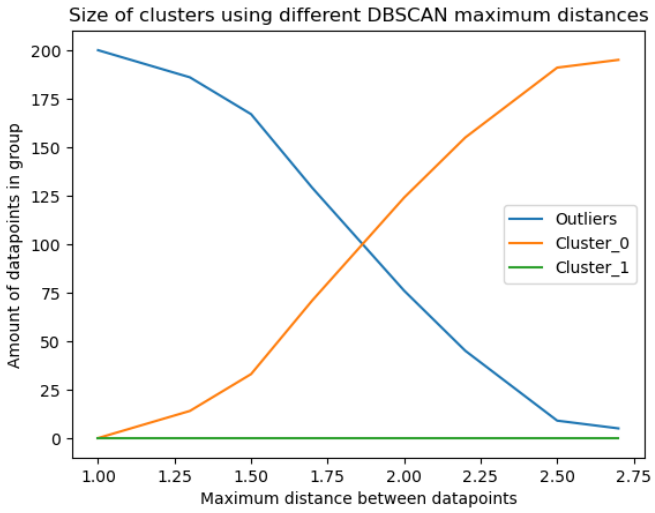

Figure 27: Sizes of DBSCAN clusters using extracted bag-of-words SIFT features

data and observations.

Conclusion: sum up your argument or experiment/research and relate it to the introduction. The conclusion should only consist of a few sentences and should reiterate the findings of your experiment/research.

4.1 Numerical pipeline

4.1.1 Discussion

Beginning by applying the PCA dimensionality reduction, when only two informative features remain, plotting them on a scatter plot allows for visualization of the data. It can be seen from **??** that there are 3 clusters, furthermore looking at the largest cluster, it can also be split into 3 different clusters that are close to each other, one at the bottom left, one at the middle right and one at the middle/right top. Even if the number of different classes are unknown, the number of clusters can be easily found by visualizing the data like this.

In addition, from the true vs predicted clustering confusion matrix graph 20, it can be seen that when the Bisecting K-means method tries to find the 5 clusters, it has a hard time finding the correct classes. The results show that the algorithm decided to split one cluster into 2 with 52 and 248 data points respectively and decided to group another cluster together with 75 and 135 data points. But when the noise in the data is removed and the data are simplified 21, the Bisecting K-means algorithm can cluster with higher precision. Through these graphs, it is also shown how two classes are split and are easily identifiable in the data and 3 clusters that are close to each other but also identifiable when looking at the centers of gravity.

Lastly, from the train/test accuracy against features **??** 7 and then comparing it to the Variance vs Features **??**, it can be seen that very few features are needed to get very high accuracy on the dataset with any method. This shows that a few features can provide a high accuracy on the dataset. This is further confirmed through the confusion matrices of the Random Forest 9 and

Gaussian Naive Bayes methods 10, it can be seen that the Random Forest method is slightly better than the Gaussian Naive Bayes but it is in the margin of error. But when the algorithms are used without the PCA reduction, the Random Forest method 8is much more accurate than the Gaussian Naive Bayes method 13.

4.1.2 Data augmentation

Data augmentation was considered to be added to the Numerical pipeline. Plain numerical data techniques such as SMOTE or SMOTE NC, are used to address the imbalance of data per cluster. A quick look in the data showed that the clusters are similar enough in size, leading to a high accuracy to begin with. More data would have provided more noise and would cause a lower accuracy in the final result.

4.1.3 Ensemble

Ensembles of different kinds of classifiers could be of used in the numerical pipeline. Ensembles are often used when the accuracy of each classifier in the ensemble is larger than $\frac{1}{C}$, where C is the number of classes in the dataset. On the numerical dataset, the minimum accuracy of eligible classifiers would have to be larger than 0.2 in this case, as the dataset contains five classes of images. Results have shown that for random forest and Gaussian naive bayes, ensembles would be usable for the numerical dataset, as they have an accuracy of over 0.2. The way this would work in the current pipeline is that all of the ensemble classifiers would take input from the same normalized random forest features, which would be produced by the feature extraction part of the image pipeline. The outputs of each of the ensemble's classifiers would then be used as input for majority voting, from which the final class prediction would then be obtained. To maximize the accuracy of the ensemble and to minimize the odds of a tie happening between votes, as many as possible different classifiers that achieve an individual accuracy of more than 0.2, should be added to the ensemble.

4.1.4 Conclusion

Concluding the gene dataset has been simplified using PCA; the clusters were then understood using Bisecting K-means and visualized using the two more informative principle components from PCA. Next, the Random Forest method was trained on the dataset to match the information to the correct output. Thus, this RNA-Sequence of genes dataset helps to find which genes are correlated with which tumors.

4.2 Image pipeline

In this section the results related to the image pipeline are interpreted. Each subsection discusses a different set of results. Furthermore the topics of data augmentation and ensembling are also discussed in relation to the image pipeline.

4.2.1 Feature extractor comparison

From the results of the feature extraction comparison, it can be concluded that for this specific image dataset, SIFT performs better than ORB using either logistic regression or an SVM. This is not necessarily what one would expect, since ORB has been shown to achieve similar results to SIFT [18]. Whether this is related to the specific dataset used or the types of classifiers, is unclear. One hypothesis is that the smaller descriptor size of ORB compared to SIFT (32 VS 128), makes it

so the classifiers have less useful information on the image data.
Despite SIFT together with logistic regression being the best combination in terms of accuracy, it cannot be concluded that logistic regression is also better than an SVM, as the SVM actually performs better than logistic regression, using ORB features.

4.2.2 Clustering images vs extracted features

The results of DBSCAN on the image data an the extracted features of the image data, seem to indicate that DBSCAN is not appropriate for this data, as it is unable to find more than 1 cluster in either the image data, as well as the extracted features data, despite the data actually having 5 different classes.
The reason that the cluster size variations happen for different ranges of maximum distance (25-50 vs 1-3), between the image data and the extracted features, is likely because the images are of a higher dimension (22500) compared to the extracted feature vectors (100).
Conclusions about DBSCAN finding clusters that separate between classes are not possible, as only one cluster could be found by DBSCAN.

4.2.3 On the final pipeline

The final pipeline accuracy of 0.6 using 5-fold kfold is decent, considering the within class difference of the image dataset, however it is far from perfect.

4.2.4 On the usage of data augmentation

The usage of data augmentation was initially considered to be a part of the image pipeline. Images could be shifted spatially, shifted in terms of brightness, or rotated, for example. This can be of benefit to image classification pipelines, that make use of a convolutional neural network for example. However, because the feature extraction part of the pipeline makes use of SIFT, data augmentation is not very useful anymore. The reason for this is because SIFT is invariant to several of these data augmentation techniques. This means that if data with these techniques applied were added, SIFT would just extract the same features as if these techniques were not applied. Practically, the training data would then just contain several duplicates of the original set's images. For this reasons data augmentation was not applied to the image pipeline.

4.2.5 On the usage of ensembling

The usage of an ensemble of different kinds of classifiers could be of use in the image pipeline. A rule of thumb for the usage of ensembles is that the accuracy of individual classifiers in the ensemble has to be larger than $\frac{1}{C}$, where C is the amount of classes in the dataset. In the case of the image dataset the minimum accuracy of eligible classifiers would have to be larger than 0.2 in this case, as the dataset contains 5 classes of images.
The results have shown that for at least logistic regression and an SVM, it would be possible to use them in an ensemble for the image dataset, as they have an accuracy of over 0.2. The way in which this would work in relation to the current pipeline, is that all of the ensemble classifiers would take input from the same normalized SIFT bag-of-words features, produced by the feature extraction part of the image pipeline. The outputs of each of the ensemble's classifiers would than be used as input for majority voting, from which the final class prediction would then be obtained. To maximize the accuracy of the ensemble, and to minimize the odds of a tie happening between

votes, as many as possible different classifiers that achieve an individual accuracy of more than 0.2, should be added to the ensemble.

Individual contributions

- Tim Eckhart (3952509): Image pipeline: feature extraction, clustering, final pipeline setup. Introduction, methods, results, discussion/conclusion on: feature extraction, clustering, classification.
- Kyriakos Antoniou (s5715881): Contributed to the numerical pipeline with Abi about 50/50 both worked and helped on each part of the pipeline. Individually wrote and gathered the Results and Conclusion and a small check of the report.
- Robert Power (s5332419): Contributed to the overall design of the image date PR pipeline. Specifically, sampling of the dataset, contributed towards development of the models using ORB and SVM, and collection interpretation of results associated with these models. With respect to the report, final editing and overview of report, and wrote the Introduction (section 1) as well as Sections (1.1.1, 1.1.3 (SVM).
- Abi Raveenthiran (s4010132): Numerical pipeline, coding mostly done together with Kyriakos. Writing Introduction and methods of numerical pipeline and checking on other parts.

References

[1] Thomas Bayes. Lii. an essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, frs communicated by mr. price, in a letter to john canton, amfr s. *Philosophical transactions of the Royal Society of London*, (53):370–418, 1763.

[2] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part IV 11*, pages 778–792. Springer, 2010.

[3] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004.

[4] Dick de Ridder, Jeroen de Ridder, and Marcel J. T. Reinders. Pattern recognition in bioinformatics. *Briefings in Bioinformatics*, 14(5):633–647, 04 2013.

[5] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996.

[6] Evelyn Fix and Joseph Lawson Hodges. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale de Statistique*, 57(3):238–247, 1989.

[7] Carl Friedrich Gauss. *Theoria motus corporum coelestium in sectionibus conicis solem ambientium*, volume 7. FA Perthes, 1877.

[8] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[9] Vojislav Kecman. Support vector machines–an introduction. In *Support vector machines: theory and applications*, pages 1–47. Springer, 2005.

[10] J Kreer. A question of terminology. *IRE Transactions on Information Theory*, 3(3):208–208, 1957.

[11] Michael P LaValley. Logistic regression. *Circulation*, 117(18):2395–2399, 2008.

[12] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004.

[13] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA, 1967.

[14] Emmanuel Okafor, Lambertus Schomaker, and Marco Wiering. Wild-Anim Dataset, 2019.

[15] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572, 1901.

[16] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[17] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Computer Vision–ECCV 2006: 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006. Proceedings, Part I 9*, pages 430–443. Springer, 2006.

[18] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.

[19] Michael Steinbach, George Karypis, and Vipin Kumar. A comparison of document clustering techniques. 2000.

[20] Maxime Vidal, Nathan Wolf, Beth Rosenberg, Bradley P Harris, and Alexander Mathis. Perspectives on Individual Animal Identification from Biology and Computer Vision. *Integrative and Comparative Biology*, 61(3):900–916, 05 2021.

A Supplementary materials (optional section)

Use this appendix if you need to show some additional results, graphs, etc.