PLANKTON IMAGE CLUSTERING USING SIMILARITY METRIC LEARNING

Lappeenranta-Lahti University of Technology LUT

Computational Engineering, Bachelor's Thesis

2022

Joona Ylijoki

Examiner: Tuomas Eerola

ABSTRACT

Lappeenranta-Lahti University of Technology LUT
School of Engineering Science
Computational Engineering

Joona Ylijoki

Plankton image clustering using similarity metric learning

Bachelor's thesis
2022
29 pages, 16 figures, 2 tables
Examiner: Tuomas Eerola

Keywords: plankton recognition; clustering; computer vision; pattern recognition

Technological advancement has evolved the imaging equipment used in plankton imaging. Nowadays it is possible to take images more efficiently. Due to the increased amount of images the processing time is longer. The recognition process has been automatized, for example, with neural networks but the requirement to train the new species into the model is slowing the process. One of the proposed solutions for this problem is open-set plankton recognition which utilizes similarity metric learning and embedding vectors to recognize the species in the image. The embedding vectors for the same species are closer to each other than the embedding vectors of two different species. Open-set classification is enabled by a set threshold value which indicates the probability for the images to belong to one of the existing classes.

Clustering is grouping of the data based on the similarities of the data points. It can be utilized to analyze data sets without previous knowledge about the labels in the data. In this thesis K-Medoids clustering method was used to group the embedding vectors of the plankton species based on their cosine distances. The clustering was visualized by using smaller subsets that contained embedding vectors from only three classes. The performance of the clustering was evaluated by calculating the purity for the clusters. The purity of the cluster indicates the percentage of the dominant class in the cluster. The clustering purity was calculated for the visualized clusters and the whole data. The data contained 37 840 embedding vectors and 50 different plankton species. The whole data was clustered with purity of slightly over 82%. The results were promising for using clustering to group plankton images.

TIIVISTELMÄ

Lappeenrannan-Lahden teknillinen yliopisto LUT
School of Engineering Science
Laskennallinen tekniikka

Joona Ylijoki

Plankton kuvien klusterointi hyödyntäen samankaltaisuus metriikkaa

Kandidaatintyö
2022
29 sivua, 16 kuvaa, 2 taulukkoa
Tarkastaja: Tuomas Eerola

Avainsanat: planktoneiden tunnistaminen; klusterointi; konenäkö; hahmontunnistus

Teknologian kehitys on uudistanut kuvantamislaitteita, joita käytetään myös planktoneiden kuvantamiseen. Kuvia saadaan nykyään otettua tehokkaammin. Kasvaneen kuvamäärän vuoksi niiden käsittelyyn kuluva aika on pidempi. Tunnistusta on automatisoitu esimerkiksi neuroverkkojen avulla, mutta uusien lajien kouluttaminen niihin hidastaa prosessia. Ratkaisuksi tähän on ehdotettu avointa planktonin tunnistusta, jossa hyödynnetään samankaltaisuus metriikkaa ja piirrevektoreita kuvassa olevan lajin tunnistamiseksi. Saman lajin edustajien piirrevektorit ovat lähempänä toisiaan kuin kahden eri lajin edustajien piirrevektorit. Avoin luokittelu mahdollistetaan asettamalla kynnysarvo, joka ilmaisee todennäköisyyden kuvalle kuulua johonkin olemassa olevista luokista.

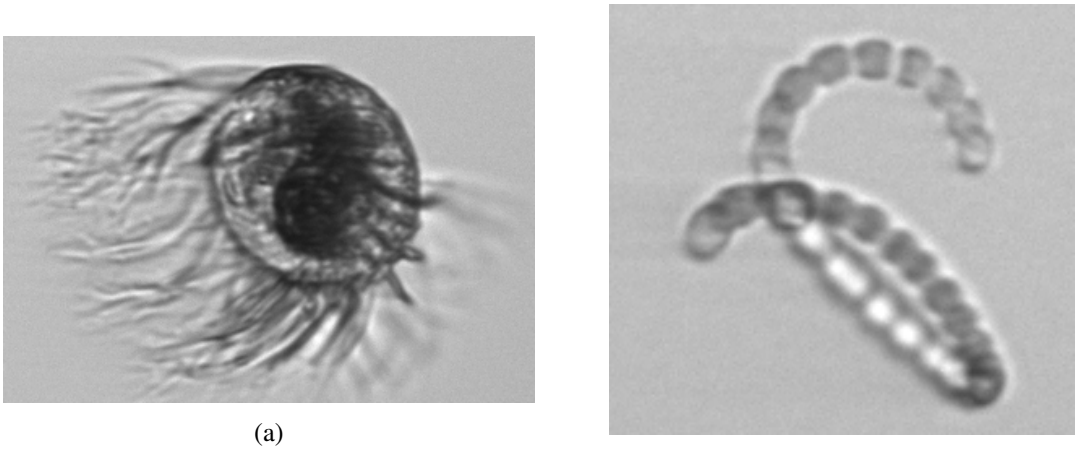
Klusterointi on datan ryhmittelyä sen sisältämien samankaltaisuuksien perusteella. Klusterointia voidaan hyödyntää datan analysoinnissa ilman tietoa datan sisältämistä nimikkeistä. Työssä käytettiin K-Medoids klusterointimenetelmää, jolla ryhmiteltiin planktonlajien piirrevektoreita niiden kosinietäisyyksien perusteella. Klusterointia visualisoitiin käyttämällä pienempiä osa-joukkoja, jotka sisälsivät vain kolmea eri luokkaa. Klusteroinnin onnistumista arvioitiin laskemalla klustereiden puhtaus. Klusterin puhtaus osoittaa klusterin dominoivan luokan suhteellisen osuuden klusterin sisältämistä kaikista luokka-arvoista. Klusteroinnin puhtaus laskettiin visualisoiduille klustereille ja kaiken tutkimuksessa käytössä olleen datan klusteroinnille. Käytetyssä datassa oli 37 840 piirrevektoria, sisältäen 50 eri luokkaa eli planktonlajia. Tulokseksi kaiken datan klusteroinnille saatiin noin 82% puhtaus. Tulokset olivat lupaavia klusteroinnin käyttämiseksi planktoneiden ryhmittelyssä.

CONTENTS

1 INTRODUCTION

1.1 Background

Plankton are small organisms living in the oceans and lakes all over the world. Plankton are divided into their own groups by size or the type of the plankton. Most common way of categorizing plankton is to divide them based on their main types. These types are phytoplankton which include plant types plankton and zooplankton which include animal types plankton [1]. Example images of zooplankton and phytoplankton can be seen in Figure 1.



(a) (b)

Figure 1. Example images of plankton: (a) Ciliata plankton (zooplankton); (b) Dolichospermum anabaenopsis plankton (phytoplankton)

Especially phytoplankton have a major role in ocean ecosystems. Phytoplankton, like other plants as well, are capable of photosynthesis. This means that they can consume carbon dioxide and transform it to oxygen. The amount of carbon dioxide consumed by phytoplankton can be compared to forests which signifies that the plankton does not only have a important role in the oceans but on the whole world. Portion of this carbon sinks to the bottom of the ocean when phytoplankton, or other aquatic animals that has eaten phytoplankton, die. This is the second factor on phytoplankton's major role in the ecosystem. Other aquatic animals like zooplankton, fishes, krills and even whales use them as food. Phytoplankton are one of the first members of the ocean food chain which continues all the way to the largest aquatic creatures and also to humans through consumption of fish and other marine animals. Similarly to plants, phytoplankton growth and amount changes throughout the year. Different climates in varying geographical locations affect the size

of the populations and their survival. Because plankton are so small and drift around easily, also changing winds and currents affect the populations location. Ocean chemistry and biology models has predicted that phytoplankton productivity is declining because of human actions like increased greenhouse gases. Also the ocean surface water temperature is increasing through climate change and humans are harming the ocean ecosystems for example through the plastic waste that has been thrown there. For these reasons it is important to study more about plankton activity and the effects humans are causing to them. One of the ways is photographing plankton populations in oceans to identify what kind of plankton species lives in what areas and how does for example changing weather circumstances affect the populations [2].

This research is a part of the FASTVISION-plus project that focuses on plankton recognition. As a part of the project Mohamed et al. [3] studied similarity learning in plankton recognition. The objective for that research was to train a deep learning model for recognizing plankton through similarities in the images taken from the plankton. In this thesis the embedding vectors built by the deep learning model were clustered in order to determine the performance of the learnt similarity metric.

1.2 Objectives and delimitations

The main objective of this research was to study existing clustering methods to determine how well does the learnt similarity metric from Mohamed et al. [3] perform in unsupervised categorization of plankton. This can be divided into the following objectives:

1. Review of clustering methods
2. Research how does the clustering work for the plankton image data
3. Implement the coding for the clustering
4. Analysis of the clusters' purity

Clustering method's purity is based on how well does the method succeed to categorize the plankton images into the clusters corresponding actual classes. The embedding vectors were calculated from the plankton images by using the deep learning model that was trained in the research by Mohamed et al. [3].

This research was delimited in a way that the plankton image data only includes images of phytoplankton. The plankton images were also taken with one specific imaging equipment.

1.3 Structure of the thesis

Section 2 includes commonly used methods for plankton recognition and a recent proposal for solving some of the major problems in plankton recognition. Section 3 includes the theory for clustering in general, the different clustering methods that were reviewed in this research and most common distance measures used in clustering. Section 4 presents the proposed method to use in the clustering of plankton images. Section 5 contains introduction of the data, what tests were performed and how, and the results for the proposed clustering method. Section 6 includes the conclusion for this research.

2 PLANKTON RECOGNITION

2.1 Plankton recognition methods

In the early ages of plankton research, the recognition was done manually. Experts in the field handled all the images one by one. When the amount of images needed to be recognized grew, experts did not have time to process every image. Solution to this problem was to automize the recognition process. One of the first ways of automatically recognizing plankton was to use handmade features from images [4]. This method relied on comparing, for example, the texture and the shape of the image. Usage of machine learning has since generalized in the field and replaced some of the former methods and resources.

Progression in the technological field has created numerous advanced ways that are utilized for new ways of recognizing plankton. One of the advanced tools used to heighten the efficiency is submersible imaging flow cytometer which can be used to take images of particles in the water. Advancing technology enables the constant improvement in image resolution and the amount of images that can be taken. McLane research laboratories' The Imaging FlowCytobot can take approximately 30 000 high resolution images every hour [5]. Plankton imaging instruments are often automated or semi-automated which means that the imaging can be for example triggered automatically by certain amount of individual particles. Triggers make use of laser-induced fluorescence and light scattering. All the data will be sent from the imaging instrument to shore instantaneously after the image has been taken [6].

Increasing dataflow from the imaging equipment leads to problems when time taken to analyze all the gathered information is increasing simultaneously. Automatic methods need to be implemented to go through all the image data and to recognize the plankton species from those images. One solution is to use convolutional neural networks (CNNs) that have been proven to be able to recognize objects and categorize the images almost as well as an expert in the field [7]. Unfortunately, there is one major problem with using CNN-based classifier in image recognition. It is only effective enough if the image data is similar to the data that has been used to train the CNN. It cannot recognize new species that has not been trained to it. In order to add the new species class to the model, the CNN needs to be retrained with the new data. In plankton recognition this causes problems because the species vary geographically and seasonally [8].

2.2 Open-set plankton recognition

Mohamed et al. [3] proposed an open-set plankton recognition method. The method utilizes metric learning with image embeddings in the way that the images representing same species have embedding vectors close to each other and images representing different species have embedding vectors that are further away. Open-set classification is enabled by a set threshold value which designates the probability for the images to belong to the same class. New classes can be added to the model with no need to train them into it because the model does not need to learn any class-specific image features. Explanatory diagram for the method is presented in Figure 2.

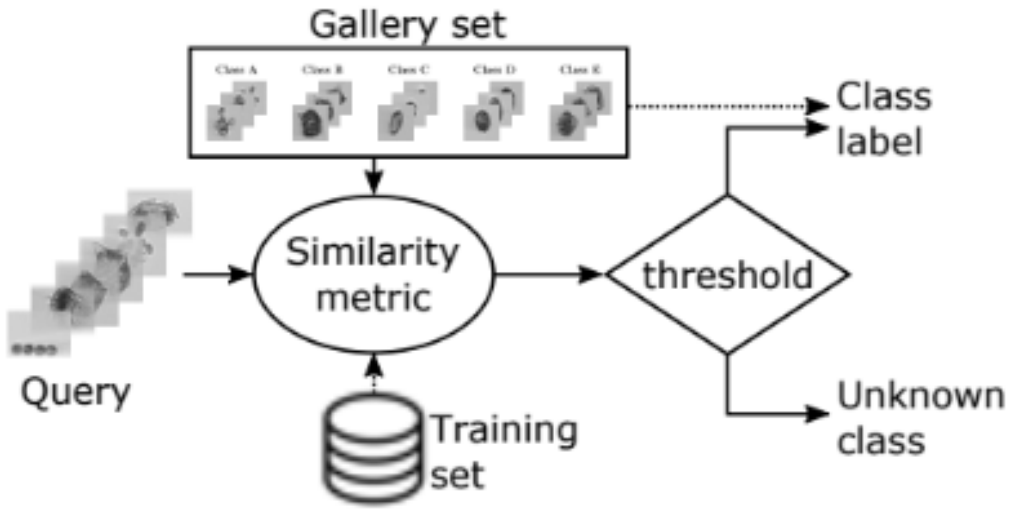


Figure 2. Open-set plankton recognition [3].

Mohamed et al. [3] advocated to use ArcFace loss [9] collaborated with Generalized mean pooling (GeM) [10] when training the similarity metric. ArcFace utilizes a similarity learning technique that enables simultaneous operation for the classification task and solving the distance metric learning. This possibility is based on the Angular Margin Loss utilized by the ArcFace.

Basic classification model tries to estimate the probability for an element to belong to a certain class based on the characteristics of that specific class. Similarity metric contrastingly compares those two elements and tries to recognize similar objects based on the characteristics of the objects.

3 CLUSTERING

Clustering is an unsupervised technique to calculate and present data with similarities or dissimilarities. The similarity is based on the data items' distance from each other. Clustering is commonly used with statistical data analysis in for example machine learning, pattern recognition and image analysis [11].

Clustering methods can be divided into hierarchical and partitional algorithms. Partitional clustering works in a way that all the clusters are determined at the same time. Hierarchical clustering algorithms in turn use previously established clusters to get subsequent clusters [11]. Illustration of the latter is presented in Figure 3.

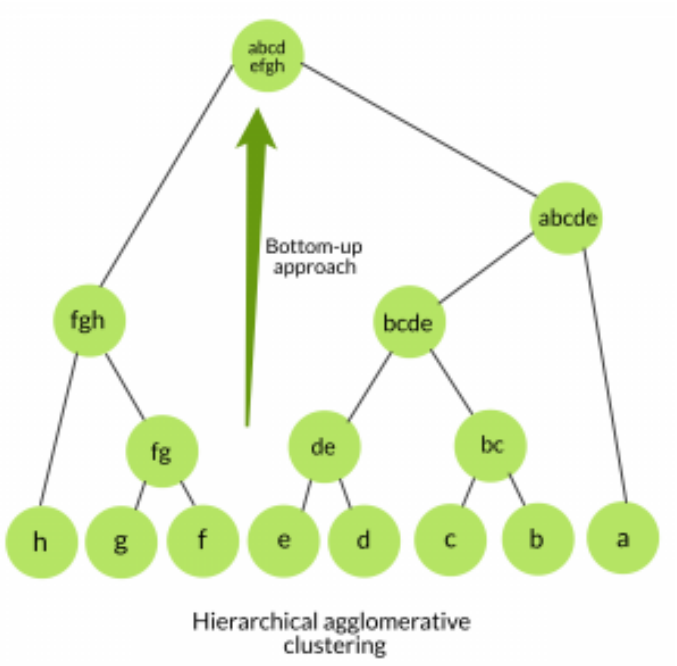


Figure 3. Hierarchical Agglomerative Clustering [12].

3.1 Distance metrics

There are numerous distance measure techniques available to use to measure distance between two points. The most straightforward distance measure is Manhattan distance which calculates the distance based on the two points' differences in for example X-axis and Y-axis values. It can be thought like moving in a chessboard where one square equals one distance unit. The definition for Manhattan distance is the sum of the absolute values

of the differences between corresponding components and it can be calculated using the formula:

$$d = \sum_{i=1}^{n} jX_i \quad Y_i j, \tag{1}$$

where n is the amount of data points and X_i and Y_i are the values at points X and Y for the particular data point.

One of the generally used distance measure techniques is Euclidean distance. Euclidean distance calculates the shortest distance between two points. Despite of the generality of Euclidean distance there is some major disadvantages. For example it is not scale invariant which designates that the calculated distances can be skewed. In order to prevent this kind of error from occurring the data must be normalized before calculating the distances. Additionally, as the dimensions in the data multiplies the more impractical Euclidean distance becomes. The definition for Euclidean distance is the square root of the sum of the squares of the differences between corresponding values. Euclidean distance can be calculated using the formula:

$$d = \sqrt{\sum_{j=1}^{m} (X_j \quad Y_j)^2}, \tag{2}$$

where m is the amount of data points and X_j and Y_j are the values at points X and Y for the particular data point.

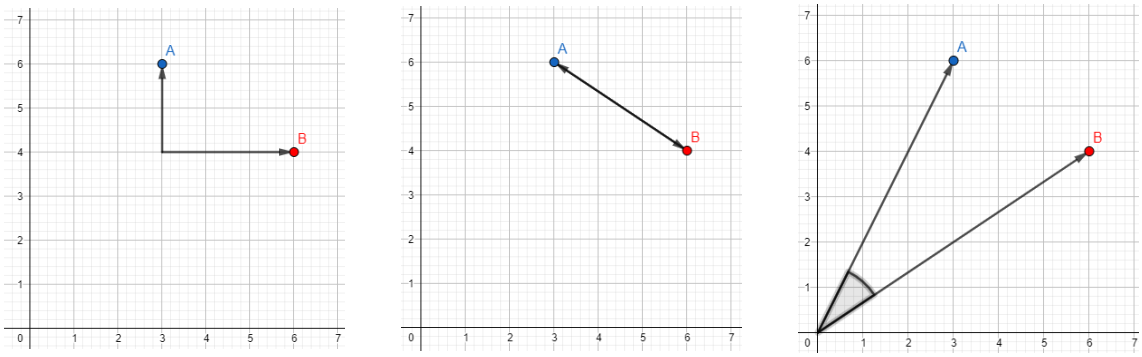
Cosine distance measure is superior to Euclidean distance when it comes to multidimensional data. The cosine similarity is the cosine of the angle between two vectors that point to the data points. Cosine similarity can be defined as the dot product of the vectors divided by the product of those vectors' lengths. Cosine similarity can be calculated using the formula:

$$CosineSimilarity(x, y) = \cos(\theta) = \frac{x \quad y}{jjxjj \; jjyjj}, \tag{3}$$

where x and y are the vectors. Cosine distance can then be calculated using the following formula:

$$CosineDistance = 1 \quad CosineSimilarity \tag{4}$$

Visualization of the distance measures is presented in Figure 4.



(a) (b) (c)

Figure 4. Different distance measures: (a) Manhattan distance; (b) Euclidean distance; (c) Cosine similarity

3.2 K-Means

Common example of partitional clustering is K-Means clustering [13]. In K-Means clustering the amount of used clusters is decided beforehand. Number of clusters will be designated in the variable K. The algorithm then chooses K amount of random points called centroids. After that each data point's distance to each of the centroids is calculated and every data point is assigned to one of the clusters defined by the centroids. Most commonly used distance measure method is Euclidean distance. Then each centroid is moved to the centre of the data points which are assigned to that centroid. This is done by calculating the mean of those data points. After that the data points' distance to each centroid is calculated again and the data points are assigned to the nearest centroid. These two steps will repeat until none of the centroids need relocating and the clustering is thus done. The steps for K-Means clustering is visualized in Figure 5.

The biggest drawback for K-Means clustering is its sensibility to outliers. Outliers are data points far away from other data points and therefore they will disrupt when calculating the mean and change the positioning of the centroids. K-Means operates well with spherical data and poorly with data that has arbitrary shape [15].

3.3 Hierarchical clustering

Hierarchical clustering does not require to specify the amount of clusters. Hierarchical clustering can be further divided into agglomerative and divisive types. In agglomerative hierarchical clustering the clusters are formed bottom-up [16]. This means that in the beginning all of the elements are in their own clusters, which are called leafs in hierarchical

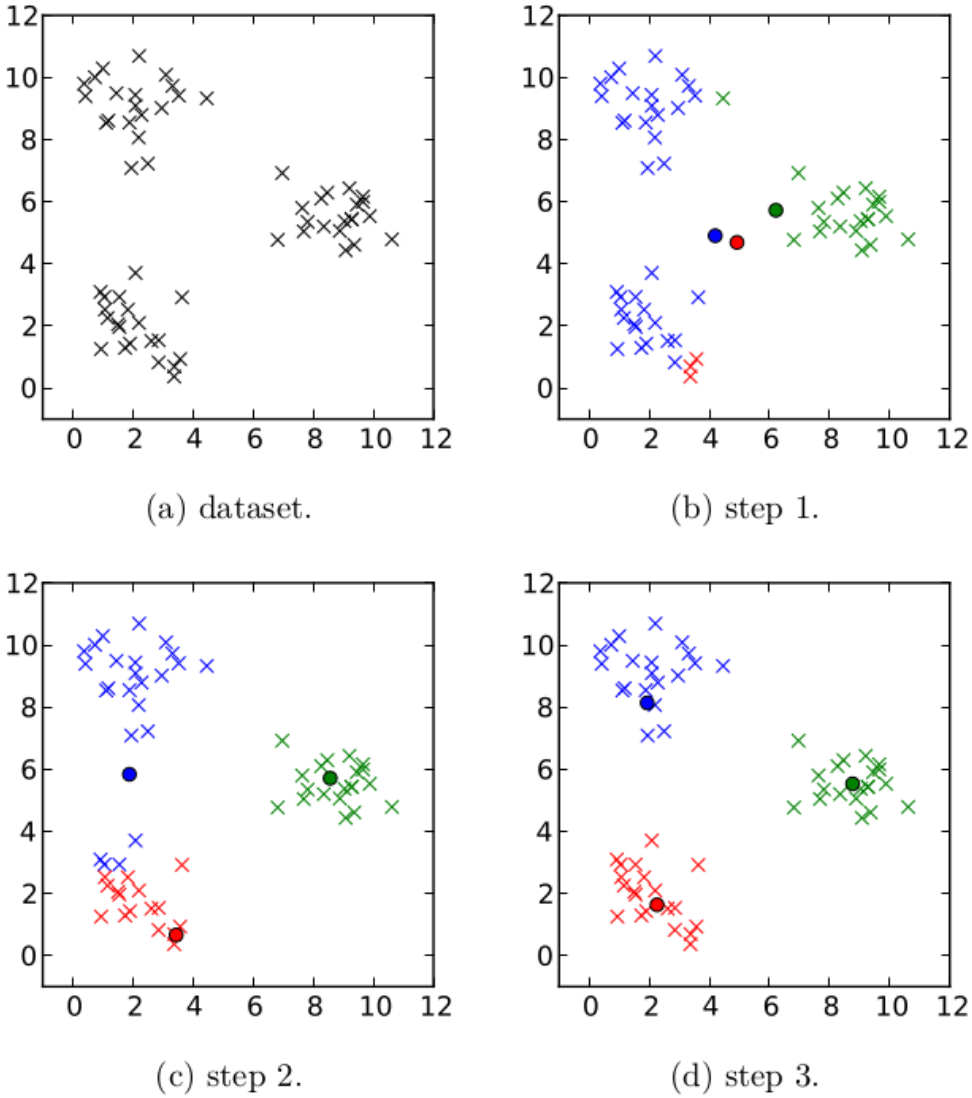


Figure 5. K-Means clustering algorithm steps visualized. [14]

clustering, and they will be combined with each other to create larger clusters. After the start, each cluster calculates the distance to every other cluster and the two clusters with the shortest distance to each other will combine into one cluster which is called node. One of the most commonly used distance measure is Euclidean distance. This process of calculating the two clusters with the shortest distance off of each other is repeated until we get to the point where two last nodes combine and form a single cluster which is called the root. The end result reminds a tree that has all the elements as leafs and the final cluster as a root. The tree shaped structure is called a dendrogram [17]. An example of dendrogram is presented in Figure 6.

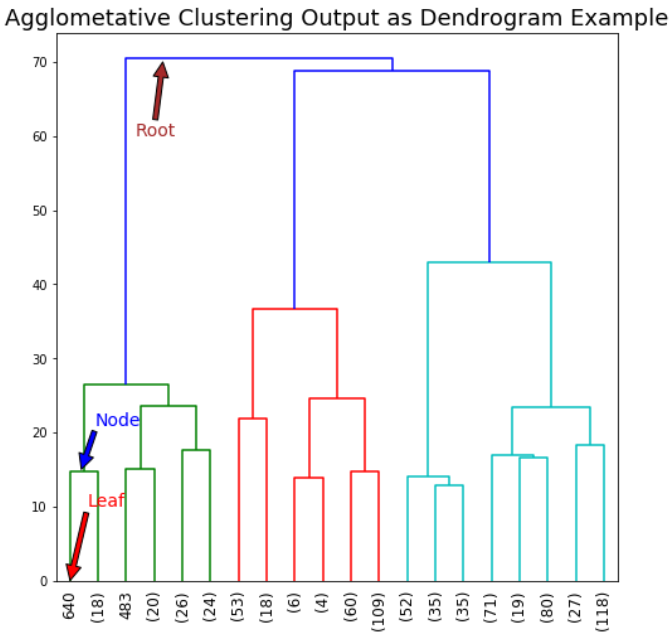


Figure 6. Hierarchical Agglomerative Clustering in Dendrogram form [18].

Agglomerative hierarchical clustering does not give any specific answer to how many clusters should the data be divided into. Oneself can decide where to cut-off the tree and that is what determines how many clusters there will be. The height of the bar is the distance from the leaf to the node. The taller the bar is, the further the two combined clusters are from each other.

Divisive hierarchical clustering is the opposite of agglomerative hierarchical clustering. In divisive clustering the elements are all in the same cluster in the beginning and are then divided to get smaller clusters.

3.4 K-Medoids

K-Medoids clustering is another example of partitional clustering. It is related to K-Means clustering but has a slightly different approach in some of the algorithm phases. While K-Means is based on minimizing the squared error, K-Medoids is minimizing the sum of the absolute error. One of the most prominent differences to K-Means is the cluster centroid selection. While in K-Means the cluster centroid can be any random point in the feature space, in K-Medoids the centroid is always one of the data points. Figure 7 illustrates the different cluster centers of K-Means and K-Medoids clustering. Because of this, K-Medoids is more resilient against outliers and therefore performs better than K-Means with clustering data with outliers in it. Similarly to K-Means clustering, the number of clusters in K-Medoids clustering is decided beforehand.

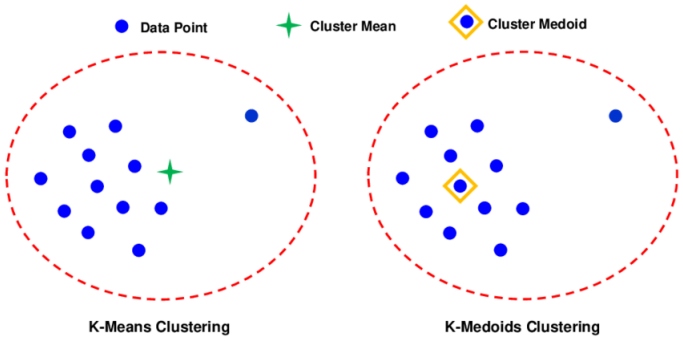


Figure 7. Cluster mean compared to cluster medoid [19]

Most frequent implementation of K-Medoids clustering is Partitioning Around Medoids (PAM) [20]. The algorithm for PAM simulates the next steps: First the medoids are selected randomly from the existing data points. The amount of medoids is decided beforehand as variable K. Then every data point is assigned to the cluster with the nearest medoid. Next for every data point, calculate the total cost for the medoid. Then reselect the medoid and calculate the new total cost for the data points. If the new cost is smaller than the cost for the previous medoid, replace the medoid to the new data point. The last three steps are repeated until the medoids do not change anymore. Finally the structure that has the lowest cost is found.

3.5 Clustering performance

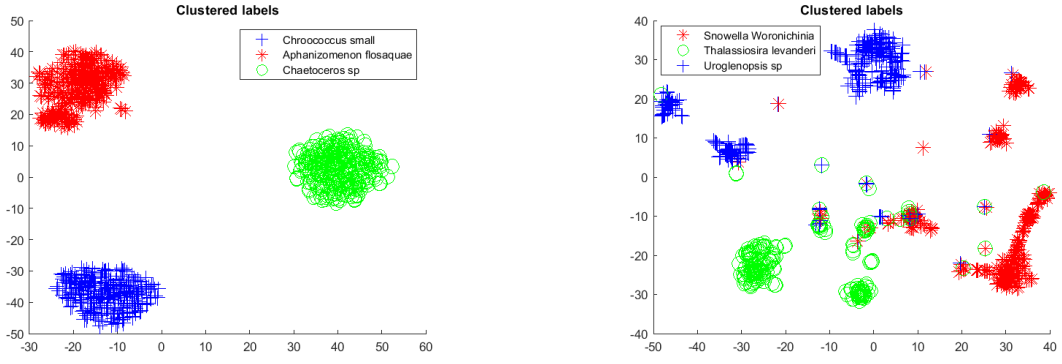
The clustering itself is followed by the evaluation on how well the clustering has succeeded. The evaluation metrics on clustering performance can be divided into two major types [21]. These are called intrinsic and extrinsic measures. Intrinsic measures are used when there is no comprehension on the data. Intrinsic measures form the performance evaluation based on cluster features like how tightly the data points are located in a single cluster or how isolated clusters are from each other [22]. Extrinsic measures contrastingly to intrinsic measures utilizes the ground truth information about the data. In extrinsic measures, the knowledge of classes and labels can be used to evaluate the purity of the cluster.

The purity can be utilized to evaluate the goodness of the clustering if the number of classes is known. Purity for each cluster can be calculated by solving each cluster's dominant class label and divide it with the total number of elements in the cluster. These cluster purities can be then added up and divided with the number of cluster to form a purity for the whole clustering. Optimal end result for clustering is that each cluster contains only one label and all of the label's elements are in that one cluster. In that case the purity for the clustering is 1. Purity can be defined as follows:

$$purity(\ \ ,\mathbb{C}) = \frac{1}{N}\sum_{k}\max_{j} j\omega_k \setminus c_j j, \tag{5}$$

where $= f\omega_1, \omega_2, ..., \omega_k g$ is the set of clusters and $\mathbb{C} = f c_1, c_2, ..., c_j g$ is the set of classes [23].

Visualization for different cluster purities is demonstrated in Figure 8.



(a) (b)

Figure 8. Example figure of clustering purities: (a) Good purity; (b) Bad purity

4 PROPOSED METHOD

The proposed method for plankton image clustering consists of acquiring the embedding vectors, clustering them, visualizing the clustering process and evaluating the performance of the clustering. The embedding vectors are calculated by using the deep learning model trained by Mohamed et al. [3]. Given the plankton image as an input to the model it outputs an embedding vector build from that image. These embedding vectors are the data for which the clustering is performed for.

In this research the proposed method for clustering the plankton images is K-Medoids clustering. Main reason for the usage of K-Medoids is the type of the data. The data containing the embedding vectors has distances in cosine format. While K-Means clustering would have been slightly simpler and lighter clustering method, the way of calculating the distances and centroids with means does not work competently in this research. Also the multidimensionality of the data can lead into problems with K-Means clustering that does not work efficiently with high dimensional data.

For the visualization to be more clear, the multidimensional data is modified to two dimensions. This is done by using t-Distributed Stochastic Neighbour Embedding method which performs well for smaller subsets of data [24]. First the pairwise distances are calculated for the high-dimensional points. Then for each high-dimensional point a standard deviation is created to predetermine the perplexity of the points. Next the similarity matrix is calculated and preliminary set of low-dimensional points is formed. Last the low-dimensional points are iteratively updated by minimizing the Kullback-Leibler divergence between a Gaussian distribution with high dimensions and t-distribution with low dimensions [24]. The proposition from acquiring the embedding vectors to calculating the clustering performance is illustrated in Figure 9.

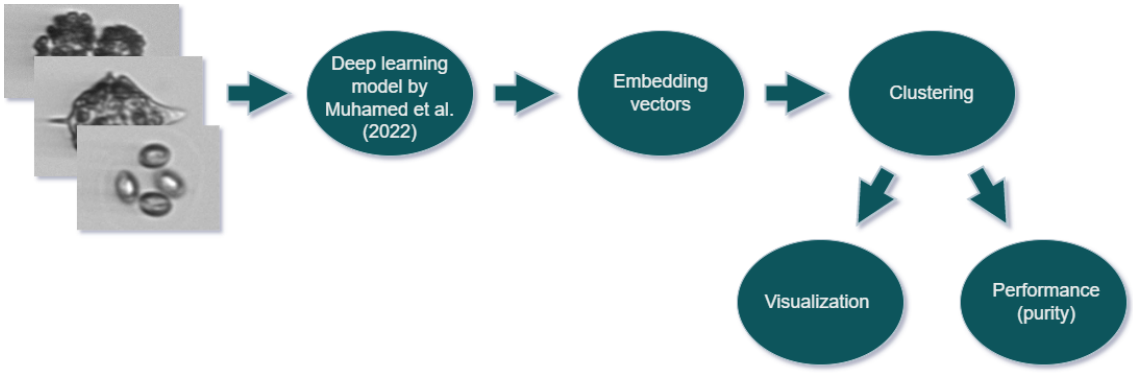
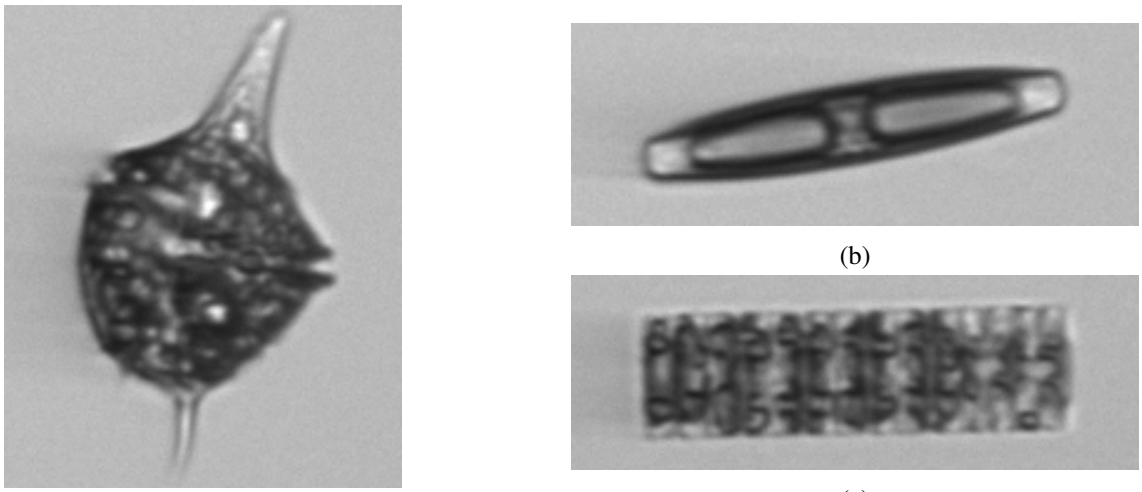


Figure 9. Pipeline figure of the proposed method

5 EXPERIMENTS

5.1 Data

The images of plankton [25] were taken from the Baltic Sea using an Imaging FlowCytobot presented in Section 2.1. Examples of plankton can be seen in Figure 10. The embedding vector data was gathered from those plankton images by running them through the deep learning model proposed by Mohamed et al. [3]. The embedding vector data contains 37 840 embedding vectors each with a length of 512. Each embedding vector corresponds a single plankton image from the set containing 50 different plankton classes. The embedding vector data contains also a label for plankton species. Labels corresponding the plankton species are listed in Table 1. Fluctuation in regularity among plankton species results in imbalance class sizes. Number of embedding vectors per label varies from 11 to 7368.



(a) (b) (c)

Figure 10. Plankton images: (a) Amylax triacantha plankton; (b) Pennales sp thick plankton; (c) Pauliella taeniata plankton

5.2 Experimental arrangements

To evaluate how successful the proposed clustering method was for dividing the embedding vectors to clusters matching the correct labels few tests were executed. At first the clustering results were visualized to determine how well does the sorting of different labels succeed. This was done by using only a small portion of the data. The reason for

Table 1. Plankton species corresponding the label numbers

Label	Species name	Label	Species name
0	Amylax triacantha	25	Heterocapsa rotundata
1	Aphanizomenon flosaquae	26	Heterocapsa triquetra
2	Aphanothece paralleliformis	27	Heterocyte
3	Beads	28	Katablepharis remigera
4	Centrales sp	29	Licmophora sp
5	Ceratoneis closterium	30	Melosira arctica
6	Chaetoceros sp	31	Merismopedia sp
7	Chaetoceros sp single	32	Mesodinium rubrum
8	Chlorococcales	33	Monoraphidium contortum
9	Chroococcales	34	Nitzschia paleacea
10	Chroococcus small	35	Nodularia spumigena
11	Ciliata	36	Oocystis sp
12	Cryptomonadales	37	Oscillatoriales
13	Cryptophyceae Teleaulax	38	Pauliella taeniata
14	Cyclotella choctawhatcheeana	39	Pennales sp thick
15	Cymbomonas tetramitiformis	40	Pennales sp thin
16	Dinophyceae	41	Peridiniella catenata chain
17	Dinophysis acuminata	42	Peridiniella catenata single
18	Dolichospermum Anabaenopsis	43	Prorocentrum cordatum
19	Dolichospermum Anabaenopsis coiled	44	Pseudopedinella sp
20	Euglenophyceae	45	Pyramimonas sp
21	Eutreptiella sp	46	Skeletonema marinoi
22	Gonyaulax verior	47	Snowella Woronichinia
23	Gymnodiniales	48	Thalassiosira levanderi
24	Gymnodinium like	49	Uroglenopsis sp

using only few classes and a fraction of their data rows was to keep the performance in a reasonable level. Visualizing all the data available would have taken unpredictable amount of time. On top of that the benefit for visualizing all the data would not have been great when the point is to make the visualization clear for the human eye. It would have only shown a obscure bundle of data points and it would have been hard to distinguish the clusters from each other. The visualization was executed with 100 or 200 embedding vectors from three different classes.

The actual evaluation of how well does the clusters correspond to the actual labels was done by calculating the purity of the clustering as described in Section 3.5.

5.3 Results

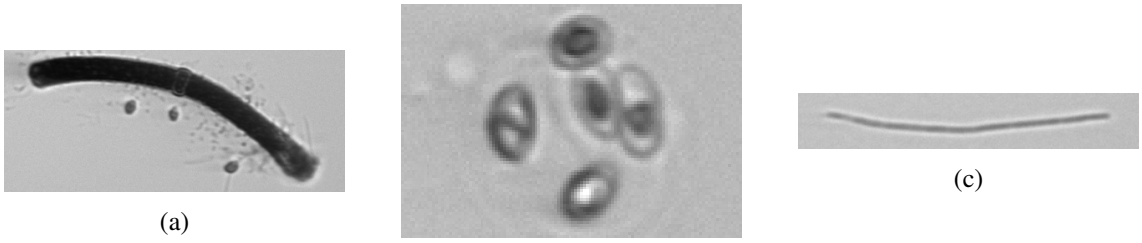
Results for cluster purities for different labels and varying sample sizes are shown in Table 2. The table shows which labels are included in the clustering and what is the size of each label set included in the test. For each formed cluster, the purities are listed in respective order to labels. The overall purity for all the clusters in each test is displayed in the last column.

The final row of the table displays the purity for the entire data including all the 50 labels. In this row the size of each label set and the purity for each cluster is defined by indicating the minimum and maximum values.

Table 2. K-Medoids clustering purities with different labels

Labels	Size of each label set	Cluster purities (respectively)	Overall purity
1, 13, 18	100	1, 0.9901, 1	0.9967
1, 13, 18	200	0.9950, 0.9901, 1	0.9950
1, 26, 49	200	0.99, 0.7663, 1	0.9188
6, 46, 48	100	0.4926, 0.8837, 0.9815	0.7859
6, 46, 48	200	0.4866, 0.9367, 0.9909	0.8047
7, 25, 41	100	0.9259, 0.9901, 1	0.9720
16, 21, 42	100	0.8448, 1, 0.9762	0.9403
21, 27, 33	100	1, 1, 1	1
21, 32, 36	200	1, 1, 1	1
27, 39, 46	100	1, 0.9901, 1	0.9967
35, 36, 37	100	1, 1, 1	1
42, 43, 49	100	0.9180, 0.5, 0.7634	0.7272
47, 48, 49	100	0.8421, 0.6892, 0.6429	0.7247
All labels (0–49)	11–7368	0.2871–1	0.8205

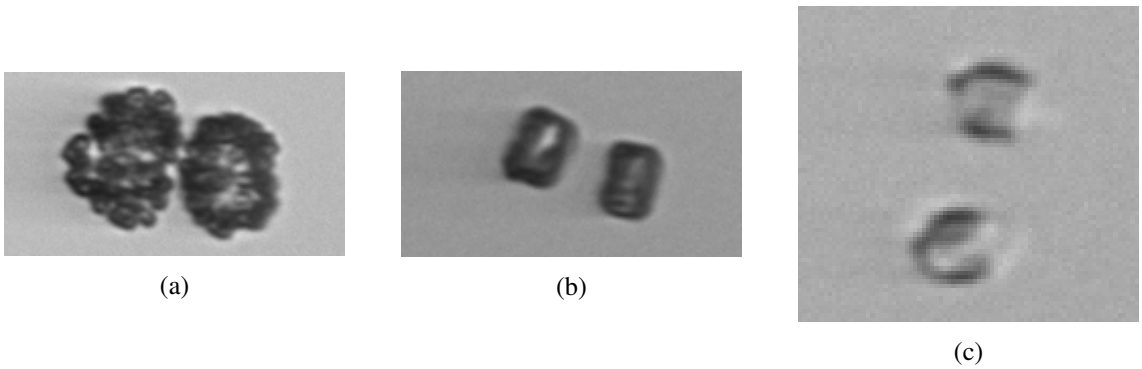
Purity for the clustering varies between the labels included in the clustering. Fluctuation of purity between individual test set sizes is not as considerable assuming the size difference is not immense. For instance for Nodularia spumigena, Oocystis sp and Oscillatoriales plankton species (labels 35, 36 and 37) the tests displayed a perfect purity of 1. So all the clusters for that set only included labels from one of those species. This would imply that the plankton species visually distinct from each other. The visual dissimilarity can be seen in Figure 11. Set containing species Aphanizomenon flosaquae, Cryptophyceae Teleaulax and Dolichospermum Anabaenopsis (labels 1, 13 and 18) displayed almost perfect purity of 0.9967 or 0.9950 subject to the size of the test set. Most of the test sets displayed a purity greater than 0.8 and even the evaluation for the entire data set displayed overall purity of 0.8205.



(a) (b) (c)

Figure 11. Visualization of plankton species with superior purity: (a) Nodularia spumigena (label 35); (b) Oocystis sp (label 36); (c) Oscillatoriales (label 37)

Despite of the majority of overall purities being 0.9 or higher, some labels clustered will only get overall purity slightly over 0.7. For example, clustering the species Peridiniella catenata single, Prorocentrum cordatum and Uroglenopsis sp (labels 42, 43 and 49) or species Snowella Woronichinia, Thalassiosira levanderi and Uroglenopsis sp (labels 47, 48 and 49), they will mingle in such way that the overall purity is only roughly 0.72. The inferior purity of the clustering would imply a similar appearance for the plankton species. Figure 12 illustrates the similarity between species Snowella Woronichinia, Thalassiosira levanderi and Uroglenopsis sp.



(a) (b) (c)

Figure 12. Visualization of plankton species with inferior purity: (a) Snowella Woronichinia (label 47); (b) Thalassiosira levanderi (label 48); (c) Uroglenopsis sp (label 49)

Figures 13, 14 and 15 shows the visualization of the test label sets and the results for clustering those sets. First two clustering results seems fairly pure with only few if any dot changes. The last clustering with species Chaetoceros sp, Skeletonema marinoi and Thalassiosira levanderi (labels 6, 46 and 48) conversely shows extensive changes in dot colors when they go through the clustering process. This produces the impurity into the cluster. In Table 2 the purity for label set 6, 46 and 48 with label set size 100 is only 0.7859.

Figure 16 shows another example of labels mixing in the clustering process. The figure displays the clustering for species Peridiniella catenata single, Prorocentrum cordatum and Uroglenopsis sp (labels 42, 43 and 49) which had overall purity of 0.7272 in Table 2.

Anomalies are possible when clustering data. An anomaly where there is only one image in the cluster was discovered when executing this experiment. Only one image in the cluster results the cluster to automaticly having a perfect purity and therefore increases the overall purity of the clustering. Consequently the success of the clustering is interpreted higher than it actually would be.

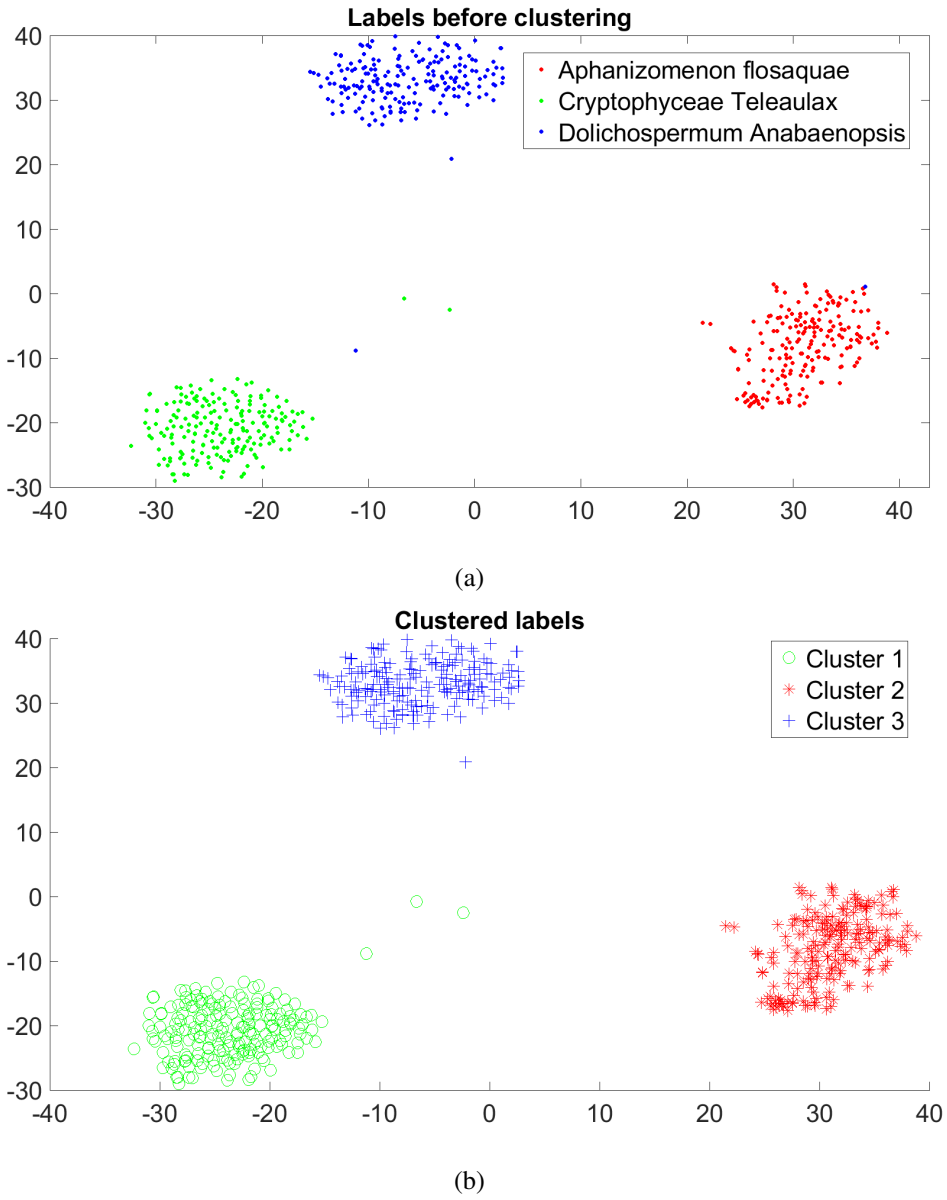
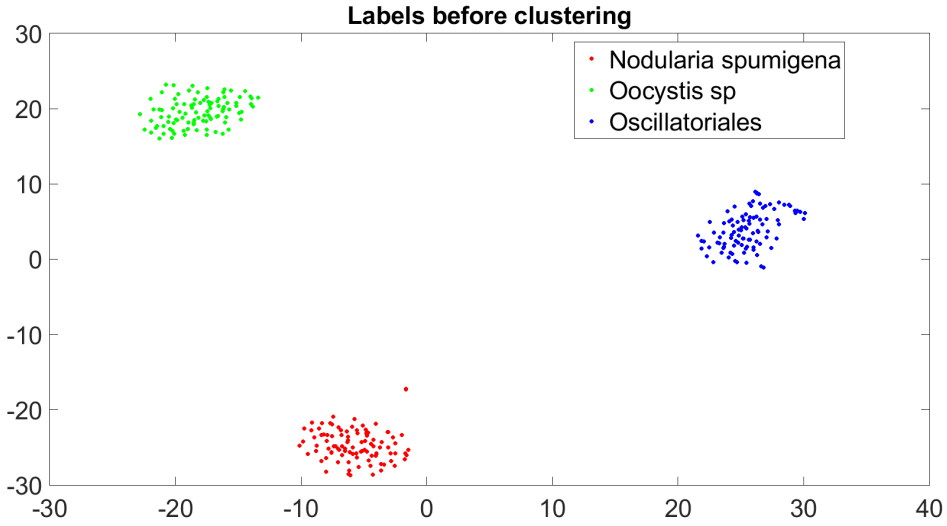
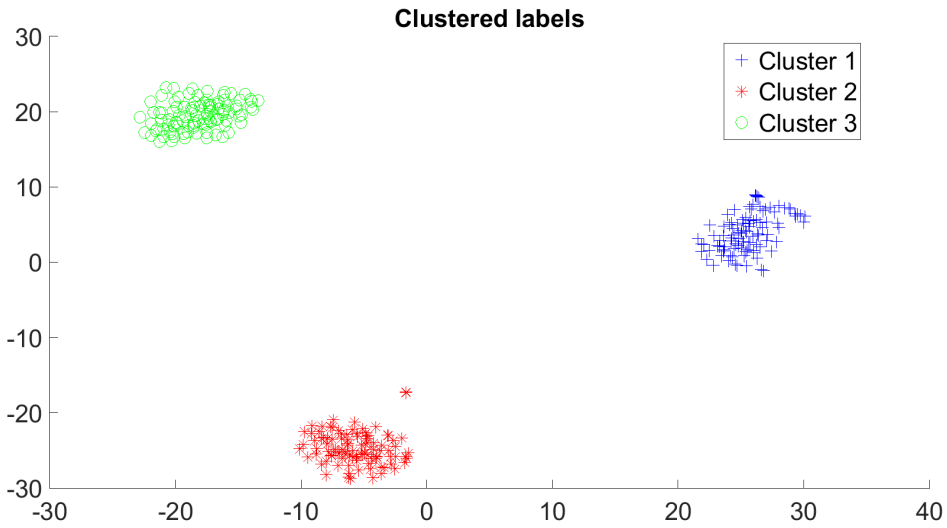


Figure 13. Example of labels and their clustering: (a) Labels before the clustering; (b) Same labels after the clustering

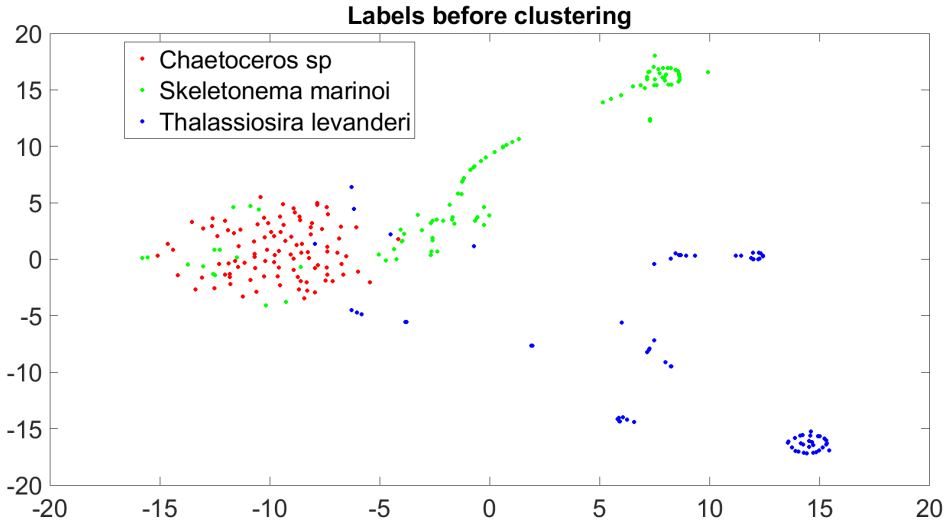


(a)

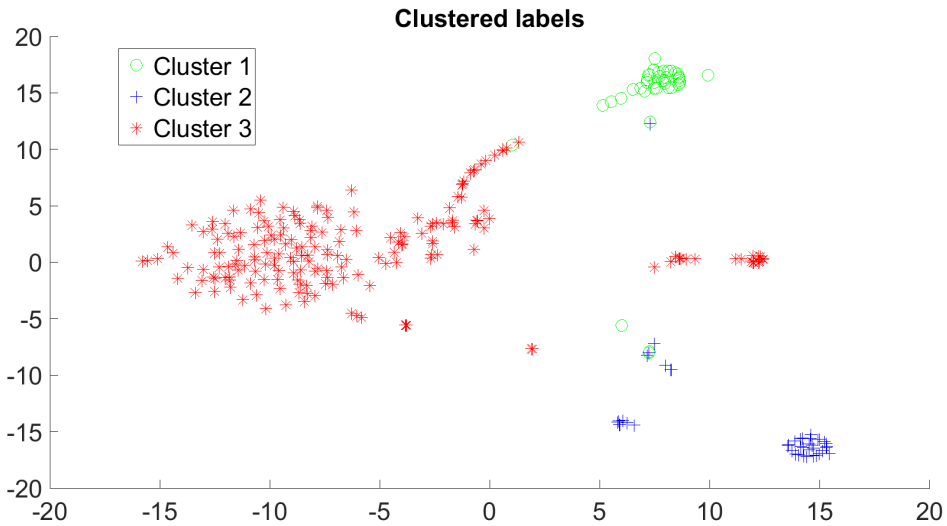


(b)

Figure 14. Example of labels and their clustering: (a) Labels before the clustering; (b) Same labels after the clustering

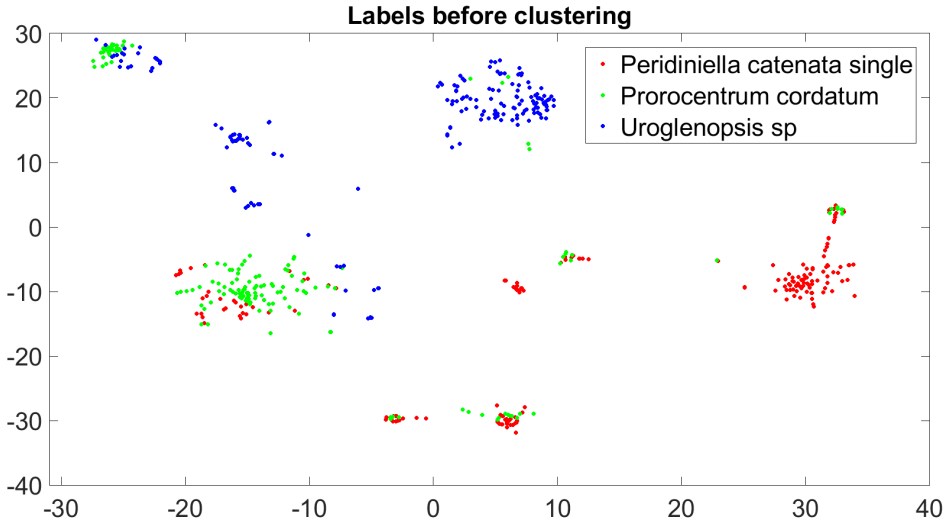


(a)

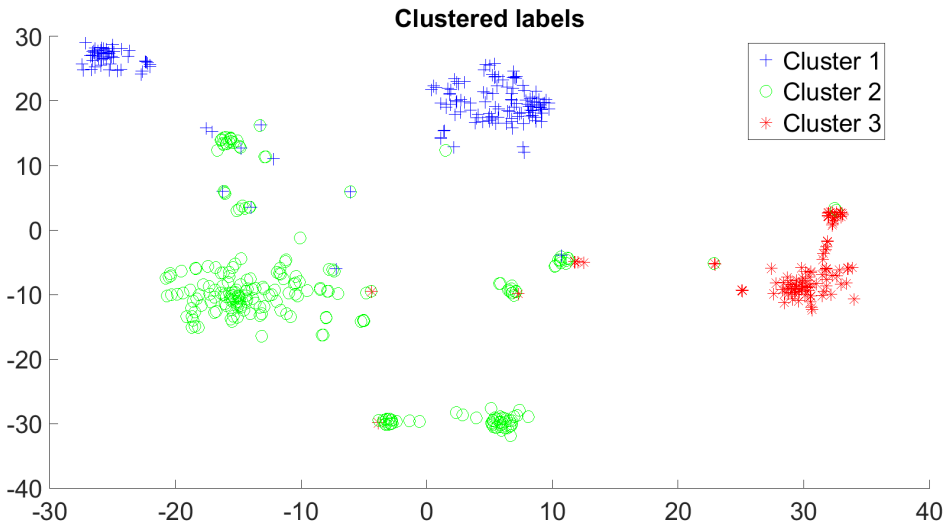


(b)

Figure 15. Example of labels and their clustering: (a) Labels before the clustering; (b) Same labels after the clustering



(a)



(b)

Figure 16. Example of labels and their clustering: (a) Labels before the clustering; (b) Same labels after the clustering

6 CONCLUSION

The objective of this research was to study existing clustering methods in order to determine how well does they succeed on categorizing embedding vectors built from plankton images. Few different clustering methods were reviewed and research was done on how would they work for the plankton data clustering.

Based on the research made, K-Medoids clustering method was selected to be utilized when grouping plankton based on their similarities. Similarity between plankton was based on the embedding vectors that were extracted from the plankton images using a CNN trained in a previous research in the field.

The clustering purity calculation and visualization of the clusters were implemented. Clustering results were analysed and discussed. Additionally the possibility of encountering anomalies with clustering was mentioned and one example was given that had been discovered during the research.

Results for the proposed method were promising but it is worth considering any anomalies in the cluster data such as clusters including only one value.

REFERENCES

[1] NOAA. What are plankton? https://oceanservice.noaa.gov/facts/plankton.html, 2021. [Online, accessed: 24.2.2022].

[2] Rebecca Lindsey and Michon Scott. What are phytoplankton? https://earthobservatory.nasa.gov/features/Phytoplankton/page1.php, 2010. [Online, accessed: 24.2.2022].

[3] Ola Badreldeen Bdawy Mohamed, Tuomas Eerola, Lasse Lensu, Heikki Kälviäinen, and Kaisa Kraft. Open-set plankton recognition using similarity learning. Submitted, 2022.

[4] Lynne Boddy, C. W. Morris, M. F. Wilkins, G. A. Tarran, and P. H. Burkill. Neural network analysis of flow cytometric data for 40 marine phytoplankton species. *Cytometry*, 15:283–293, 1994.

[5] McLane Research Laboratories, Inc., https://mclanelabs.com/imaging-flowcytobot/. *Imaging FlowCytobot*. [Online, accessed: 24.2.2022].

[6] Kaisa Kraft, Jukka Seppälä, Heidi Hällfors, Sanna Suikkanen, Pasi Ylöstalo, Sílvia Anglès, Sami Kielosto, Harri Kuosa, Lauri Laakso, Martti Honkanen, Sirpa Lehtinen, Johanna Oja, and Timo Tamminen. First application of IFCB high-frequency imaging-in-flow cytometry to investigate bloom-forming filamentous cyanobacteria in the baltic sea. *Frontiers in Marine Science*, 8:594144, 2021.

[7] Boxuan Zhong, Qian Ge, Bhargav Kanakiya, Ritayan Mitra, Thomas Marchitto, and Edgar Lobaton. A comparative study of image classification algorithms for foraminifera identification. In *2017 IEEE symposium series on computational intelligence (SSCI)*, pages 1–8, 2017.

[8] Sirpa Lehtinen, Timo Tamminen, Robert Ptacnik, and Tom Andersen. Phytoplankton species richness, evenness, and production in relation to nutrient availability and imbalance. *Association for the Sciences of Limnology and Oceanography*, 62:1393–1408, 2017.

[9] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4690–4699, 2019.

[10] Filip Radenović, Giorgos Tolias, and Ondřej Chum. Fine-tuning cnn image retrieval with no human annotation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41:1655–1668, 2018.

[11] T. Madhulatha. An overview on clustering methods. *IOSR Journal of Engineering*, 2:719–725, 2012.

[12] Dey Debomit. Ml | hierarchical clustering (agglomerative and divisive clustering). https://www.geeksforgeeks.org/ml-hierarchical-clustering-agglomerative-and-divisive-clustering/, 2021. [Online, accessed: 27.2.2022].

[13] J. MacQueen. 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, 1967.

[14] Arnold Ludovic. Unsupervised example: Clustering and k-means. https://ludovicarnold.com/teaching/optimization-machine-learning/unsupervised-example-clustering-k-means/. [Online, accessed: 3.5.2022].

[15] Srivignesh Rajan. Overview of clustering algorithms. https://towardsdatascience.com/overview-of-clustering-algorithms-27e979e3724d, 2020. [Online, accessed: 24.2.2022].

[16] Michael R Anderberg. *Cluster analysis for applications: probability and mathematical statistics: a series of monographs and textbooks*, volume 19. Academic press, 2014.

[17] Oded Maimon and Lior Rokach. *Data Mining and Knowledge Discovery Handbook*. Springer Science & Business Media, 2006.

[18] Cornellius Yudha Wijaya. Breaking down the agglomerative clustering process. https://towardsdatascience.com/breaking-down-the-agglomerative-clustering-process-1c367f74c7c2, 2019. [Online, accessed: 24.2.2022].

[19] Alireza Entezami, Hassan Sarmadi, and Behzad Razavi. An innovative hybrid strategy for structural health monitoring by modal flexibility and clustering methods. *Journal of Civil Structural Health Monitoring*, 10:845–859, 2020.

[20] Patel Abhishek and Singh Purnima. New approach for k-mean and k-medoids algorithm. *International Journal of Computer Applications Technology and Research*, 2:2, 2013.

[21] Manimaran. Clustering evaluation strategies. https://towardsdatascience.com/clustering-evaluation-strategies-98a4006fcfc, 2019. [Online, accessed: 7.4.2022].

[22] Enrique Amigo, Julio Gonzalo, Javier Artiles, and Felisa Verdejo. A comparison of extrinsic clustering evaluation metrics based on formal constraints. *Springer Science+Business Media*, 12:461–486, 2009.

[23] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*, volume 39. Cambridge University Press, 2008.

[24] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[25] Kaisa Kraft, Otso Velhonoja, Jukka Seppälä, Heidi Hällfors, Sanna Suikkanen, Pasi Ylöstalo, Sílvia Anglès, Sami Kielosto, Harri Kuosa, Sirpa Lehtinen, Johanna Oja, and Timo Tamminen. Syke-plankton_IFCB_2022. http://doi.org/10.23728/b2share.abf913e5a6ad47e6baa273ae0ed6617a, 2022.