Creating Pattern Recognition Pipelines using Traditional Machine Learning methods

Rik Vegter (S3147495)[1], Sarantis Doulgeris (S4928628)[1], Jeroen van Brandenburg (S3193063)[1], Pedro Rgz. de Ledesma (S4745779)[1]

Group [11]
Pattern Recognition (WMAI021-05) 2021-2022.2B
[1] Artificial Intelligence Department, University of Groningen
f.j.vegter.1@student.rug.nl
s.doulgeris@student.rug.nl
j.c.van.brandenburg@student.rug.nl
p.rodriguez.de.ledesma.jimenez@student.rug.nl

January 27, 2022

Abstract

This paper aims to create two machine learning pipelines, one for an image dataset, and one for a numerical dataset. We refrain from state-of-the art neural network techniques and instead use transparent traditional machine learning methods in order to create an explainable pipeline. We provide a thorough exploratory data analysis and base the decisions in our pipeline on the outcome. The pipeline for the image dataset uses SIFT combined with the bag-of-visual words method for feature extraction and consists of an ensemble classifier in order to classify wild animals. The pipeline for the numerical dataset is a computer aided diagnosis (CAD) where LDA combined with a random forest is used in order to classify different kinds of tumors. The numerical dataset pipeline achieved an accuracy of 100% where the image dataset reached an accuracy of 67.0%. We analyse in-depth the underlying principles of the different components

1 Introduction

The field of Pattern Recognition covers applying various Machine Learning techniques in order to extract interesting relations and relevant patterns from datasets. In practice, this comes down to giving a label to a certain input value. A good way to understand what Pattern Recognition is, is to give a distinction between Machine Learning and Pattern Recognition, which are two terms that are often used interchangeably. Machine Learning uses models to learn from data and make predictions based on these models. The field of Machine Learning relies heavily on statistics. 'Pattern recognition can be defined as the classification of data based on knowledge already gained or on statistical information extracted from patterns and/or their representation' [4]. This subfield of Machine Learning deals with detecting insightful regularities in data that might be hard or time-consuming to detect manually. As mentioned by our supervisor of this project (Maruf Dhali), 'all

machine learning systems are applied in pattern recognition systems, but not all pattern recognition systems are machine learning algorithms.'

Extracting patterns from big datasets has many applications. One of them is Computer Vision, which is recognizing objects or other characteristic parts of an image and being able to classify them. Previous research has had many successes with fruit recognition [26] and smile detection [1]. However, these are just two examples of the growing academic literature of Pattern Recognition.

The previous two examples give labels to images which is the main part of Computer Vision. However, many datasets are not represented by images but are numerical datasets. The Pattern Recognition literature also provides many successful examples of successful pipelines for numerical datasets. For example stock market prediction [8] and computer aided diagnosis (CAD) [18].

In the last few decades, one of the most popular Machine Learning techniques has become the use of Neural Networks. While the results of these methods are outstanding in computer vision [9] [10], one of the main issues with these algorithms is that the mechanics behind the algorithms are opaque. The recent increase in computer power allowed the implementation of many large neural networks. Often, engineers want to know the reasoning of the trained models. However, with these large Neural Networks this becomes a challenging task. This is known as the black-box problem [7].

Therefore, Machine Learning engineers occasionally seem to refrain from these methods and rely more on traditional machine learning methods in order to have a better understanding of the mechanics of an algorithm that lead the model to a certain classification. Our definition of traditional machine learning methods are the classifiers that were used before neural networks became state-of-the art. Examples are support vector machines, random forest, regression and K-NN.

In this study we create two classification pipelines, one for a numerical dataset and one for an image dataset. We compare the difference in approach and results when using traditional machine learning methods, meaning we are not using a Neural Network approach. Secondly, we try to find interesting insights within the dataset and try to interpret these. In the methods section we will describe both pipelines one by one and explain all steps along the way that were taken. Next, we will present the results per pipeline. Finally we will give a discussion where we talk about the differences in approaching the two different types of datasets and argue about the use of traditional machine learning methods in comparison to the use of neural networks techniques.

2 Methods

As previously mentioned, the aim of this paper is to create two pipelines for a numerical data set and an image data set. In this section we will analyze the data sets used per pipeline and provide a description of each component of both of the pipelines.

2.1 Image data pipeline

The goal of this part of the project is to create a pipeline that is able to classify images of animals from the *BigCats* dataset provided by the University of Groningen. This dataset contains images of the living members of the genus *Panthera*, namely the *Cheetah*, *Jaguar*, *Leopard*, *Lion* and *Tiger*.

Data

The dataset consists of jpg/jpeg images of big cats in the wild, meaning the background varies per image. This can be seen in Figure 1 which shows an example image per class. Table 1 shows the number of images per class. Every image in the dataset has a different resolution. Additionally the width and height vary per image. This is also visible in Figure 1 where the Leopard and Tiger

Class	Number of images
Cheetah	38
Jaguar	30
Leopard	31
Lion	32
Tiger	39

Table (1) This table shows the number of images per class that are in the *BigCats* dataset

Figure (1) One example image per class taken from the *BigCats* dataset. *From top-left to bottom-right: Cheetah, Jaguar, Leopard, Lion, Tiger*

images are in portrait mode while the Cheetah, Jaguar and Lion are in landscape mode. The dataset also contains duplicates and we also found that some images were in the wrong folder. All images are in color with the exception of one lion. The images also contain infants of the species. Finally, some images contain multiple animals all belonging to the same species.

Preprocessing

The previously described data contains several problems we need to cover in the preprocessing part of the dataset. We handle the 5 duplicates in the same class by simply removing them. The reason we remove duplicates from our dataset is so that the trained model will not become biased towards the duplicated samples. As previously mentioned, some images were in two classes. Based on human classification, we put these animals in the correct class in order to prevent giving the wrong label to images. Both of these preprocessing tasks were done manually. The images were not scaled to all have the same size. This is not necessary because we use the SIFT algorithm (which will be explained later) for feature extraction, which is scale invariant. Since SIFT is invariant to color we also gray-scale the images.

SIFT features are invariant to scaling and rotation. However, to our knowledge, literature does not combine SIFT features with flipped images. Flipping an image means that we multiply all x-values in the image by -1 meaning the image flips over the x-axis. We think this might make the classifier more robust to similar key points that are flipped in different images. To prevent bias for the small dataset, we decided to only use this augmentiation on the train set.

Feature extraction & dimension reduction

In order to extract features from the images we use the Scale-Invariant Feature Transform (SIFT) algorithm [19]. This algorithms first tries to find key points. SIFT does this by taking the image and blur it with Gaussian blurs with different magnitudes. These blurred images are subtracted from each other. Then, these different levels of blurred images are stacked on top of each other. Points where the difference in intensity is distinct in comparison with the neighbors for all levels, can be considered as key points. The next step of SIFT is to describe these key points. This is done by looking at the local neighbourhood of the key points. The key points are broken down into smaller neighborhoods and the gradients in these smaller areas and computed. Note that gradients are robust with respect to viewpoint changes making the descriptor invariant to rotations. Then the gradients are collected into histograms which simply count the number of times a gradient and their magnitude occurs. These smaller regions are 4 by 4 regions meaning we obtain 16 histograms. Every histogram is discretized in 45° orientation meaning 8 bits. This results in 128 values describing a key point.

The number of key points that are obtained through the SIFT algorithm varies per image. Since we need to obtain feature vectors with the same number of dimensions we use the bag-of-visual-words method [12]. For this method we create a number of clusters (N) using K-means clustering. Images are described by a histogram of the number of key points that attach to the clusters. Some images might contain more key points than others. However, the classification should be invariant to the frequency of the key points. The information lies in the distribution of descriptors over the clusters. Hence, we normalize the constructed histogram by the total number of key points in the image. This entire feature extraction method (SIFT i.c.w bag-of-visual-words) results in N-dimensional feature vectors describing an image. An advantage of the bag-of-visual words method is that we can specify the dimensionality of our data manually by manipulating the number of clusters, N. Hence, this parameter is also included in the grid search. For the possible values that we tested, see Table 2.

In order to reduce the dimensions of the histogram we use UMAP [20] which is a recent popular dimension reduction technique. Figure 2 shows the projection of the first 2 UMAP dimensions taken from the extracted histogram using the bag-of-visual words method per image. While this projection of the data only maps two dimensions, we can see that some areas emerge with concentrated similar labels. For example, in the top right corner of Figure 2 we see a lot of datapoints belonging to a tiger. However, we also observe that there are areas where datapoints belonging to different classes are close to each other.

Classification

For classification we use a Support Vector Machine (SVM)[11] and a Logistic Regression model [21], which we compare to a K-Nearest Neighbors (K-NN) model [13] as our baseline. We use the Support Vector Machine and Logistic Regression because literature shows good results for image classification using an SVM [17, 23, 24]. The K-NN classifier provides a baseline that we can improve on. To optimize the classifiers we use a grid search for several parameters. These parameters are shown in Table 2. The selection of the parameters is based on the classification accuracy. Besides the parameters for the classification models, we also consider the number of clusters during the K-means clustering as a hyperparameter for the grid search. This search is also based on the classification accuracy.

For the classification, we thus use a reduced dataset consisting of histograms representing the frequency of key points belonging to clusters (bag-of-visual words). Logistic Regressions and SVMs are not suitable for raw images, because the images are different shapes and sizes. Furthermore, obtaining a reasonable performance solely on pixel values is impossible, since these classification

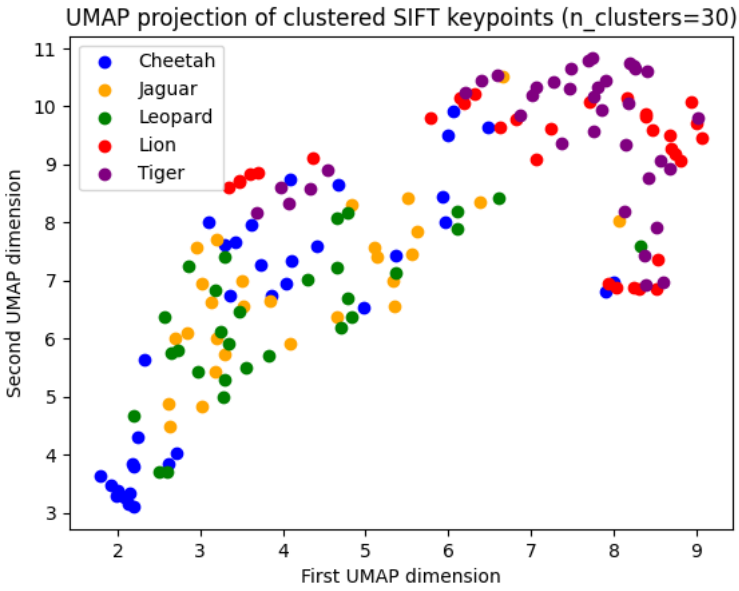

Figure (2) This figure shows the first two UMAP dimensions on the training data for 30 clusters

Classifier	**Parameter**	**Possible values**
Support Vector Machine	Norm of penalty	0.01, 0.1, 1, 10, 100, 1000
	Kernel function	Linear, rbf, poly, sigmoid
	Gamma	Scale, auto
Random Forest	Size of forest	10, 20, 30, ..., 100
	Max depth	3, 4, 5, 6, 7
Logistic Regression	Norm of penalty	L1, L2
	Tolerance stopping criteria	0.0001, 0.001, 0.01, 0.1
	Inverse of regularization strength	0.01, 0.1, 1, 10, 100, 1000
K-NN	K	1, 3, 5, 7, ..., 21
KMeans	Number of clusters	10, 20, ..., 50

Table (2) This table shows the fine-tuned parameters of the algorithms for the grid search of the image pipeline

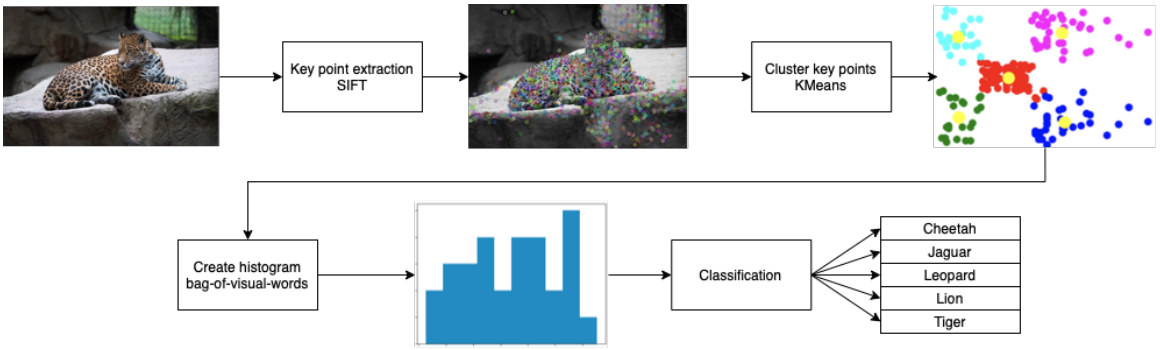

Figure (3) Pipeline for *BigCats* dataset. The classification can be either SVM, K-NN or an ensemble classifier.

algorithms are not designed to extract features on their own (in contrast to e.g. Neural Networks). By extracting features, we can ensure certain invariance, such as scale, rotation, or location of the ROI. Hence, we omit classification and clustering on raw images.

Ensemble

An ensemble classifier can create a lower variance and a lower bias and might create a deeper understanding of the data. Therefore, we also create an ensemble classifier consisting of a Support Vector Machine, Random Forest [5] and Logistic Regression. The voting rule for this ensemble classifier is the plurality rule which takes the majority vote. If all three classifiers produce a different output, the output of the classifier with the highest accuracy is selected.

Evaluation

In order to evaluate the performance of the pipeline we obtain the accuracy and analyze the confusion matrix. The accuracy and confusion will be obtained by a 5-fold cross validation. During the training phase, only the train set is used for the clustering of the SIFT-features. This prevents a bias in the K-means model.

The entire pipeline is visualized in Figure 3.

2.2 Numerical data pipeline

In this section we will describe the pipeline created for the numerical dataset. The classification task is to detect different types of tumors based on sequences of RNA of patients. The dataset which was used has been provided by the University of Groningen. It contains different samples of RNA-Sequences of patients that have been diagnosed with different kinds of cancer.

Data

The dataset contains a total of 801 samples, representing patients. Each sample is characterized by a sequence of 20530 genes which are used as features . These features have float attribute values. Every sample provided is categorized in 5 different types of genomes: BRCA, KIRC, LUAD, PRAD

and COAD. In Table 3 the distribution of the genomes is shown. There are no missing values in the dataset.

Type of tumor	**Occurrences in the dataset**
BRCA	300
KIRC	146
LUAD	141
PRAD	136
COAD	78

Table (3) Number of genomes.

Preprocessing

Several preprocessing steps have been applied to the data before passing them to the model. First of all, a normalization process of the data has been done. This step is necessary since not all the features are in the same scale and could affect the performance of the model. Therefore, the scaling of the data without distorting differences in the range of values is a required pre-step. Secondly, a visualization algorithm suitable for high dimensional data has been applied in order to try to obtain insights of the data. The algorithm applied is the t-distributed Stochastic Neighbor Embedding algorithm (t-SNE) [16]. It is an unsupervised non-linear dimensionality reduction algorithm used for data visualization. It transforms high dimension space in low-dimensional space of two or three dimensions, which makes it appropriate for our dataset. The t-SNE algorithm tries to preserve the neighborhood of the points and reflect it in a low dimension space. It is stated that it works on high dimensions but our dataset is described by very high-dimensional feature vectors. Therefore, before applying the t-SNE algorithm it is essential to reduce the dimensions, while maintaining as much information possible. This is done by applying Principal Component Analysis (PCA) on the dataset. PCA finds all the orthonormal vectors of the original data. These vectors are computed using eigenvalue decomposition of the co-variance matrix of the data. Based on that, maximum number of components depends on the number of examples and features and it is equal to $min(n-1, p)$ where n in the features and p the number of patterns. After applying PCA and t-SNE we visualized that there where a small number of outliers in the dataset (Figure 8). So we used the Local Outlier Factor (LOF) algorithm for detecting and removing them in order to not cause a problem in the further analysis. This algorithm is an unsupervised procedure which computes density deviation of a k-dimensional point with its neighbors. Outliers are the samples that have significant less density with respect to their neighbors.

Feature extraction/ dimensionality reduction

For this part, two algorithms are going to be evaluated for the dimensionality reduction step based on the classification performance. An unsupervised algorithm, PCA, and a supervised one, the Linear Discriminant Analysis (LDA). Both of the algorithms try to preserve the variance of the data while reducing the number of features of the dataset. Although both methods try to retain the variation present in the dataset, the LDA focuses on maximizing the class variation, which make it perfect for a dataset with high level of clustering like the one provided for this assignment. Furthermore, the LDA reduces dimensionality from the original number of features to $C-1$ features, where C is the number of classes while the PCA to the number of samples as maximum. Both algorithms are going to be evaluated in the numerical pipeline and optimized based on the performance of the models.

Classification

For classifying different types of tumors based on RNA sequences of gene expressions of patients we have selected two algorithms to predict the categories of samples. We will compare these to a K-NN algorithm as a baseline. We implemented the Decision Tree algorithm and the Random Forest algorithm, which are multiple trees ensembled in one classifier. Both algorithms have been chosen as suitable for this feature space, due to the clustering of samples of the same category. This makes it easier for the two algorithms to delimit the feature space in classification regions. This tell us that different types of tumors have distinctive differences with the others, making easier to cluster them.

Clustering

K-Means algorithm has been selected for clustering the dataset. As an unsupervised clustering method, it will identify similar points of the dataset, and gives insights of the distribution of the dataset based on the proximity of the points. The clustering was applied to the original data and the best-reduced data (LDA dataset) of the selected features. Clusters are evaluated with two methods, with the elbow and the Silhouette methods. The number of cluster are selected based on the outputs of this two techniques.

Evaluation

The evaluation scheme involves the splitting of that dataset into a training dataset which is 80% of the original, and the remaining examples as the testing dataset. The metrics that we are interested is the models accuracy (percentage of correct classifications), AUC (provides an aggregate measure of performance across all possible classification thresholds) and the F-score metric which the harmonic mean of the precision and recall.

Summary of the pipeline

In the Figure 4 displays an overview of the complete pipeline, since the data is provided till the model validation. However, there is another flow where the model is cluster and evaluated.

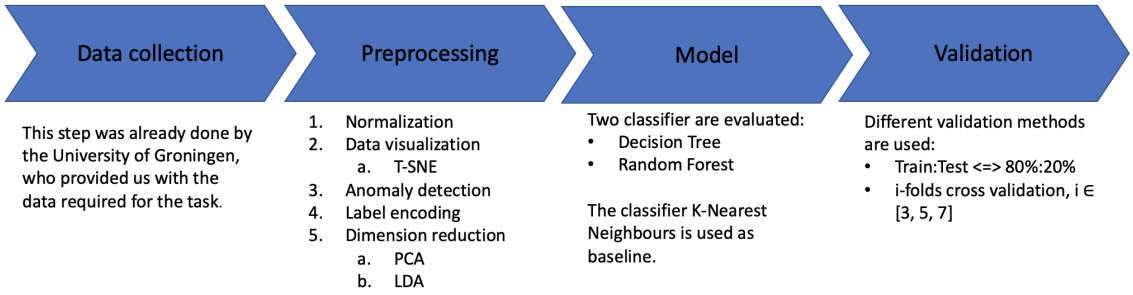

Figure (4) Overview of the flow of the data.

3 Results

In this section we will present the results per pipeline. We start with the Image pipeline and then we will discuss the numerical pipeline.

Classifier	Accuracy
SVM	65.2%
Random Forest	62.1%
Logistic Regression	65.4%
K-NN	57.8%
Ensemble Classifier	67.0%

Table (4) Accuracy per classifier on the Image dataset based on 5-fold cross validation

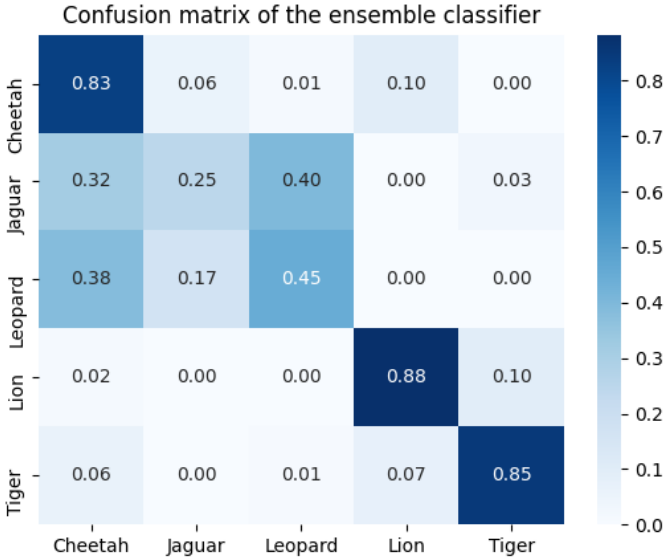

Figure (5) Normalized confusion matrix for the ensemble classifier for the image dataset over 5-fold cross validation. The true labels are on the y-axis and the predicted labels on the x-axis

Class	F1 score
Cheetah	0.64
Jaguar	0.34
Leopard	0.48
Lion	0.86
Tiger	0.86
Mean	0.63

Figure (6) F1 scores of the ensemble classifier for the image dataset over 5-fold cross validation

3.1 Results Image Classification

The results are gathered after the optimization of the classifiers using the grid search. An SVM is used with a penalty norm of 100, a scale RBF kernel function and a scaled kernel coefficient. The Random Forest is initialized with 80 trees with a max depth of 7. The Logistic Regression uses an L2 norm with a inverse regularization strength of 1000 and a tolerance of 0.0001. For the KNN, K was set to 11. All the classifiers were trained on the clustered SIFT features determined by the K-means algorithm, with K=30. The average accuracies of the 5-fold cross validation are shown in Table 4. From this table we can see that the ensemble classifier performs best with an accuracy of 67.0%. Therefore, we will analyze the results of this classification method in more detail.

In the Methods section we talked about how a classifier might benefit from training on flipped images. We tried this with our classifier with the highest accuracy which is the ensemble classifier. This resulted in an accuracy of 65.3% which is not an improvement of the accuracy without data augmentation. Therefore, the results presented in the rest of this section are the results obtained without data augmentation.

Figure 5 shows the confusion matrix of the ensemble classifier, over a 5-fold cross validation. The best predicted classes are the Cheetah, Lion and Tiger, as can be seen on the diagonal. What stands

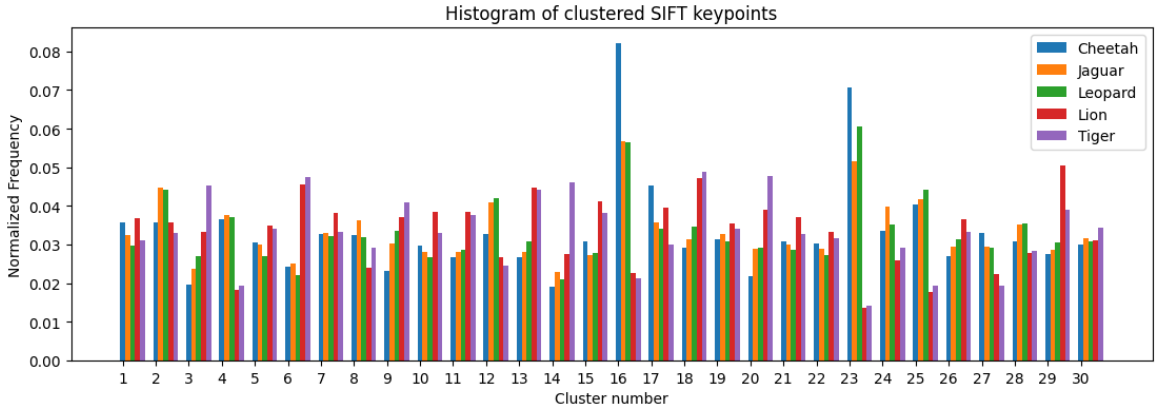

Figure (7) This image shows the the average clustered SIFT key points of all the images per class. This k-means model, with k=30, is trained on the whole dataset after preprocessing.

out, it that the Leopard is only classified correctly 45% of time time. The recognition of the Jaguar is even worse, with a true positive rate of 25%. They are both often falsely classified as each other or as as a Cheetah. Furthermore, the ensemble often predicts the class Cheetah and thus might be biased towards that class. While having a high true positive rate, as shown in Figure 5, the class Cheetah has an average low F1 score, as shown in Table 6.

As mentioned before, the features used during the training of the classifiers are based on the SIFT features, which are classified to 30 clusters using the K-means clustering algorithm. After normalization, every image is represented by a similar histogram of length 30. To improve our understanding of the created features, we calculated the average histogram per class. These are shown in Figure 7. This bar plot shows that a high frequency of key points for a specific cluster can indicate a specific class. For example, if a newly presented image has a high frequency for cluster number 14, it will likely belong to the class Tiger. Furthermore, the bar plot also shows similarity between certain classes. Cluster 23, for example, represents a feature which is more frequent in images of Leopards, Cheetahs and Jaguars, then for Tigers or Lions. Similar splits between these classes can be observed in many clusters (4, 6, 11, 16, 23, 25, 27 and 29). The repeating frequency split between Leopards, Cheetahs and Jaguars versus Tigers and Lions, indicate the presence of similar visual features.

3.2 Results Numerical Classification

During the preprocessing phase and data analysis, the visualization algorithm applied (t-SNE) gave us a perspective of the distribution of the data. After applying the algorithm the data was displayed in five different distinct clusters (Figure 8). Each cluster has samples of one category (excluding some minor outliers). This gave us many insights for the selection of the dimensionality reduction and the classification algorithms.

The optimization of the parameters of dimension reduction and classification algorithms starts with a first approach of their initial parameters to afterwards evaluate their performance and tune them. The first approach of the LDA is a reduction to the number of classes - 1 (max number of features reduced), when the variance capture by the algorithm is the maximum enable (100%). The reduction algorithm is already significant reducing to the maximum number of features of the LDA. On the other hand, for the PCA, the variance of all the components was computed (Table5). It can

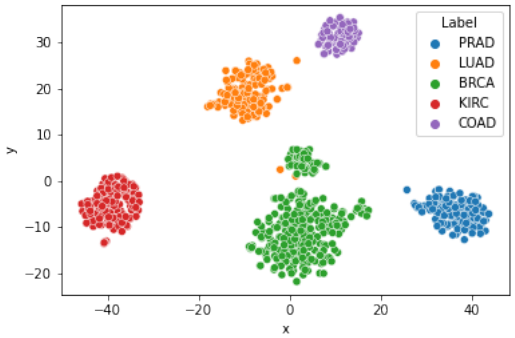

Figure (8) In this figure we can see the result of applying the t-SNE algorithm for data visualization where the data has been reduced to two features, x and y. The data has been labeled afterwards.

be seen that the 10 largest principal components explain 60% of the variance. As a start point, an amount of variance of 70% was selected thus the number of features will get reduce from 20530 to 86.

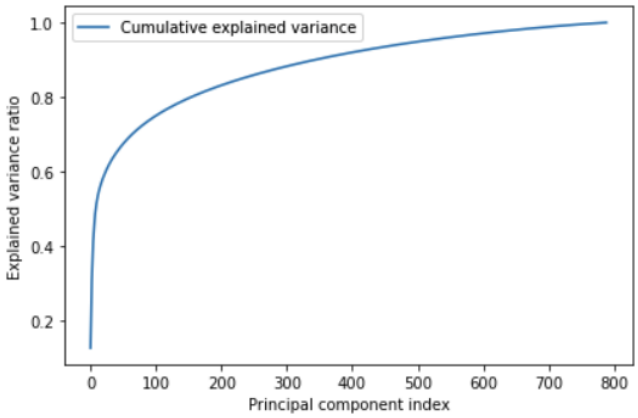

Figure (9) In this figure are shown accumulative variance depending on the number of features of the reduced dataset.

Despite the significant reduction of the PCA algorithm, to increase the variance captured will require to add a large number of features while the accumulative variance will increment less than 1% (Fig.**??**). The parameters of classification algorithms were initialized with an initial approach as well (Table 5).

In our first approach, where the performance of each of the reduction algorithms were computed with each of the classification algorithms we obtained high accuracy(Table 6). The different configuration between algorithms presented in Table 6 returned a maximum accuracy of around 99.3%, which makes the starting point of optimization in high levels. The results obtained agree with the preprocessing visualization insights or the results obtained with the clustering algorithm, where in both steps the data was presented distributed by categories. Despite this, additional optimization

Classifier	**Parameter**	**Value**
Decision tree	Criterion of split	Entropy
	Splitter	Best
Random Forest	Number of trees in the forest	100
K-NN	K	9

Table (5) This table shows the value of the parameter of the classifiers with the first approach

tasks are performed in order to evaluate the number of features optimal without affecting the model its performance and the algorithm parameters tuning.

First approach performance	PCA	LDA
Decision tree	93.5%	99.3%
Random Forest	99.3%	100%
K-nearest neighbors	99.3%	100%

Table (6) Accuracy of the classification with different reduced dataset and classification algorithms.

We evaluated how the performance of the model changes by decreasing the number of features. The configuration PCA-DT reaches a relatively steady value for five features and after (Figure 10). On the other hand, we evaluated the performance of LDA-DT configuration where the performance is maximized with four features (Figure 11). This is also expected because we keep the most information possible. The performance of the PCA-RF reach a steady value (99%) with seven features, while the configuration LDA-RF reach a maximum with four features(Figure 10 and (Figure 11). Finally, the configuration PCA-KNN reach a steady value with seven features and the LDA-KT with three features (Figure 14 and Figure 15). Due to the nature of this dataset, K-NN is the best out of all.

The results show us that the LDA configuration results in a higher accuracy with less number of features compared to PCA. The LDA algorithm is more suitable for the task due to the well distributed dataset. As we explained in previous section, LDA maximizes the variance between categories. We compute the performance of the models with the best-reduced dataset obtained with the dimension reduction algorithm LDA. The accuracy of the DT algorithm with four features is 99.3%, with just one misclassification. The RF did not have a single misclassification. This matches with our initial hypothesis. The accuracy of the K-NN was 100% as well. We also calculated the performance on original dataset, obtaining 92% with the DT classifier, 100% with the RF and K-NN classifier. The DT model performance was improved by tuning their parameters, increasing this way the performance to 100%.

In the clustering step with the K-means algorithm of the original dataset, the elbow method shows it is optimal to select four or five clusters. These values minimize the within-cluster sums of squares (WCSS). With the Silhouette validation algorithm, we compute the average score for different number of cluster. The peaks are found with five, six and seven clusters. We omit discussing using seven clusters, because we obtain the same Silhouette score using less clusters. The maximum score converges with the maximum number of clusters that is equal to the number of samples. Using a higher numbers of clusters is not optimal, following Occam's razor. We computer the Silhouette score of the selected clusters for each point for the (16 and 17). Each point is evaluated based on the

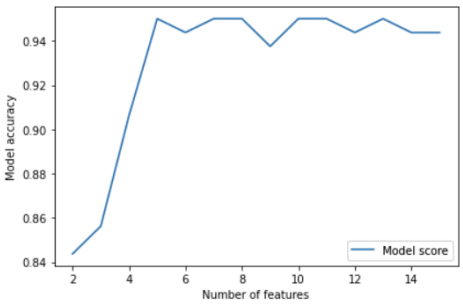

Figure (10) Performance of the configuration PCA-DT for different number of features.

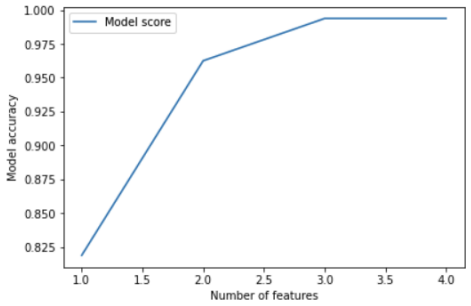

Figure (11) Performance of the configuration LDA-DT for different number of features.

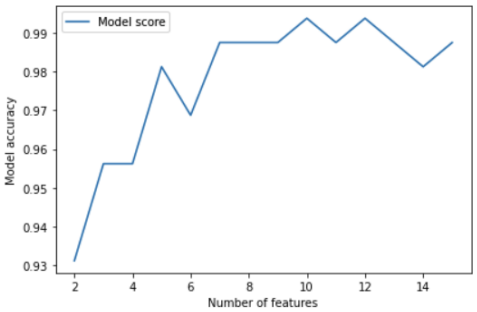

Figure (12) Performance of the configuration PCA-RF for different number of features.

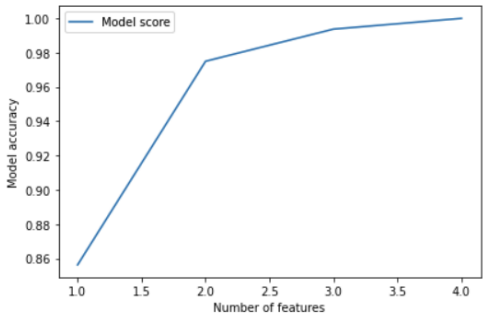

Figure (13) Performance of the configuration LDA-RF for different number of features.

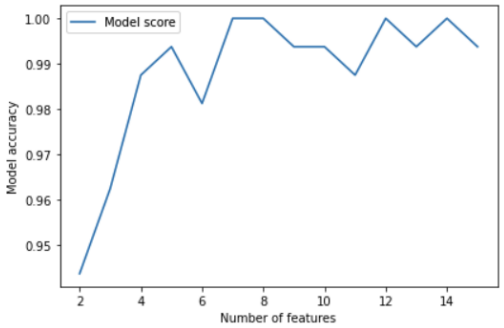

Figure (14) Performance of the configuration PCA-KNN for different number of features.

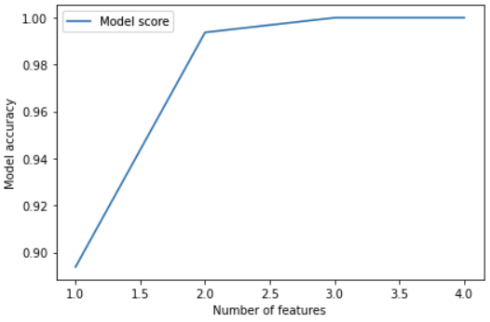

Figure (15) Performance of the configuration LDA-KNN for different number of features.

distance to his own cluster and the cluster closest to this one. Clustering the dataset in six clusters leads to more outliers (points closer to clusters belonging to another class) and more clusters. For all this, the number of clusters is set to five.

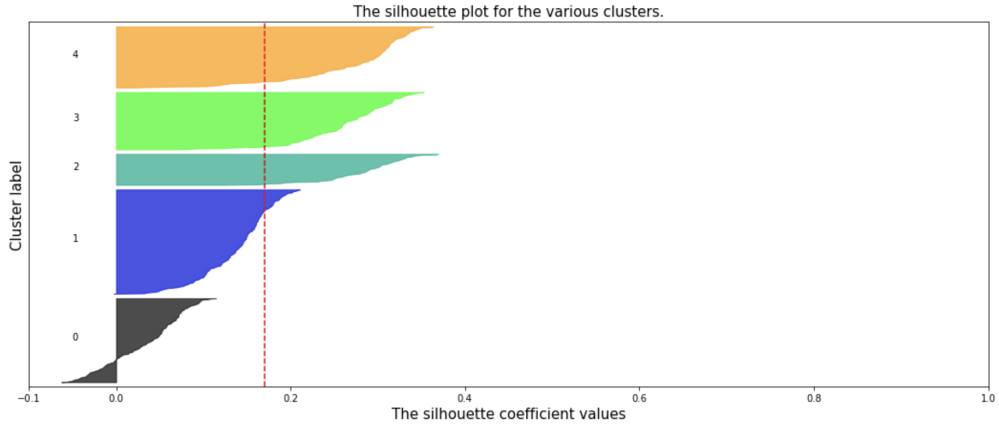

Figure (16) Silhouette scores for points of the 5 clusters of the original dataset.

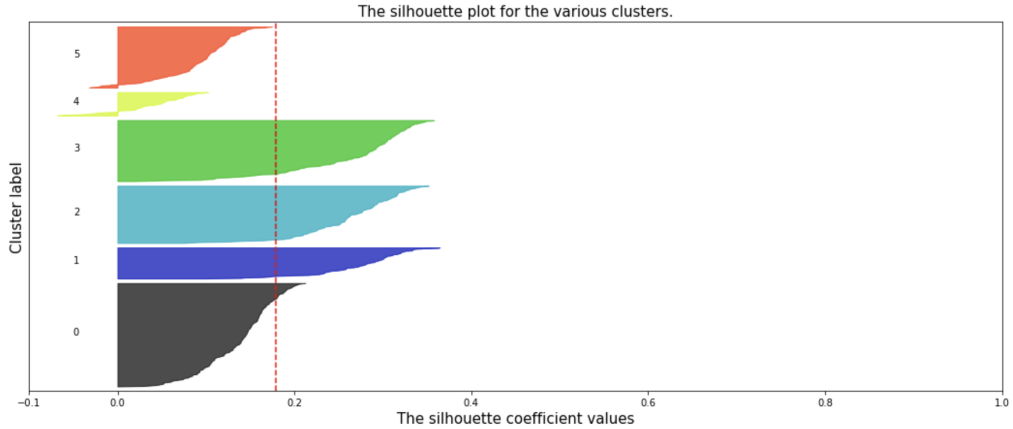

Figure (17) Silhouette scores of each cluster of the 6 clusters of the original dataset.

Clustering the best-reduced data of the selected features show us higher scores. The elbow method indicates an optimal number of clusters of five, while the Silhouette algorithm presents higher values of clustering in contrast with the not reduced dataset. The highest Silhouette scores are obtained with five clusters (0.83 average Silhouette score) and six (0.76 average Silhouette score). Furthermore, plotting individual scores shows us that with five clusters there are (almost) no outliers (18 and 19).

Figure (18) Silhouette scores for points of the 5 clusters of the best-reduced data.

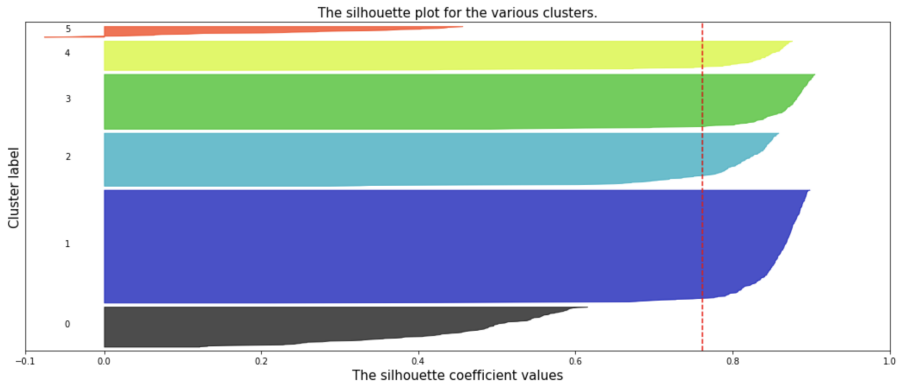

Figure (19) Silhouette scores of each cluster of the 6 clusters of the best-reduced data.

4 Discussions and/or conclusions

In order to structure this discussion we will first interpret the results of the image pipeline. Then we will interpret the results for the numerical pipeline.

4.1 Discussion: Image data pipeline

As we noticed during our first observations of the image data, the Leopard, Cheetah and Jaguar really look alike. They contain similar features (such as the spots) and we therefore sometimes struggled to differentiate them ourselves. As mentioned in the Result section, the similarities are confirmed by the histogram of clustered SIFT key points, as shown in Figure 7. This can also be noticed by the mistakes in the confusion matrix in Figure 5.

Creating a well performing classifier using the BigCats dataset is difficult due to several factors. As mentioned in the Methods section, we removed several duplicates. After this selection, we were only left with 160 images, which is very few for a multi-class image classification problem. Furthermore, the animals in the images are in many different poses and often only partially observable. Hence, important body parts of the animal, e.g. the head, are often not (clearly) visible. These

parts are often key to differentiating them. Hence, using more and clearer images will definitely improve the performance of the classifiers.

Since the SIFT algorithm is invariant to scaling and rotation we did not do any data augmentation. However, we were curious whether flipping the image had a positive effect. To our best knowledge, the PR literature did not try this in combination with SIFT. Flipping an image means inverting the x-axis positive values by negative values and vice versa. As previously mentioned in the Results section, the accuracy did not improve using data augmentation. Therefore, we conclude that flipping does not improve the accuracy when using SIFT.

Within the Image pipeline we use SIFT. However, many more feature extraction algorithms exist. SIFT only extracts shape features. One can also extract color features or texture features [22]. A combination of these type of features might also improve the accuracy. For the *BigCats* dataset we leave this for future research.

In the Results section we present a histogram of the average clustered SIFT key points of all the images per class in Figure 7. From this histogram we can infer several conclusions about the data. Some clusters are important for the classification of several animals. We can see that in cluster 16 the frequency of key points belonging to a Cheetah is dominant. This means that the center of this cluster represents a key point that is characteristic for a Cheetah. However, we also see that the Jaguar and Cheetah frequency of cluster 16 is dominant in comparison to the Lion and the Tiger. Cluster 16 apparently represents a key points that is not only characteristic for a Cheetah, but also for a Leopard and a Jaguar. This key point might represent a shared feature, such as the patterns on the fur which look similar. More of these key points can be seen in Figure 7. It would be interesting to visualize such key points in order to explore these key points and analyze them. We leave this for future research.

Currently we use SIFT on the entire image including the background which is irrelevant for classification. In order to prevent SIFT from extracting keyp oints from the background we can cut out the animal. However, some images contains multiple animals which makes this process more difficult. A possible method for automated image cropping for animals is presented in [6]. While this method was tested on giraffes, it could be interesting to evaluate the performance on the *BigCats* dataset.

An ensemble classifier is able to enhance performance by combining the predictions of different models. Adding more classifiers to the ensemble will likely increase this effect. Another possible improvement might be the voting system that we currently use for our ensemble classifier. Currently we use the plurality rule for voting and if we obtain a tie we use the model with the the highest accuracy. Another method would be predicting probabilities of classes, instead of classes themselves. The ensemble would then choose the class with the highest summed probability.

4.2 Discussion: Numerical data pipeline

First by having a glance at the dataset, we can see a large number of features are used to describe an RNA-sequence. While exploring the data, we observed that some features (267) have zero contribution (to the explanation of the variance) meaning that the feature was the 20530-*dimensional* zero vector. Also the number of examples is very small compared to the number of features. This is the so called *curse of dimensionality*[3]. In higher dimensions the data becomes very sparse and tend to move away from the center. The sparsity of data points prevents data organization techniques which are required to extract information about them.

Before applying any preprocessing, we tested a Decision Tree accuracy on the data. The model showed a high accuracy, given the nature of the dataset. This made us think that the data, regardless their high-dimensionality, should be quite well clustered in space. This became very clear when we

applied the visualization algorithm. The data appeared to form almost unique clusters without any "noisy" examples. This made us conclude that the expected results with the selection of a suitable model, will lead to a nearly perfect classifier. This classifier would be optimal in general, if all new genomes have the same structure in the future. Data points which fall between clusters or in the wrong cluster will have a high change of misclassification.

Despite the peculiarity of this dataset, PCA and LDA are not performing optimally with this high-dimensional dataset. Due to the sparsity of data, which we described above, it is difficult to find the principal component with the higher variance. As for the LDA, this is known as the small sample problem (SSS)[15]. This problem states that the eigenvectors and eigenvalues become impossible to calculate if the dimensionality d is very large compared to the number of training examples N. Is is due to the fact that the within-class scatter matrix becomes singular. There are multiple ways to tackle the relatively small dataset compared to its dimensions. One way is with the implementation of the Direct LDA algorithm[25], Regularized LDA algorithm[14] and many more. Based on our analysis one proposed way to approach this problem, is the PCA+LDA technique[2]. The first step is to apply PCA. This will lead to a significant amount of lower dimensionality. Then we apply LDA on the dimension-reduced dataset, dealing with the singularity of the scatter matrix. On contrast, this technique has drawbacks, one that stands out is the loss of information when applying the PCA.

Our results for the numerical pipeline had the outcome that we expected. Because of the high and distinct concentration of the classes all of our models achieved nearly perfect scores, even when testing for dataset which was reduced in dimensions by PCA. We kept only the 70% of the information and still the accuracy was extremely high. Also the Random Forest method, which is an ensemble method of Decision Trees, had a better result from a single Decision Tree. This is also the reason behind our choice of models, to see the differences between these two.

Finally, every method that we applied to optimize the results (K-fold cross validation) resulted in better performances from the model but not to noticeable point. The advantage of applying K-fold, is to gain an insight about how your model will perform, for small changes in the training dataset.

4.3 General Discussion

As mentioned in the Introduction, the goal of this project is to create two pipelines, using traditional machine learning methods, where one is based on a dataset containing images, and one is based on a numerical dataset. One of the main differences during the approach of the two pipelines, is that with the image pipeline, we cannot directly feed this to a classifier (and expect good results), where with a numerical dataset this might be possible. From the image dataset we first need to extract features in order to extract relevant features from the image. While it is also true that for a numerical dataset, not all data is relevant, the transformations numerical datasets require is less complex. In order to prevent creating opaque pipelines like neural networks, a lot more data analysis is required for both pipelines. When using a neural network, one can feed a network with the image or numerical data directly (in most cases). However, when using traditional machine learning techniques, one needs to explore the dimensionality, feature extraction methods and several types of classifiers in order to achieve an acceptable accuracy while a state-of-the-art convolutional neural network approach might achieve a higher accuracy with minimal data analysis. As previously mentioned a neural network is an opaque method where it is difficult to explain the reasoning behind the decisions this network makes as the network expands. However, with the techniques used in traditional machine learning techniques, we are able to explain the reasoning of our pipelines and improve on them. Even though these methods might not reach the state-of-the art neural network results, using these methods could be beneficial to overcome the problem of seeing classifiers as black-boxes and might be a trade-off the scientific community needs to consider.

Individual contributions

1. Image Pipeline:
   (a) Code: Jeroen & Rik (for specific contributions see GIT)
   (b) Report Methods + Results + Discussion: Jeroen & Rik
2. Numerical Pipeline:
   (a) Code: Pedro & Sarandis
   (b) Report Methods + Results + Discussion: Pedro & Sarandis
3. Report; General Discussion + Introduction: Jeroen & Rik
4. Task 2: Semi-supervised learning
   (a) Code: Sarandis & Pedro
   (b) Report: Jeroen & Rik

References

[1] Yang Bai, Lihua Guo, Lianwen Jin, and Qinghua Huang. A novel feature extraction method using pyramid histogram of orientation gradients for smile recognition. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 3305–3308. IEEE, 2009.

[2] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.

[3] Richard Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

[4] Devyani Bhamare and Poonam Suryawanshi. Review on reliable pattern recognition with machine learning techniques. *Fuzzy Information and Engineering*, 10(3):362–377, 2018.

[5] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[6] Patrick Buehler, Bill Carroll, Ashish Bhatia, Vivek Gupta, and Derek E Lee. An automated program to find animals and crop photographs for individual recognition. *Ecological informatics*, 50:191–196, 2019.

[7] Davide Castelvecchi. Can we open the black box of ai? *Nature News*, 538(7623):20, 2016.

[8] Roberto Cervelló-Royo, Francisco Guijarro, and Karolina Michniuk. Stock market trading rule based on pattern recognition and technical analysis: Forecasting the djia index with intraday data. *Expert systems with Applications*, 42(14):5963–5975, 2015.

[9] Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3642–3649. IEEE, 2012.

[10] Dan Claudiu Ciresan, Ueli Meier, Jonathan Masci, Luca Maria Gambardella, and Jürgen Schmidhuber. Flexible, high performance convolutional neural networks for image classification. In *Twenty-second international joint conference on artificial intelligence*, 2011.

[11] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[12] 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.

[13] Evelyn Fix and Joseph L Hodges Jr. Discriminatory analysis-nonparametric discrimination: Small sample performance. Technical report, California Univ Berkeley, 1952.

[14] Jerome H. Friedman. Regularized discriminant analysis. *J. Amer. Statist. Assoc*, pages 165–175, 1989.

[15] Kohji Fukunaga. Introduction to statistical pattern recognition-second edition. 1990.

[16] Geoffrey Hinton and Sam T Roweis. Stochastic neighbor embedding. In *NIPS*, volume 15, pages 833–840. Citeseer, 2002.

[17] Jianming Li, Shuguang Huang, Rongsheng He, and Kunming Qian. Image classification based on fuzzy support vector machine. In *2008 International Symposium on Computational Intelligence and Design*, volume 1, pages 68–71. IEEE, 2008.

[18] Ming Li and Zhi-Hua Zhou. Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 37(6):1088–1098, 2007.

[19] David G Lowe. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1150–1157. Ieee, 1999.

[20] Leland McInnes, John Healy, and James Melville. Umap: uniform manifold approximation and projection for dimension reduction. 2020.

[21] John Ashworth Nelder and Robert WM Wedderburn. Generalized linear models. *Journal of the Royal Statistical Society: Series A (General)*, 135(3):370–384, 1972.

[22] Dong ping Tian et al. A review on image feature extraction and representation techniques. *International Journal of Multimedia and Ubiquitous Engineering*, 8(4):385–396, 2013.

[23] AWD Udaya Shalika and Lasantha Seneviratne. Animal classification system based on image processing & support vector machine. *Journal of Computer and Communications*, 4(1):12–21, 2016.

[24] Qing Song, Wenjie Hu, and Wenfang Xie. Robust support vector machine with bullet hole image classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 32(4):440–448, 2002.

[25] Hua Yu and Jie Yang. A direct lda algorithm for high-dimensional data - with application to face recognition. *Pattern Recognit.*, 34:2067–2070, 2001.

[26] Yudong Zhang and Lenan Wu. Classification of fruits using computer vision and a multiclass support vector machine. *sensors*, 12(9):12489–12505, 2012.