Lappeenranta-Lahti University of Technology LUT
School of Engineering Science
Computational Engineering and Technical Physics
Computer Vision and Pattern Recognition

Ghebrehiwet Hagos

PLANKTON RECOGNITION USING SIMILARITY LEARNING

Master's Thesis

Examiners:	Prof. Heikki Kälviäinen
	Prof. Vyacheslav P. Shkodyrev
Supervisors:	D.Sc. Tuomas Eerola
	Prof. Lasse Lensu
	Prof. Heikki Kälviäinen

ABSTRACT

Lappeenranta-Lahti University of Technology LUT
School of Engineering Science
Computational Engineering and Technical Physics
Computer Vision and Pattern Recognition

Ghebrehiwet Hagos

Plankton recognition using similarity learning

Master's Thesis

2021

52 pages, 16 figures, 12 tables, 4 appendices.

Examiners: Prof. Heikki Kälviäinen
Prof. Vyacheslav P. Shkodyrev

Keywords: computer vision, image processing, metric learning, similarity learning, Siamese network, Triplet loss function, plankton

Several automated classification methods for plankton images have been developed. These methods typically require an explicit description of features, data augmentation, and are not suitable for classes with a few example images. In recent years, deep metric learning has become a popular technique for many applications. This thesis focuses on classifying plankton images using a convolutional neural network (CNN) based on metric learning. The basic idea is to develop a model trained with a Triplet loss function using CNN architecture for feature extraction. The training model decreases the distance between similar objects and increases the distance between dissimilar objects. The classification is done by searching the image from the set of examples with minimum distance to the query image. The result shows that the accuracy of most plankton classes is good with similarity learning even with a few example images in an unbalanced dataset without data augmentation. However, some plankton images were not classified correctly. Further investigating of architectures and similarity classifier is needed to improve the accuracy.

PREFACE

First of all, I would like to say thank you to my supervisors, D.Sc. Tuomas Eerola, Professor Lasse Lensu, and Professor Heikki Kälviäinen for their strong encouragement, assistance, and guidance.

Secondly, I would like to express my gratitude to the Lappeenranta-Lahti University of Technology and Peter the Great St. Petersburg Polytechnic University for allowing me to participate in this Double Degree program.

Lappeenranta, June 1, 2021

Ghebrehiwet Hagos

CONTENTS

LIST OF ABBREVIATIONS

2-D	Two-dimensional
ANN	Artificial neural network
CNN	Convolutional neural network
CV	Computer vision
DenseNet	Densely connected convolutional network
DML	Deep metric learning
FP	False Positive
IFCB	Imaging Flow Cytobot
K-NN	K- nearest neighbors
MAX	Maximum
MIN	Minimum
MSE	Mean squared error
MLP	Multi-layer Perceptron
PCA	Principal component analysis
ReLU	Rectified linear unit
ResNet	Residual network
RF	Random forest
sp.	Species
STD	Standard deviation
SVM	Support vector machine
SYKE	Finnish Environmental Institute
TP	True Positive
VGG	Visual geometry group

1 INTRODUCTION

1.1 Background

Planktons [1] are varied size aquatic microorganisms without the ability to move against a water current direction. In general, plankton is subdivided into phytoplankton and zooplankton. Phytoplankton is a plant-type class of plankton, which produce their food through a process of photosynthesis. In [2] marine phytoplankton is a groundwork of marine ecosystems and plays a great role in the marine ecological environment and aquatic diversities. Phytoplankton also has an important role in the biochemical cycle of elements. Phytoplankton is the ocean's main plant which has a great role in the marine food chains. Phytoplankton has proved to be also a good environmental indicator and a great contributor to the oxygen of the earth's atmosphere [1]. The marine zooplankton is an animal-type that differs both taxonomically and structurally and feeds other plankton. Zooplankton is varied in size from small unicellular species to jellyfish (several meters in diameter) [3].

The species distribution of plankton populations varies in size and changes rapidly [4]. Identify plankton in species level is hard and requires careful research on each type of plankton population. Traditional ways of collecting a plankton sample and manually analyzing it with a microscope are very laborious [5]. Due to the recent development of automation, the modern imaging devices can capture larger volume images. Classification of the collected plankton images into species is a difficult task for a human being. This leads to a need for automated identification and classification of images of plankton species using computer vision and image processing methods. Examples of phytoplankton images of different classes are presented in Figure 1.

Computer vision (CV) teaches machines in a way similar to what humans would do to interpret imagery. Classification or identification of plankton species can be done using CV and image processing techniques. Usually, a CV system operates in four phases: collecting images, preprocessing images, extracting features, and making decisions [6]. Different methods had been proposed for classification, such as support vector machine (SVM) [7], random forest (RF) [8] and neural network [9].

Traditional machine learning techniques required feature engineering of data before classification, but deep learning techniques use data directly to learn the image features [10, 11]. A CV system feature extraction stage extracts predefined image features from the input files. The recognition of the input images is carried out in the decision making process based on these features. These methods require an explicit description of features, data preprocessing, and a large dataset for training [12]. A convolutional neural network (CNN) was designed for extraction of features from image to improve the classification accuracy. However, CNN is a discriminative model, that is not well suited to classify classes with a few samples. In recent years deep metric learning addressed this problem for many applications.

This master's thesis is part of the FastVision project [13]. The objective is to classify plankton images using the similarity learning technique. The features of two input images are extracted using two CNN with joint or shared parameters to optimized similarity. Then, based on the extracted features calculate the distance between the two images using distance metric methods such as Euclidean distance, cosine distance. The CNN for feature extraction is trained, such a way that the distance is reduced between similar images and is increased between dissimilar images.

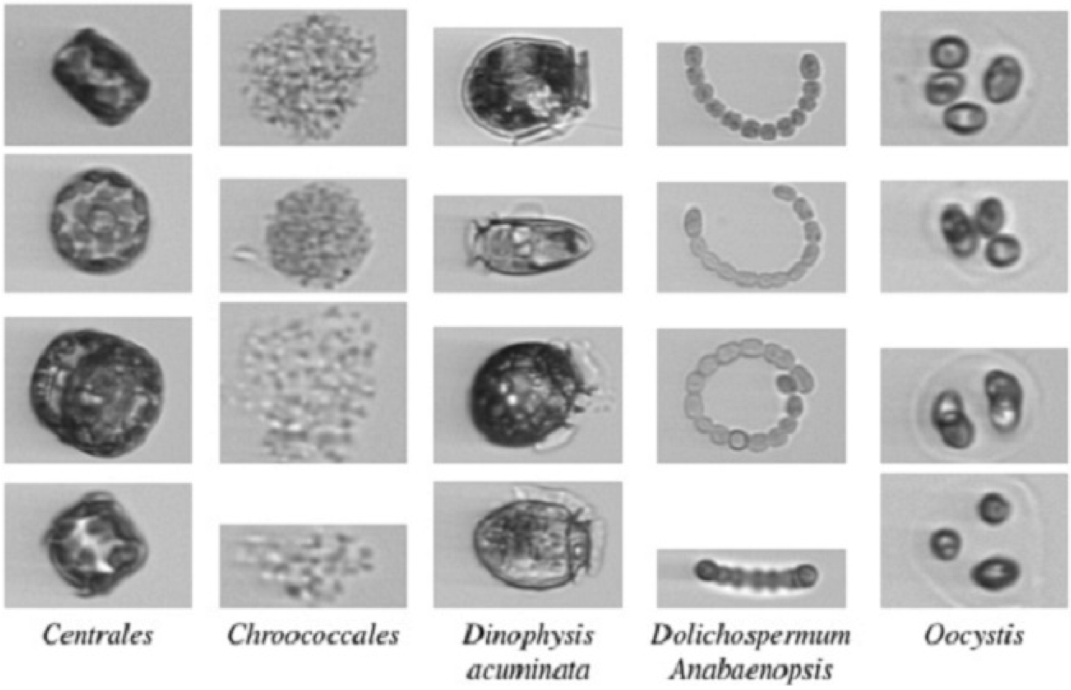


Figure 1. Example phytoplankton images. [6]

1.2 Objectives and delimitations

The main objectives of this research are as follows:

1. To implement and to train CNN-based on similarity learning for plankton images.
2. To develop a plankton classification system by utilizing the implemented similarity learning.
3. To evaluate the developed recognition system plankton image data.

This study deals primarily with phytoplankton, but the classification technique utilized in this thesis work can applicable be for other plankton group datasets.

1.3 Structure of the thesis

Chapter 2 explains an introduction to neural networks and components of CNN and how to train the networks. Chapter 3 describes plankton imaging and methods of existing automatic recognition of plankton. Chapter 4 discusses basic concepts of deep metric learning (DML), how to learn CNN-based on similarity learning, how to classify images using similarity learning and plankton recognition using similarity learning approach and methodology. Chapter 5 describes the data, the experiments and the results of the experiments. The current work as well as possible future work, are discussed in Chapter 6. Finally, in Chapter 7, the conclusions are presented.

2 CONVOLUTIONAL NEURAL NETWORK

2.1 Neural networks

The neural network system is able to formulate the program without explicitly programming. This is done by training a system to analyze patterns, recognize, classify and predict a result with given data. A neural network contains learnable parameters (weight and bias) and training related hyperparameters (learning rate, the batch size, or the number of epochs). In neural network learning the model repeatedly modify the weights until gets the desired output.

2.1.1 Neuron

The human brain process is a very efficient process that can adapt itself to perform a large range of tasks by learning during its lifelong process. A neuron is made up of a body called soma with several branches in it. It interacts with other neurons by transmitting electrical signals. These electrochemical junctions are called synapses. It fires a sequence of spikes across the axon when the inputs reach a certain threshold. The nucleus (a central part of the neuron) returns to its stationary state while a signal is fired. When it enters this state, it stops shooting and fired the signals are transmitted to other neurons. At last, the signal is transmitted from one neuron to another by synaptic connections [14]. A biological neuron is shown in Figure 2.

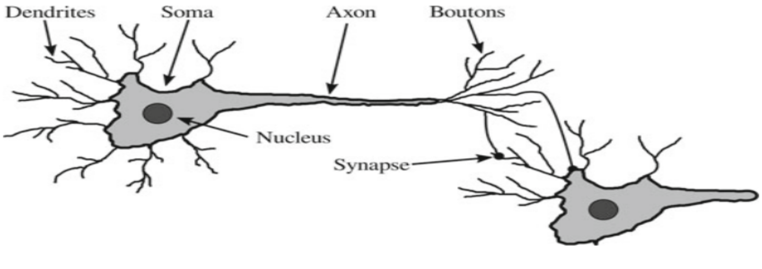


Figure 2. Diagram of a biological neuron. [14]

An artificial neural network (ANN) is a computer model enthused by the brain like assembly of neural networks to solve a problem with high complexity. The structure of artificial neuron as shown in Figure 3 perform similar to the human brain neurons [14]. Firstly, the input value is computed by an artificial neuron. Then, the bias value is added after multiplying the input by a transposed weight vector. At last feed to an activation function g(x) (as in Figure 3), which accepts a real number and return a real number based on the activation function of the neuron by applying computation. The mathematical function of neuron f(x) can be formulated as [14]:

$$f(x) = g(wx^T + b) \tag{1}$$

where x is the input of neuron, w is weighted vector and b is bias.

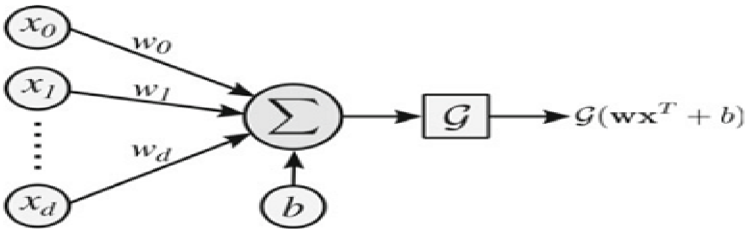


Figure 3. Diagram of an artificial neuron. [14]

2.1.2 Feedforward network

The concept behind neural network is that, through connections defined by the network architecture, several neurons can be linked together to perform complex computations. A neural network's structure is typically described as a graph with nodes representing neurons and edges connecting one neuron's output to another neuron's input [8]. A feedforward network is one type of neural network consists of one or more layers in which at least one neuron is included in each layer. In general, the input vector is supplied to a network input layer, then hidden layers are processed sequentially and a final result is provided to an output layer. Figure 4 describes an example of a fully connected feedforward neural network.

For any continuous function, the feedforward neural network with enough neurons may serve as a universal approximator [14]. It is necessary to find appropriate architectural parameters for a given task when designing such a network, such as the number of hidden layers, the number of neurons in each layer, and the type of activation functions for each layer.

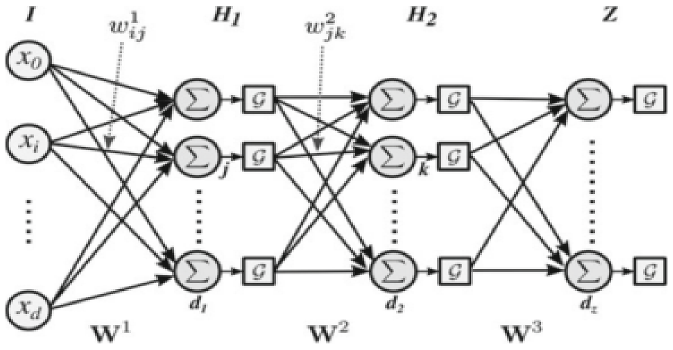


Figure 4. An example of a fully connected feedforward neural network. [14]

2.1.3 Network training

Machine learning is available in three forms: supervised learning, unsupervised learning and reinforcement learning [15]. The most widely used approach for classification is supervised learning, where learning is carried out by training a labeled input data value with the intended output. The objective of classification is to learn a model for correctly classifying input data into predefined categories and provide this information based on a discrete set of labels. Unsupervised learning is another type of machine learning, which learning by training unlabeled input data.

The value of weights and biases are the key parameters for the neural networks to learn. The training of a neural network is to find weights and biases that approximate the network output y for each input x. To guide the training processes, a loss function must be established to calculate how well this target is achieved [16]. The loss function can be, for example, mean squared error (MSE) defined as [16]:

$$G(w,b) = \frac{1}{2n}\sum ||y(x) - d||^2 \tag{2}$$

where function $G(w,b)$ represents a quadratic cost function, x is input, n is the number of inputs, w and b are the weight and the bias respectively, d is a vector of desired outputs and $y(x)$ output for the input x from the neural network.

The cost function is minimized by adjusting the values of weights and biases in the training algorithm. Increase in the number of variables and network parameters, the minimizing of the cost function becomes worse. This problem is avoided by a gradient descent method for finding the local minimum. First, an initial point is randomly selected. Then, the gradient descent algorithm iterates up to a maximum number of iterations or up to a defined threshold reached by moving in the direction of a negative gradient of an actual point to update the position [16].

Backpropagation is another effective way of calculating partial derivations of complex functions, which increases the learning speed. The feedforward layer as defined in Equation 1 is followed by multiple hidden layers and an output layer. After completing the output layers, the error's backward propagating starting from the final layer toward the first layer. Backward propagating errors is calculated as [16]:

$$\delta^l = ((w^{l+1})^T \delta^{l+1}) \odot \sigma'(z^l) \tag{3}$$

where $(w^{l+1})^T$ is transposed weight vector, δ error of propagation at $l + 1$ -th layer, σ' is activation function, z^l is input neurons and $\odot$ is an element wise product of vectors. When training the network, backpropagation calculates the backward errors through the network to adjust the network's parameters.

2.2 Convolutional neural networks

CNN is a special type of neural network for the feature extraction and classification of images by utilizes a special type of linear mathematical operation called convolution for grid-like topology data processing [17]. CNN consists of convolution layers, pooling layers, and fully connected layers. To avoid the problem of low computing memory and processing capacity the network can be split depth-wise [18]. In image processing, the CNN is now one of the most widely used image feature extractors. The layers and the structure of some architecture is described in this section.

2.2.1 Building blocks

Convolution

The convolutional layer accomplishes distinct convolution of the input matrix in its most general form [14]. An image or a feature map produces a heap of transforming matrices with different filters and outputs. Let now assume that x and w are on integer and a discrete convolution parameter t defined as [19]:

$$s(t) = (x * w)(t) = \sum x(a)w(t-a) \tag{4}$$

where $s(t)$ estimation of a position of a spaceship, x is the input, w is the convolution and a is the age of measurement.

A scale of filters, a stride, and a zero-padding are different parameters involved in discrete convolution. It is possible to describe the mapping of distinct convolution among 2 two-dimensional (2-D) matrix in the convolution layer as [19]:

$$f(x,y) = M * N(mx, ny) = \sum_{i=0}^{w}\sum_{j=0}^{h} M(i,j)N(mx+i, ny+j) \tag{5}$$

where $f(x,y)$ is the convolved matrix, M is the convolution filter, N is the matrix to be convolved, w and h are the width and length of M respectively, m and n are the vertical and the horizontal strides respectively.

The stride is a parameter that modified a movement of a kernel and variable a and b. Sometimes the size of the stride is greater than 3 and the actual size is calculated as [14]:

$$size = \frac{W-P}{s} + 1 \quad \times \quad \frac{H-Q}{s} + 1 \tag{6}$$

where W and H are width and height of the image respectively, P and Q are the kernel width and height respectively, s is the stride of the convolution operation and $\frac{W-P}{s}$ $+1$ and $\frac{H-Q}{s}$ $+1$ are integer value of the stride and the filter size. Extraction of local features like edges, color, or gradient orientation is done by convolving the input image with multiple filters and can learn with each layer of the network [14].

Pooling

In reducing the dimensionality of the input maps, pooling has a major role [16]. Sometimes the pooling layer is referred to as downsampling. An average pooling and a max-pooling are common pooling methods, where either a maximum or an average value is determined with a pooling window. The most common pooling is max-pooling. The mapping of the function for the max-pooling can be describe as [6]:

$$g(x, y) = \max(N([mx, mx + w], [ny, ny + h])) \tag{7}$$

where g is the pooling matrix, N is the input matrix, m and n are the vertical and the horizontal strides respectively, and w and h are width and height of the pooling window respectively.

Fully connected layer

A layer after the convolutional and the pooling layers of CNN is a fully connected layer if all neurons are fully connected. The main use of a fully connected layer is to accept input from all pooling or convolution layers and predict a final classification decision of the task or final probabilities for each label. Convolution and pooling layers extract features from an image and fully connected layers perform classification based on these extracted features.

2.2.2 Architectures

As individual datasets differ with observed features, there is no golden rule to proceed. Therefore a specific network structure is required. It is a difficult job to construct the CNN [14]. However, some laws come from current, time-proven solutions. The convolution layers typically use 3x3, 5x5, and 7x7 sizes of the convolution filters. Some of the most common examples of architecture of CNN are AlexNet [18] visual geometry group (VGG) [20], and residual network (ResNet) [21].

AlexNet

In 2012, Alex Krizhevsky designed a CNN named AlexNet [18]. The architecture of AlexNet contains up to 60 million parameters, 650 000 neurons, and 8 layers (5 Convolutional Layers and 3 Fully Connected Layers). The input to the AlexNet structure is RGB images of 256 × 256 pixels with first the convolution window layer of 11 x 11 size and reduced into 3 x 3. The last layers of AlexNet are three fully connected layers and one softmax classifier with rectified linear unit (ReLU) for the activation function. The structure of AlexNet architecture is shown in Figure 5.

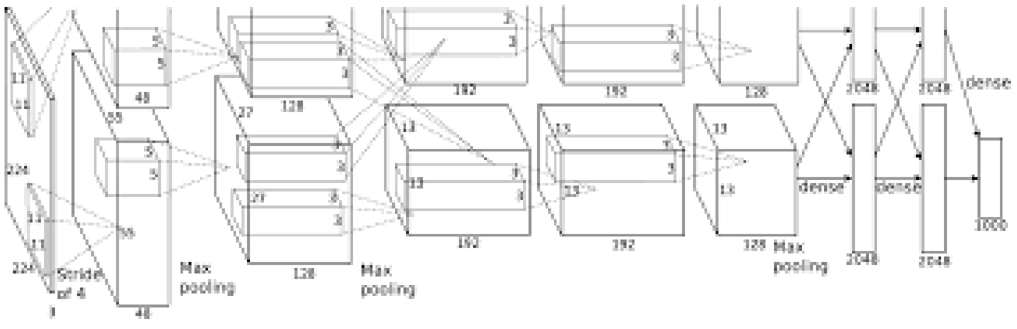


Figure 5. AlexNet architecture. [18]

VGG

In [20], VGG was proposed for large-scale image classification. Different types of VGG with different depth were proposed, such as VGG11, VGG16, VGG19.The structure and layers of VGG16 and VGG19 architectures can be seen in Figure 6.

VGG16 is a model of VGG, which has 16 convolutional layers and a very standardized architecture. VGG16 has 3x3 convolutions. It is currently one of the most popular methods for feature extraction from images. It is used as a feature extractor in a variety of applications.

VGG19 is another version of VGG CNN architecture. The number 19 refers to the number of layers with weights that can be trained. There are 16 convolutional layers and three fully connected layers. The architecture takes a (224, 224, 3) RBG image as the input. In VGG-19, the convolution layers use (3, 3) filters, with the number of filters increasing powers of two (64, 128, 256, 512). Stride length is one (pixel) in all convolution layers, with one (pixel) padding on each side. There are five sets of convolution layers, two of which have 64 filters, two of which have 128 filters, four of which have 256 filters, and the other two sets each have four convolution layers with 512 filters.

ResNet

A ResNet [21] is one of the architectures of CNN, which shortcut connection is added to each pair of 3×3 filters. A shortcut connection is to skip one or more layers. In plain CNN architecture such as AlexNet has a vanishing gradient problem (preventing the weight from altering its worth effectively). A ResNet was developed to avoid vanishing gradient problem while increasing the layers and network parameters. A ResNet allows up to

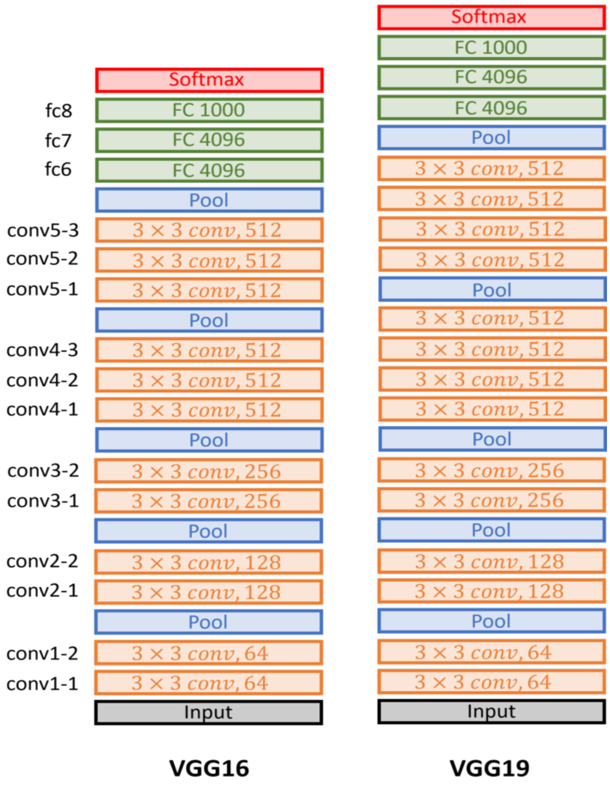


Figure 6. The structure and layers of VGG16 and VGG19 architectures. [22]

hundreds of layers to be trained with persuasive accuracy.

ResNet achieved a smaller error rate on the ImageNet [23] test set among different VGG and Googlenet. Each architecture result was compared by selecting the different depths of the convolution layer with different stride, average pooling, and one fully connected layer with a softmax classifier. The accuracy of a 34-layer architecture was improved by utilizing these parameters in the ImageNet dataset.

DenseNet

Densely connected convolutional network (DenseNet) proposed in [24] achieved high performance for object recognition with less memory and computation for the benchmark tasks. In DenseNet leverages shortcut bonds between layers in the same way as ResNet. In DenseNet each layer is connected with a feedforward mode to every other layer and receives feature maps from every preceding layer. The structure of DenseNet architecture is described in Figure 7.

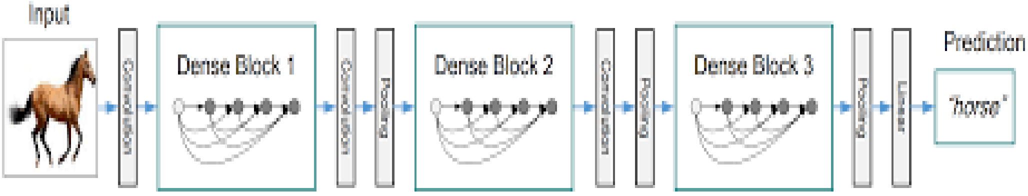


Figure 7. DenseNet Architecture. [24]

2.2.3 Regularization

Regularization is a method for slightly altering the learning procedure so that the model generalizes better. As a result, the efficiency of the model improves in previously unseen data. A classifier is more susceptible to overfitting when a limited amount of training data is available. There are multiple types of methods of preventing overfitting in the neural network. In [25] overfitting was addressed using drop out. Data augmentation [26] is another regularization technique based on generating new data to enlarge the dataset.

Dropout

In [25] a dropout method was proposed for preventing overfitting in a neural network. Dropout involves neural network training to turn off some of the layers of the fully connected layers. Dropout randomly excludes a proportion of neurons in the hidden layers and is used for reducing overfitting and improve the accuracy of the neural networks for supervised learning. The probability of a node not contributing to the network is determined. This probability is used to consider or refuse the node in the network.

Data augmentation

Deep neural networks require a large amount of data for training the network. Data augmentation is one technique to increase the amount of data. For example, in [26] data augmentation was implemented to generate new data to enlarge the training dataset and reducing overfitting. Affine transformations are one technique of data augmentation for correcting and modifying the geometric structure of the image in image processing. A common method of affine transformations is reflection (flipped images around an axis), scale (scaling image along main axes), rotate (rotating image around their center), and shear (shearing image along an axis). Image resizing, image filtering, and cropping images are techniques used in data augmentation.

3 PLANKTON RECOGNITION

In this section, the method of plankton imaging is discussed, followed by reviewing existing feature engineering and CNN-based plankton recognition. A summary of the existing methods of plankton classification is described at the end of the section.

3.1 Plankton imaging

Plankton imaging technology plays an important for studying plankton. Before a development of specific plankton imaging technology, the traditional way of collecting and examining plankton images was done using microscopes. This method is the time consumed and required more skilled manpower. Even though automated imaging technology challenging and complex, it able to detect and collect samples of different sizes at different conditions [27]. Imaging Flow Cytobot (IFCB), FlowCam, FastCam (a prototype system), CytoSense, and CytoSub are some of the modern instruments used for collecting plankton remotely [27].

The imaging IFCB is a submersible instrument that captures high resolution (1um) images and measures chlorophyll fluorescence of nano- and microplankton-sized particles using a combination of flow cytometry and video technology [28]. IFCB monitors the optical properties of actual moves up and down the plankton as they pass through a laser beam. IFCB is usually deployed for months at a time to provide accurate information about the phytoplankton composition. The laboratory operator can interact to keep track of results and make changes to imaging procedures as needed.

In [29] a technique for collecting small plankton images using a flow cytometer (FlowCytobot) was proposed. In FlowCytobot, the plankton cells are passed through a focused laser beam, then optical properties of individual suspended cells are registered. FlowCytobot imaging can be conducted unattended for two months, tracking findings and, if necessary, modifying sampling processes. Such integrated approaches help to discover and study each individual for a long time.

3.2 Automated plankton recognition methods

With the development of imaging technology, a large number of plankton images able to be collected. This technology leads to a need for automatic recognition of planktons. In the last decades, different automatic techniques have been developed for the classification of plankton images. Some of these methods are feature engineering-based, but the most recent ones mainly utilize CNN.

3.2.1 Plankton recognition using feature engineering

Many CV problems were solved with feature-based approaches before convolutional neural networks were commonly used [30]. Feature vectors are extracted from images that define objects form, length, width, color histogram, and many others. These acquired values are supplied to the classifier, such as SVM [7], RF [8] or k- nearest neighbors (K-NN) [8].

The study in [31] studied different handcrafted feature approaches for the automatic classification of diatoms. Diatoms are a type of microorganisms of algae with high species-level biodiversity. The dataset was collected in Spain. Each diatom image was categorized into 80 classes, each of which had an average of about 100 samples. These samples were augmented by rotating and flipping into 300 samples. Each sample consisted of 1460 features. The best result was achieved with a random forest classifier using all features (morphological, textural, space-frequency, statistical descriptors).

In [32] studied the classification of phytoplankton based on the co-occurrence matrices feature and SVM classifier. The dataset used was 20 000 images collected by the plankton video recorder device and labeled by the human into 7 classes. SVM SVM performed better when compared with a learning vector quantization neural network as a classifier with moment invariants, Fourier descriptors, and granulometry as features.

In [33] studied different types of features and classifier for classification of phytoplankton. The dataset was collected from the eastern channel consists of seven different classes, each with 100 samples labeled by a specialist. Each sample contains eight signals describing length, internal structure, chlorophyll pigment, and others. These were transformed into a set of features and classified by K-NN, SVM, and RF. The best accuracy was obtained with RF.

3.2.2 Plankton recognition using convolutional neural networks

CNN is an image recognition method for enhancing performance by extracting features from images. Previously, several numbers of work are done for the classification of plankton images based on CNN. Some of the existing methods are explained in this subsection.

In [34] proposed deep learning with CNN for classifying phytoplankton images (microalgae image dataset obtained with the FlowCAM system). Data augmentation was used to increase the amount of data available for training. Using deep learning with CNN and data augmentation technique achieved a better result than a human being.

The study in [6, 35] proposed CNN-based method for recognition of plankton images captured with an Imaging Flow Cytometer and compare with the existing random forest method. Data augmentation was used to solve the imbalance of classes and a large disparity of images was solved by image resizing and padding. Using of A CNN with AlexNet architecture achieved better performance than existing RF [6]. The method was able to obtain a good result using CNN architecture after data augmentation for imbalanced dataset.

In [36] an attempt to find a more efficient model for plankton classification, such as CaffeNet, VggNet-19 and ResNet was made. The dataset was from National Data Science Bowl with 121 classes of 30336 images. The samples were rescaled to be 256 × 256 common sizes and augmented to increase the data and to prevent overfitting by rotating images. Best accuracy was achieved by ResNet-19 architecture while the CaffeNet architecture takes less storage space and runs faster. A top-5 accuracy of 95% are achieved for this plankton dataset by making CNN smaller.

In [37, 38] CNN-based classifier of plankton images with varying image sizes was studied. The dataset used was provided by the Finnish Environment Institute, was captured by a FlowCytobot (FCBI). Data augmentation and normalization were the data preprocessing mechanisms of the dataset. The method improved Al-Barazanchi architecture [39] by CNN with spatial pyramid pooling layer were used to accept different size, CNN with patch cropping (DeepWriter) [40], Multi-stream CNN (scaling input into multiple different sizes) [41], CNNs with included metadata (adding of information to the original dimensions of the image) [42]. The accuracy of AlBarazanchi architecture was improved using the combination of AlBarazanchi, Ał-Barazanchi_2, and DeepWriter. However, the best accuracy was obtained using the combination of InceptionV3_2 and DeepWriter.

In recent, both CNN and the random forest classifier have been the most common methods for automated plankton classification. Table 1 represents a comparison of some existing methods for plankton classification of different data and the different number of classes in various papers. Some of them were feature engineering-based, and some of them were CNN-based feature extractors. The accuracy of each method was varied with a different dataset and the number of classes. CNN strategies appear to do better than the feature engineering-based approaches for plankton classification. The reason is the process of feature extraction of CNN is superior to feature engineering methods of feature extraction for classification plankton images. Having more classes and an imbalanced dataset decreases the accuracy. Data augmentation was used in the CNN-based feature extraction methods. These existing methods typically require further preprocessing such as data augmentation and are not suitable for classes with a few example images.

Table 1. Existing plankton classification summary.

Publication	Data	Classes	Samples	Accuracy
Bueno *et al.* [31]	Diatom	80	8000	97.56
He *et al.* [32]	Plankton images	7	20 000	72.00
Correa *et al.* [34]	Microalgae	19	29,449	88.59
Grönberg *et al.* [6]	Phytoplankton	43	2150	82.70
Yan *et al.* [36]	Plankton dataset	121	30,336	96.00
Bureš *et al.* [37]	Microplankton	44	2200	96.20

4 METRIC LEARNING

4.1 Background and basic concepts

The primary goal of metric learning is to learn a similarity between images using distance metrics in machine learning. To learn metrics, reduce the distance between similar images and increase the distance between dissimilar images. A metric learning technique that conducts the data learning process would have a greater ability to differentiate between the sample data [10]. Metric learning has been successfully employed in the area of face recognition [43], image classification [44], and person re-identification [45].

4.1.1 Deep metric learning

In recent years DML has been proposed, which uses distance metric for learning tasks. The aim of DML is to train the neural network with network parameters of weights and biases [46]. The main parts of DML are insightful input data, the network model architecture, and a metric loss function. As the name suggests, metric learning is a technology for mapping images to a metric space. In mapping, images from the same class get closer while images from different classes get farther from each other.

4.1.2 Siamese networks

A Siamese network is a family of neural network architectures with two or more similar subnetworks that share the same weights and parameters. Siamese networks accept one input for each of the subnetworks. Then each network's extracted feature is used for comparing the similarity of these inputs. Two input points (x_i and x_j) are similar if they are from the same class or dissimilar for different classes. DML based Siamese network is trained to minimize the contrasting loss as [46]:

$$L(\{w^{(m)}, b^{(m)}\}_{m=1}^{M}) = \sum_{(i,j)\in S} h(d_f(x_i, x_j) - T_1) + \sum_{(i,j)\in D} h(T_2 - d_f(x_i, x_j)) \tag{8}$$

where T_1 and T_2 are positive thresholds $T_1 < T_2$, h(x) max(0,x), $d_f(x_i,\ x_j)$ the distance between two points x_i and x_j and S(i,j) and D(i,j) are set of index of similar and dissimilar pair respectively.

A fewer amount of data is needed to learn based on the similarity of data. In [47, 48] dimensionality reduction by learning to map and a similarity metric for face verification with Siamese networks was proposed. The approach does the non-linear mapping of the images to low dimensions and learns the similarity of two inputs by a distance metric.

4.1.3 Triplet learning

An extension of the Siamese networks is the triplet network with three input images in a training phase. These inputs are an anchor (x_i), positive example (x_i^+) and negative example (x_i^-). The triplet network learns with the Triple loss function to reduce the distance between images from the same class and increase the distance between images of different classes. Triplet networks are learned using the Triplet loss function. The Triplet loss function is calculated as [46]:

$$L(\{w^{(m)}, b^{(m)}\}_{m=1}^{M}) = \sum h(T + d_f(x_i, x_i^+) - d_f(x_i, x_i^-)) \tag{9}$$

where $d(x_i, x_i^+)$ and $d_f(x_i, x_i^-)$ are distances, $h(x)$ loss function and T is a margin or threshold.

In [49] DML based triplet network architecture was proposed to learn by distance metric between the images. Figure 8 shows Siamese network and triplet network architectures.

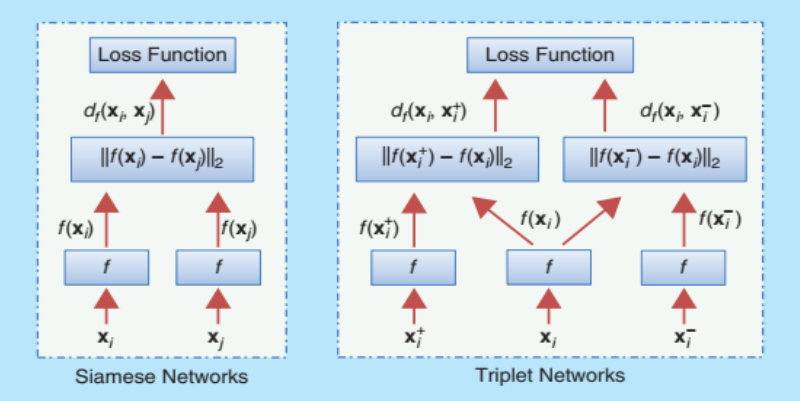


Figure 8. Two main structures of metric learning: Siamese and triplet architecture. [46]

The concept of similarity and dissimilarity in the Triplet loss architecture can be used in learning distributed embedding. A triplet network is a form of neural network architecture with shared weight in which three parallel networks are trained using the Triplet loss function. During prediction time, an input image is routed through a single pre-trained network to compute distributed embedding representations of the input image. It is a form of neural network architecture in which multiple parallel networks with shared weights are trained.

In image similarity learning the cost function for the Triplet loss is as follows [50]:

$$L(A, P, N) = \max(0, d(A, P) - d(A, N) + margin) \tag{10}$$

where $d(x, y)$ is the distance between extracted feature of images, A is the anchor, P is the positive and N is the negative triplet.

4.2 Convolutional neural network-based similarity learning

In a traditional metric learning-based method the feature representation of the data and the metric are not learned together. Similarity learning techniques use the distance comparer approach to define the distance metric-based similarity between images to enhance classification accuracy. The CNN possible solution is to jointly optimize the representation of the input sample based on the similarity measure. Siamese architecture is one of the structures used to learn the similarities between images. Siamese network training is to produce average weight by training the two CNNs separately, but an average weight is shared between the two networks.

In [45] idea of DML is studied training of Siamese based on CNN for person re-identification, to identify a person from two images. The key idea is to learn a similarity metric from the image using DML with two CNNs connected using a cosine layer to measure the distance metric and binomial deviance to evaluate the cost of similarity between the two images [45]. Figure 9 shows the structure of Siamese networks for person re-identification.

In [48] metric learning based on distance similarity was designed for face verification. The method extracted two sets of deep features of the two inputs using a Siamese architecture based on CNN. A minimizing discriminating loss function was used to assign a smaller metric distance of the face image of the same person, and assign a larger metric distance

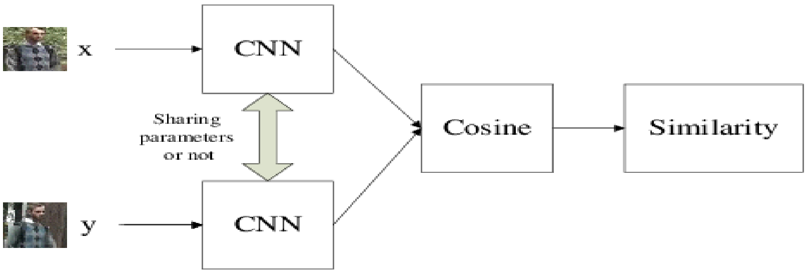


Figure 9. Siamese network for person re-identification. [45]

for the face of different persons.

A method for face verification based on DML using the cosine metric as learning metric was proposed in [51]. To implement the face verification, original images were preprocessed using resizing and cropping. Then, features were extracted using CNN from input images. The extracted features were reduced in dimension using principal component analysis (PCA). The similarity score was measured using the cosine metric distance. The method was tested on the Labeled Faces in the Wild dataset and achieved higher accuracy.

4.3 Image classification using similarity learning

Deep similarity learning performs the classification images by performing 3 main processes. These processes transform the data into vector features using deep learning. The distance is computed between vectored data using distance metrics (e.g., Euclidean, Manhattan distance, Mahalanobis, cosine). Contrastive loss is used to learn the model by putting similar data closer together (i.e., reducing the distance between them) and dissimilar data further away from each other (i.e., increasing distance between them). Classification is obtained by finding the most similar images in the gallery set with example images from known classes.

One way to create image similarity is to use the CNN architecture to extract features and use these features to compute the similarity between these features. In [50], one-shot learning, Siamese networks, and Triplet loss were described for image classification. The work was to classify a digit using similarity learning. The main idea was to represent the data to a new space using one-shot triplet learning in deep learning. Compute the distance and select the minimum distance by a solid similarity comparer. Figure 10 shows the triplet structure for image classification.

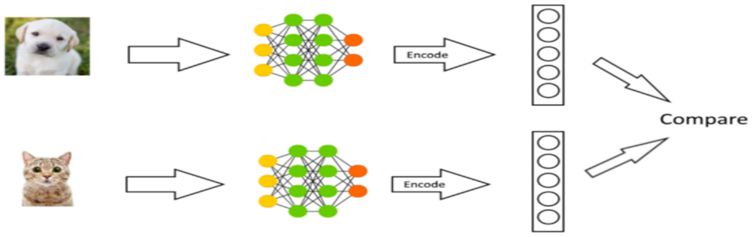


Figure 10. Siamese network for image classification using a similarity comparer. [50]

Similarity learning learns to measure the similarity between two images by using a distance metric. In [52] a Mahalanobis distance was used for the classification of labeled large margin data. K-NN based metrics training has been shown to lead to substantial improvement in the classification of varying size data.

In [53] proposed trainable similarity for a test specifically designed for the second phase classifier. In the second phase classifier, recognition is carried out in two-stage. The similarity measure S(I, J) between two the same images I and J was introduced to produce trainable based on the correlation coefficient and calculated as [53]:

$$S(I, J) = \frac{\sum (I_i - I)(J_i - J)}{\sqrt{\sum (I_i - I)^2 \sum (J_i - J)^2}} \tag{11}$$

where I and J are images, I_i and J_i are the intensity of the image I and J at pixel i^{th}. A fisher separability factor is a similarity-based separation between targeted and non-targeted training objects. The similarity is dependent on a local image similarity in the local image regions. The similarity measure on local is the mean of all measurements and calculated as [53]:

$$S_{\text{mean}}(I, J, R) = \frac{1}{|R|} \sum_{i=1}^{|R|} s(I, J, r_i) \tag{12}$$

where I and J are images, R is a set of local image regions, and $S(I, J, R)$ is the similarity measure. This method tested for pedestrian recognition comprises 7302 images of 100 people to identify the true pedestrian or reject the false pedestrian.

4.4 Plankton recognition using similarity learning

This research is to study the similarity learning approach for plankton image classification. In this subsection, the main steps of plankton image recognition using similarity learning are described. Firstly, the image is preprocessed to the required size and prepare a triplet batch from the preprocessed images. Secondly, build and train the model. Thirdly, extract image features using the trained network and classify the image using the distance similarity comparer.

4.4.1 Data preprocessing

The sizes of plankton images are varied. To be fitted to the CNN model, the raw images should be resized. Images are resized by reducing the larger side and enlarge the smaller to fit the given boundary. This helps to avoid distortions and maintaining the aspect ratio of the original image. During this method, bicubic interpolation is used for resizing image pixels. Then images are padded to fill the remaining space on the boundary. Gaussian noise is applied to reduce artificial edges induced by homogeneous areas. Finally, the images are normalized to change the range of pixel values between 0 and 1. In the study, all images are resized to 224 X 224 size.

4.4.2 Building and train of model

Two Siamese models are used for the evaluation of similarity learning on plankton. Each of the Siamese network architectures is with 3 shared instances of CNN. The models are based on VGG19 and VGG16 CNN architectures for feature extraction. Three different input images pass through three CNN networks. These three networks are the same architecture with shared weight. The output of the Siamese model is the L2 distance between the triplets. Figure 11 shows the structure of triplet metric learning for plankton recognition.

After the image preprocessing and the model are built, triplet (anchor, positive and negative) batches are generated. The anchor and positive images are from the same class and the negative image is from a different class. To generate good triplets, harden triplets are used, with the negative being closer to the anchor than the positive. At each iteration, batches of triplets, consists of hard and random triplets, are created from a large batch of

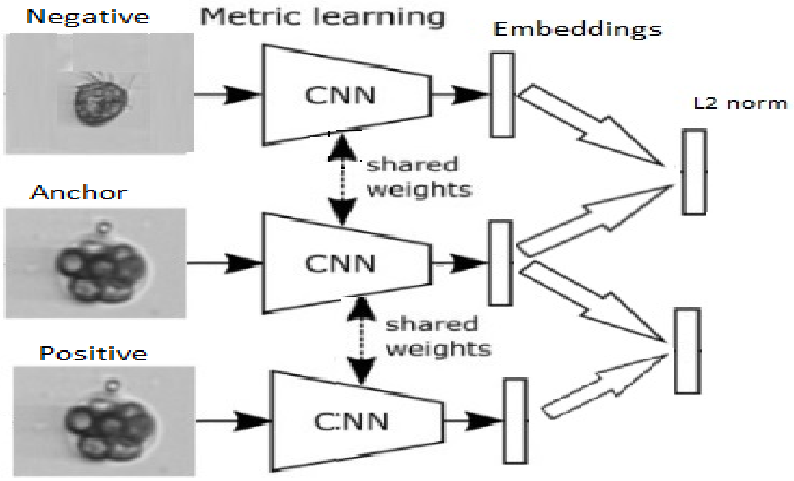


Figure 11. Structure of triplet metric learning for plankton recognition.

200 random triplets. The triple images are fed into the fine-tuned CNN architecture with shared weight.

Triplet loss is a function for the learning of similarity between the images of plankton. The Siamese network receives a triplet input to generate the new embedding. Two intermediate distances (L2) are outputs by feeding 3 images of triplets. Triplet loss distance is computed as in Equation 10. The network is trained to produce a new embedding space directly correspond to plankton image similarity, where the distance between the similar images is small and the distance between the dissimilar images is large.

4.4.3 Classification

After the model is trained to produce a new embedding, the experiment of similarity learning for the classification of plankton images is obtained using distance similarity comparer. To use the distance similarity comparer needs to prepare gallery set images from the train data. Figure 12 shows the classification steps using the distance comparer. Gallery set images are obtained from the train set. These images are selected randomly from each class. Query images are a set of the image to be classified. The distance between each query image to all images in the gallery set is computed to find the most similar one. It is important to select images that represent well to the query images.

To use metric similarity measurement, similarity metric such as L1-norm (Manhattan

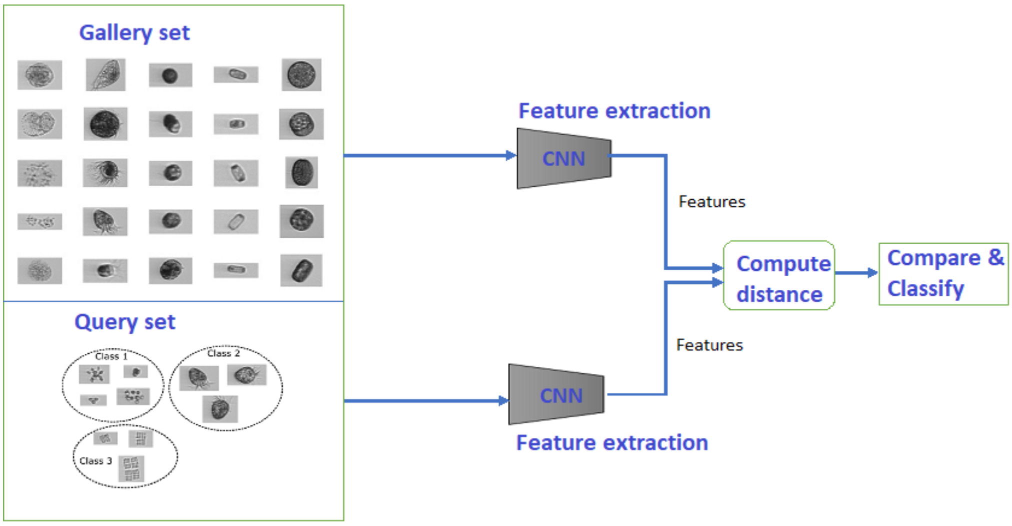


Figure 12. Image plankton classification by similarity comparer between the selected gallery and query set.

distance) and L2 norm(Euclidean distance) is defined. For the classification of plankton images, the Euclidean distance is used. If f is a neural network function for input x, $f(x)$ is its output. The Euclidean distance between $f(x_i)$ and $f(x_j)$ is calculated as [46]:

$$d(x_i, x_j) = d(f(x_i), f(x_j)) = ||f(x_i) - f(x_j)|| \tag{13}$$

where x$_i$ and x$_j$ are input images and $f(x_i)$ and $f(x_j)$ are features extracted from the neural network i.e., CNN.

After preparing the gallery and query sets, each image is fed to the trained CNN network to extract feature vectors. The Euclidean distance between each query image feature and the gallery image features are computed using the extracted feature vectors. Then, the calculated distances are sorted. Finally, the query image is classified to the class with minimum distance in the gallery image.

5 EXPERIMENTS

5.1 Data

The plankton dataset used in the study is supplied by the Marine Research Centre of a Finnish Environmental Institute (SYKE) [54] and is linked to the FastVision project [13]. These images were collected using IFCB. The dataset contains images of 50 labeled plankton classes (species). As seen in Table 2, each class contains a different number of images per class, which ranges from 19 (Amylax triacantha) images to more than 12000 (Dolichospermum anabaenopsis) images per class. There is quite large variance in both aspect ratio and size of images. Figure 13 shows examples of phytoplankton images.

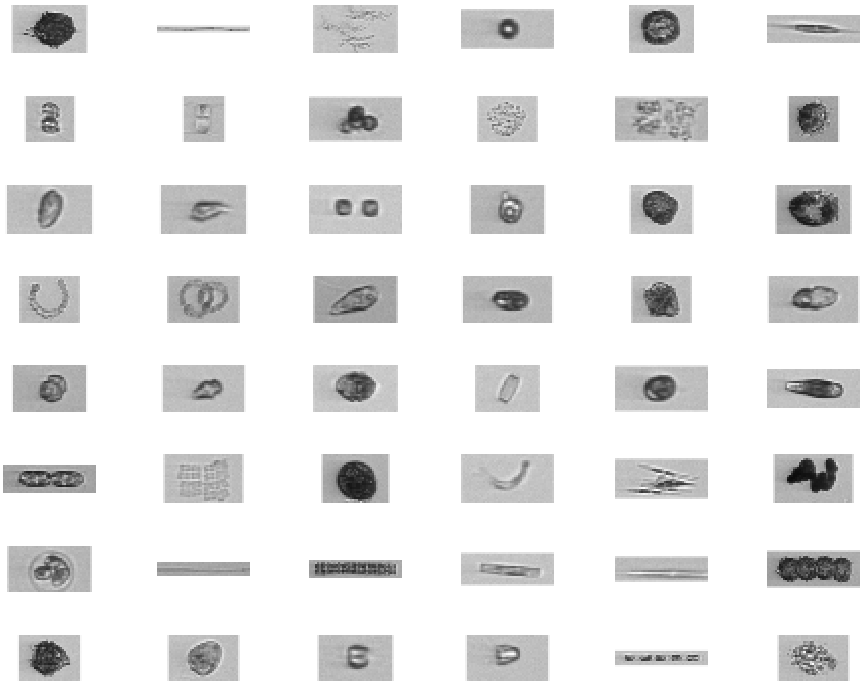


Figure 13. Example phytoplankton images in the data.

Table 2. Class names and number of images per class in the dataset.

Class name	**Images**	**Class name**	**Images**
Amylax triacantha	19	Aphanizomenon flosaquae	6989
Aphanothece paralleliformis	29	Beads	125
Centrales sp.	480	Ceratoneis closterium	45
Chaetoceros sp.	1382	Chaetoceros sp. single	213
Chlorococcales	95	Chroococcales	142
Chroococcus small	827	Ciliata	243
Cryptomonadales	713	Cryptophyceae teleaulax	6830
Cyclotella choctawhatcheeana	102	Cymbomonas tetramitiformis	199
Dinophyceae	1433	Dinophysis acuminata	217
Dolichospermum Anabaenopsis	12280	Dolichospermum anabaenopsis coiled	2504
Euglenophyceae	102	Eutreptiella sp.	2247
Gonyaulax verior	22	Gymnodiniales	69
Gymnodinium like	158	Heterocapsa rotundata	614
Heterocapsa triquetra	3276	Heterocyte	263
Katablepharis remigera	54	Licmophora sp.	74
Melosira arctica	43	Merismopedia sp.	98
Mesodinium rubrum	1132	Monoraphidium contortum	327
Nitzschia paleacea	65	Nodularia spumigena	169
Oocystis sp.	842	Oscillatoriales	4440
Pauliella taeniata	119	Pennales sp. thick	210
Pennales sp. thin	781	Peridiniella catenata chain	193
Peridiniella catenata single	899	Prorocentrum cordatum	276
Pseudopedinella sp.	379	Pyramimonas sp.	1224
Skeletonema marinoi	4128	Snowella woronichinia	2956
Thalassiosira levanderi	2537	Uroglenopsis sp.	516

5.2 Evaluation criteria

To evaluate the performance of the models, in this work, accuracy was used as the measurement tools. The percentage of accuracy for each class was calculated as:

$$Accuracy(\%) = \frac{Number\ of\ correctly\ classified\ images}{Total\ number\ of\ test\ images} \times\ 100\% \tag{14}$$

$$= \frac{TP}{TP + FP} \times\ 100\% \tag{15}$$

where True Positive (TP) is the number of the correctly classified images and False Positive (FP) is the number of the incorrectly classified images.

The classification of each model is evaluated from top-1 up to top-5 accuracy. Top-1 accuracy is the accuracy of the first most similar images must be the same as the desired label. Top-5 accuracy means that any of the five highest similar images must be the same as the desired label.

The confusion matrix was used to represent the result of each class accuracy more visually. The evaluation process was taken on the test dataset by selecting the minimum distance between the query image and gallery images. For each class, the maximum (MAX), mean, standard deviation (STD), and minimum (MIN) of the accuracy were computed.

5.3 Description of experiment

The research focused on the similarity learning method to classify the plankton images. The models were trained in the Triplet loss technique using a one-shot learning approach. The Triplet loss learning maps a plankton image to a new distance space, which is easy to classify using the distance similarity. Choosing good triplets inputs to the network is needed during training to map each class to a new hyperspace.

For the experiment, the dataset was subdivided into 3 subsets. Subset 1 includes classes of at least 1000 images each. Subset 2 includes at least 100 images each. All the classes are included in Subset 3. Table 3 shows subsets used for the experiment.

Table 3. The dataset subsets.

Dataset	Class size	Number of classes
Subset 1	>=1000	14
Subset 2	>=100	39
Subset 3	>10	50

Each subset was split into 20% for testing and 80% for training. Each sample contains a maximum of 1000 images for training and a maximum of 60 images for testing. Triplet's batches of the train data were prepared at each iteration using hard triplets. Each of the batch's samples includes an anchor image, a positive image, and a negative image.

In the experiment, two Siamese models with the Triplet loss function were evaluated to classify plankton images. The first model was based on pre-trained VGG19, whereas the second model was based on pre-trained VGG16 CNN architecture for feature extraction. Feature vectors were trained and similar images were pulled close together and dissimilar images were pushed away. To reduce the number of parameters, the top layers of the architecture were frozen. Then, two dense layers with 4096 and 50 units were added for better embedding (smaller representation of the images). The forced margin between triplets was added to prevent the network from learning a simple solution. The final layer does not have any activation function to have the full range of embedding values. Finally, a lambda layer was added to force the encoding to remain on the n-dimensional hypersphere. The model was trained to output embedding. To reduce the loss of train an Adam optimizer with a 0.00006 learning rate was used. The learning rate was selected by trying different values and observing the effect on the loss. The value which gives the best loss was selected. A regularization of L2 (1e-3) was applied to the kernel weights as the regularization function, and a drop out of 0.5 was used to avoid overfitting. Both of the Siamese networks using CNN architecture were trained for each subset.

The classification was done by computing the distance between the query images with the gallery images prepared from the dataset. Then select the class with minimum distance as the most similar class. In this experiment of similarity learning for plankton, the images were evaluated in 3 subsets prepared based on the number of images as shown in Table 3. In each subset experiment, one to six number of references (number of the images from each class in the gallery set) images were selected randomly. The evaluations of each number of references was iterated five times. The MAX, mean, STD and MIN accuracy were computed.

The experiment was repeated with 3 Subsets. Subset 1 consists of 14 classes with 840 query images. Subset 2 consists of 39 classes with 1457 query images. Subset 3 consists of 50 classes with 1574 query images. Each of the pre-trained CNN architectures trained 20,000 iterations of training.

5.4 Results

This subsection shows the results with the two Siamese network-based on VGG19 and VGG16 CNN for each subset. The evaluation was repeated up to five times. Finally, the result of each single class accuracy is computed.

Results using a balanced dataset with more than 1000 images each (Subset 1)

The experiment on Subset 1 was to test the classes with more 1000 images (Subset 1). Tables 4 and 5 show the result using VGG19 and VGG16 CNN architectures for feature extraction respectively. Table 6 shows the summary results for subset 1 using of VGG19 and VGG16 CNN architecture for feature extraction in both Siamese models from top-1 to top-5.

Table 4. The top-1 accuracy on Subset 1 using the Siamese network with VGG19 CNN architecture and the gallery set was selected randomly at each iteration for each number of reference images.

	Number of reference images in the gallery set					
Iteration	**1**	**2**	**3**	**4**	**5**	**6**
1	0.63	0.66	0.71	0.69	0.73	0.74
2	0.59	0.68	0.67	0.70	0.73	0.77
3	0.65	0.69	0.67	0.72	0.73	0.78
4	0.59	0.67	0.70	0.73	0.72	0.75
5	0.61	0.67	0.68	0.70	0.71	0.70
MAX	0.65	0.69	0.71	0.73	0.73	0.78
Mean	0.61	0.67	0.68	0.71	0.72	0.75
STD	0.023	0.011	0.016	0.015	0.008	0.018
MIN	0.59	0.66	0.67	0.69	0.71	0.73

Table 5. The top-1 accuracy on Subset 1 using the Siamese network with VGG16 CNN architecture and the gallery set was selected randomly at each iteration for each number of reference images.

	Number of reference images in the gallery set					
Iteration	**1**	**2**	**3**	**4**	**5**	**6**
1	0.56	0.63	0.64	0.65	0.70	0.73
2	0.45	0.63	0.63	0.64	0.70	0.69
3	0.54	0.61	0.60	0.66	0.66	0.75
4	0.53	0.61	0.72	0.64	0.69	0.70
5	0.54	0.66	0.67	0.70	0.66	0.68
MAX	0.56	0.66	0.72	0.70	0.70	0.75
Mean	0.52	0.63	0.65	0.66	0.68	0.71
STD	0.038	0.018	0.041	0.022	0.018	0.026
MIN	0.45	0.61	0.60	0.64	0.66	0.68

Table 6. The comparison result of top-1 up to top-5 average accuracy on Subset 1 with VGG9 and VGG16.

Base Model	Top-1	Top-2	Top-3	Top-4	Top-5
VGG19	0.75	0.87	0.93	0.96	0.98
VGG16	0.71	0.88	0.92	0.95	0.96

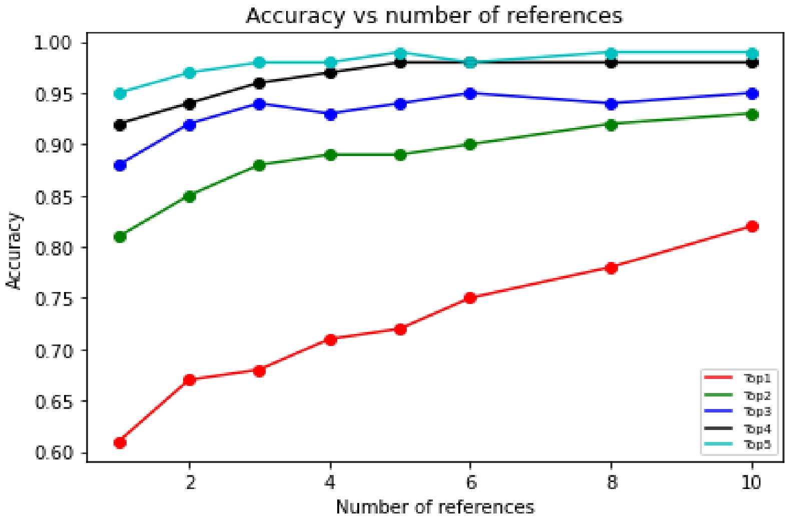


Figure 14. Evaluation of the impact of the number of references in the accuracies of top-1 up to top-5 on Subset 1 in model 1.

The average top-1 accuracy was 75% and 71% for model 1 and model 2 respectively using six references. According to the results from Table 6, both models show the average result of top-1 and quite good result of top-5 accuracy. Figure 14 shows the effect of the number of the reference image in the gallery set in the accuracy for Subset 1. The confusion matrix in Appendix 1 shows the accuracy of each class in Subset 1 using a randomly select gallery set.

Results using an unbalanced dataset with more than 100 images each

Experiment on Subset 2 was in the classes with more than 100 images per class. The results of model 1 and model 2 are shown in Tables 7 and 8 respectively. Table 9 shows the summary result on Subset 2 from top-1 to top-5 accuracy using both models.

Table 7. The top-1 accuracy on Subset 2 using VGG19 and the gallery set was selected randomly at each iteration for each number of reference images.

	Number of reference images in the gallery set					
Iteration	**1**	**2**	**3**	**4**	**5**	**6**
1	0.48	0.55	0.62	0.66	0.68	0.70
2	0.49	0.56	0.60	0.62	0.64	0.69
3	0.45	0.58	0.60	0.62	0.69	0.71
4	0.51	0.61	0.61	0.63	0.62	0.68
5	0.52	0.57	0.63	0.60	0.68	0.69
MAX	0.52	0.61	0.63	0.66	0.69	0.71
Mean	0.49	0.57	0.61	0.62	0.66	0.70
STD	0.02	0.021	0.013	0.02	0.027	0.010
MIN	0.45	0.55	0.60	0.60	0.62	0.68

Table 8. The top-1 accuracy on Subset 2 using VGG16 and the gallery set was selected randomly at each iteration for each number of reference images.

	Number of reference images in the gallery set					
Iteration	**1**	**2**	**3**	**4**	**5**	**6**
1	0.49	0.54	0.60	0.60	0.61	0.68
2	0.46	0.53	0.61	0.56	0.58	0.64
3	0.48	0.56	0.59	0.62	0.64	0.67
4	0.51	0.51	0.55	0.62	0.58	0.60
5	0.53	0.50	0.60	0.59	0.62	0.63
MAX	0.53	0.56	0.61	0.62	0.64	0.68
Mean	0.49	0.53	0.59	0.60	0.61	0.64
STD	0.022	0.016	0.021	0.020	0.023	0.029
MIN	0.46	0.51	0.55	0.56	0.58	0.60

Table 9. The comparison result of top-1 up to top-5 average accuracy on Subset 2 based on VGG19 and VGG16.

Base Model	**Top-1**	**Top-2**	**Top-3**	**Top-4**	**Top-5**
VGG19	0.70	0.88	0.90	0.93	0.94
VGG16	0.64	0.78	0.83	0.87	0.90

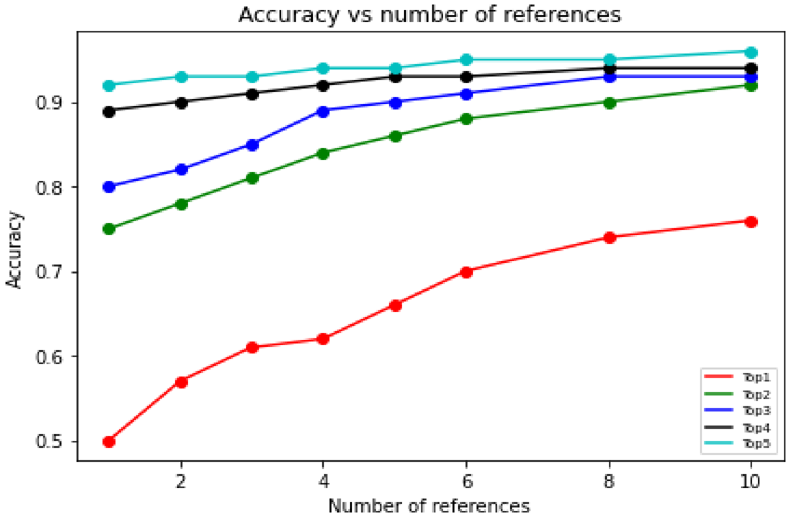


Figure 15. Evaluation of the impact of the number of references in the accuracies of top-1 up to top-5 on Subset 2 in model 1.

The best average top-1 accuracy was 70% and 64% for model 1 and model 2 respectively using six references on Subset 2. According to the results from Table 9, both model results show the average result of the top-1 and quite good result of top-5 accuracies. Figure 15 shows the effect of the number of the reference image in the gallery set in the accuracy on Subset 2. The accuracy of each class in Subset 2, using the randomly select gallery, is shown in Appendix 2.

Results using an unbalanced dataset with more than 10 images each

In the experiment on Subset 3, all classes of the dataset were considered. Tables 10 and 11 show the results using both models. Table 12 shows the summary results on Subset 3 using of both CNN architectures. The final evaluation of similarity learning for plankton images has been experimented with in Subset 3. The accuracy of each class was evaluated using the distance similarity approach.

Table 10. The top-1 accuracy on Subset 3 using VGG19 and the gallery set was selected randomly at each iteration for each number of reference images.

	Number of reference images in the gallery set					
Iteration	**1**	**2**	**3**	**4**	**5**	**6**
1	0.52	0.52	0.59	0.61	0.64	0.65
2	0.41	0.57	0.63	0.63	0.65	0.66
3	0.49	0.53	0.55	0.63	0.63	0.67
4	0.44	0.53	0.51	0.55	0.67	0.60
5	0.51	0.56	0.64	0.62	0.63	0.69
MAX	0.52	0.57	0.64	0.63	0.67	0.69
Mean	0.47	0.54	0.59	0.61	0.64	0.65
STD	0.042	0.019	0.037	0.029	0.015	0.031
MIN	0.41	0.52	0.51	0.55	0.63	0.60

Table 11. The top-1 accuracy of Subset 3 using VGG16 and the gallery set was selected randomly at each iteration for each number of reference images.

	Number of reference images in the gallery set					
Iteration	**1**	**2**	**3**	**4**	**5**	**6**
1	0.49	0.46	0.5	0.54	0.58	0.60
2	0.48	0.48	0.61	0.60	0.56	0.61
3	0.44	0.55	0.57	0.59	0.61	0.62
4	0.38	0.47	0.54	0.57	0.59	0.57
5	0.46	0.55	0.52	0.55	0.56	0.58
MAX	0.49	0.55	0.61	0.60	0.61	0.62
Mean	0.45	0.51	0.55	0.57	0.58	0.60
STD	0.039	0.038	0.035	0.023	0.019	0.018
MIN	0.38	0.46	0.50	0.55	0.56	0.57

Table 12. The comparison result of top-1 up to top-5 average accuracy on Subset 3 based on both models.

Base Model	**Top-1**	**Top-2**	**Top-3**	**Top-4**	**Top-5**
VGG19	0.65	0.82	0.87	0.93	0.94
VGG16	0.60	0.79	0.82	0.86	0.90

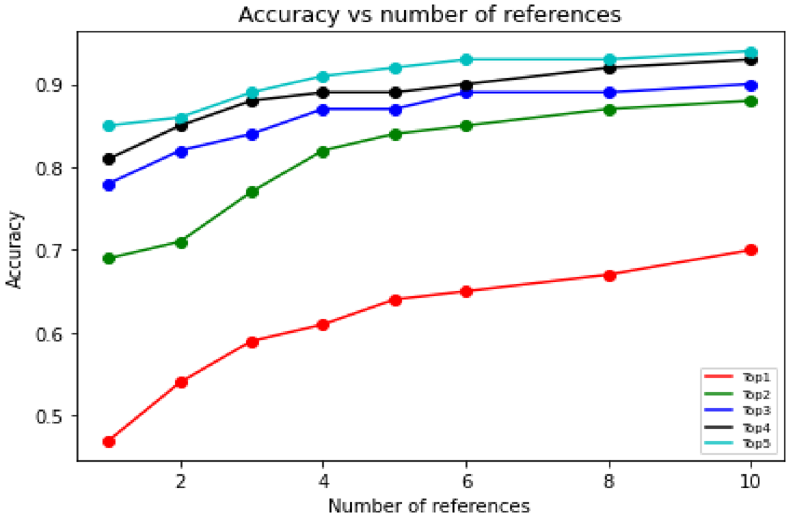


Figure 16. Evaluation of the impact of the number of references in the accuracies of top-1 up to top-5 on Subset 3 in model 1.

Using Subset 3, the best average top-1 accuracy was 65% and 60% for model 1 and model 2 respectively using six reference images. According to the results from Table 12, both models' experiments show the good result of top-1 and quite good result of top-5 accuracy. The confusion matrix in Appendix 3 shows the accuracy of each class in Subset 3 using a randomly select gallery set. Appendix 4 shows the MAX, the mean, the STD, and the MIN accuracies of each class by iterating 6 times using a random select gallery set.

According to the results in Tables 10 and 11, the accuracy in each iteration fluctuated in each randomly selected gallery set. The reason for this is that the images selected for the gallery set has an impact on the classification accuracy. The classification accuracy in both models was increased with an increase in the number of references. However, with the increase of reference gallery images, the computation time also increases. The STD of accuracy was higher when selecting one image per class as a gallery set.

In the experiment of all subsets, the Siamese network using the VGG19 CNN structure achieved higher accuracy. If the selected gallery images represent well to the query images, good accuracy was achieved and low results represent the selected gallery set was not represented well to the query images. The accuracy of each class is highly dependent on the selection of the gallery test set and the number of references. Figures 14, 15 and 16 show the effect of number of references in the accuracy for each subset respectively.

6 DISCUSSION

6.1 Current study

The study focused on CNN-based on similarity learning for plankton images and implementation of classification using similarity comparer. The dataset used for this study is highly unbalanced and some classes are also similar to each other. In the study, three subsets with a minimum of 1000, 100, and 10 images for each class, were created. Fine-tuned VGG19 and VGG16 CNN models were used for building the two Siamese models. These models were trained using the Triplet loss function with shared weights to pass the input images through the same embedding function. To compare two images, the input image must be able to be transformed into a new representation with reduced sizes, such as a vector. To perform that network should be trained to produce good embedding. The image classification was carried out by using a distance similarity between gallery images and the query image. Each model's performance was assessed for each subset. To realize the accuracy and evaluate the effect of the random selection of gallery sets was used and the experiment was repeated five times. Then, average accuracy was computed.

The experiment on Subset 1 (1000 images per class) was to evaluate the plankton class with a large number of images. The subset data used for training and testing was equal for all classes. The best top-1 accuracy obtained with six reference images was 75% and 72% for VGG19 and VGG16 respectively. The best top-5 accuracy was 97% and 98% for both models. The accuracy of classification was improved with a larger number of reference images in the gallery set for each class. According to the confusion matrix in Appendix 1, the dolichospermum anabaenopsis coiled, mesodinium rubrum, snowella woronichinia, cryptophyceae teleaulax, skeletonema marinoi and oscillatoriales were among the highest accuracy classes, whereas images of dinophyceae and dolichospermum anabaenopsis were among the lowest accuracy classes.

The experiment on Subset 2 (more than 100 images per class) was to evaluate the plankton class with an unbalanced dataset using the similarity comparison approach. The top-1 accuracy achieved with six reference images was 71% and 64% for VGG19 and VGG16 respectively. According to the confusion matrix in Appendix 2, images of mesodinium rubrum, dolichospermum anabaenopsis coiled, pauliella taeniata, peridiniella catenata chain and snowella woronichinia were among the highest accuracy classes. The chroococcales and chroococcus small plankton were among the lowest accuracy classes.

The experiment on Subset 3 (more than 10 images per class) was to evaluate all the plankton classes within the dataset. The dataset is an unbalanced number of images ranges from 19 to over 1,000 images(for the classes with more than 1000 samples). As seen in Table 12, VGG19 based model achieved 65% and 94% accuracy for top-1 and top-5 respectively. The results from all subset experiments show that the accuracy of every single class is ranged from 100% to 0%. Most of the plankton classes obtained a good result, but the same plankton classes obtained low classification accuracy on all the 3 subsets. Based on the confusion matrices and accuracy of each class, potentially problematic groups were defined from the dataset. According to Appendices 3 and 4, images of amylax triacantha, cyclotella choctawhatcheeana, melosira arctica, pauliella taeniata, peridiniella catenata chain, cymbomonas tetramitiformis, nodularia spumigena, dolichospermum anabaenopsis coiled and pennales sp. thick plankton classes are among the highest accuracy classes. Images of aphanothece paralleliformis, chroococcales, gonyaulax verior, and uroglenopsis sp. are among the lowest accuracy classes. Figure 16 showed the impact of the number of references on the accuracy of classification. It clearly showed the accuracy is increased with the increase of the number of reference images per class in the gallery set. Based on experiments in 3 different subsets, the unbalanced number of images in the training data has no notable impact on the accuracy of each class, but a good triplet is needed for one-shot learning. The one-shot learning approach is good for the class that does not have enough train data.

6.2 Future work

For the future, it would be interesting to see how much the classification result could be improved with more sophisticated ways to utilize the learning similarity metric, such as K-NN. Applying similarity learning techniques to a variety of plankton datasets and comparing the results to various methods and also investigating alternative deep learning architectures to improve the performance.

7 CONCLUSION

This study aimed to test the Siamese network trained using the Triplet loss function for the classification of plankton. The dataset considered in the study was highly unbalanced. The Siamese models were based on VGG19 and VGG16 CNN architectures for the feature extraction. The triplet's batch was generated using hard triplets at each iteration. Finally, classification was done by comparing the distance between the images from the gallery set with the query image per class.

Various experiments on both models were carried out. In all experiments, the classification accuracy varied based on the selection of random gallery set images. If the selected gallery images represented well the classes a high accuracy was obtained. The accuracy of similarity-based classification is influenced by the number of reference images in the gallery set and the selection of gallery set images. The one-shot learning using the Triplet loss function is good for the class that has few training images. The unbalanced distribution of data does not have a notable effect on classification, but one-shot learning requires a good triplet selection procedure. The best accuracy was achieved with more reference images.

The top-1 accuracy for individual classes varied. Some classes were classified with good accuracy, while some achieved low accuracy. The potentially problematic classes were defined from the dataset. These include gonyaulax-verior, chroococcales, uroglenopsis sp. and aphanothece paralleliformis. Finally, for the future, further investigating of architectures and similarity classifiers is needed to improve the accuracy.

REFERENCES

[1] P. Falkowski. The role of phytoplankton photosynthesis in global biogeochemical cycles. *Photosynthesis Research*, 39:235–258, 1994.

[2] Qiong Li, Xin Sun, Junyu Dong, Shuqun Song, Tongtong Zhang, Dan Liu, Han Zhang, and Shuai Han. Developing a microscopic image dataset in support of intelligent phytoplankton detection using deep learning. *ICES Journal of Marine Science*, 77:1427–1439, 2020.

[3] Carol Lalli and Timothy R Parsons. *Biological oceanography: an introduction*. Elsevier Butterworth-Heinemann, 2rd edition, 1997.

[4] Iain M. Suthers and David Rissik. *Plankton: A guide to their ecology and monitoring for water quality*. CSIRO PUBLISHING, 2009.

[5] K. Bengt, A Felipe, C. Veronique, L. Arnaud, W. Guillaume, and Jukka s. Jerico-next. novel methods for automated in situ observations of phytoplankton diversity. d3.1. *ARCHIMER Ifremer's institutional repository*, 2017.

[6] Osku Grönberg. Plankton recognition from imaging flowcytometer data using convolutional neural networks. Master's thesis, Lappeenranta University of Technology, Finland, 2018.

[7] Cortes C. and Vapnik V. Support-vector networks. *Machine Learning*, 20:273–297, 1995.

[8] Shalev Shwartz S. and Ben David S. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 1st edition, 2014.

[9] Jerome Friedman Trevor Hastie and Robert Tibshirani. *The Elements of Statistical Learning*, volume 1. Springer, 2001.

[10] Mahmut Kaya and Hasan Şakir Bilge. Deep metric learning: A survey. *Symmetry*, 11(9):1066, 2019.

[11] Maram Hamid. Semi-supervised learning for plankton image classification. Master's thesis, Lappeenranta University of Technology, Finland, 2019.

[12] Samsuddin Ahmed, Abol Basher, Abu Naser Rashid Reza, and Ho Yub Jung. A brief overview on deep metric learning. In *Proceedings of Korea Next Generation Computing Society At: Jeju Island, Korea*, June 2018.

[13] Fastvision project. https://www2.it.lut.fi/compvis/fastvision/. Accessed: 2021-1-28.

[14] H. H. Aghdam and E. J. Heravi. *Guide to Convolutional Neural Networks: A Practical Application to Trafc Sign Detection and Classifcation*. Springer International Publishing, 1st edition, 2017.

[15] M. I. Jordan and T. M. Mitchell. Machine learning: Trends, perspectives, and prospects. science. *American Association for the Advancement of Science*, 349(6245):255–260, 2015.

[16] M. A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 1st edition, 2015.

[17] Kunihiko Fukushima. Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4):193–202, 1980.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[19] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 1st edition, 2016.

[20] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of International Conference on Learning Representations (ICLR)*, pages 1409–1556, 2015.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[22] datahacker.rs. Cnn vgg16 and vgg19. http://datahacker.rs/deep-learning-vgg-16-vs-vgg-19/. Accessed: 2021-5-20.

[23] O. Russakovsky, J. Deng, H. Su, J. Krause, and S. et al Satheesh. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2015.

[24] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[25] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.

[26] Rafael C Gonzalez and Richard E Woods. *Digital Image Processing. Prentice-Hall, Inc.* 3rd edition, 2006.

[27] M. Sieracki, M. Benfield, A. Hanson, C. Davis, and C. et al Pilskaln. Optical plankton imaging and analysis systems for ocean observation. In *Procceedings of OceanObs'09: Sustained Ocean Observationsand Information for Society*, pages 21–25, 2009.

[28] Robert J. Olson, Alexi Shalapyonok, Daniel J. Kalb, Steven W. Graves, and Heidi M. Sosik. Imaging flowcytobot modified for high throughput by in-line acoustic focusing of sample particles. *Limnology and Oceanography: Methods*, 15:867–874, 2017.

[29] Robert J Olson and Heidi M Sosik. A submersible imaging-in-flow instrument to analyze nano-and microplankton: Imaging flowcytobot. In *Limnology and Oceanography: Methods*, volume 5, pages 195–203. Association for the Sciences of Limnology and Oceanography (ASLO), 2007.

[30] Ellen J. S., Graff C. A., and Ohman M. D. Improving plankton image classifcation using context metadata. In *Limnology and Oceanography: Methods*, volume 17, page 439–461. Association for the Sciences of Limnology and Oceanography (ASLO), 2019.

[31] G. Bueno, Ó. Déniz, Aníbal Pedraza, Jesús Ruiz-Santaquiteria, J. Salido, G. Cristóbal, María Borrego-Ramos, and S. Blanco. Automated diatom classification (part a): Handcrafted feature approaches. *Applied Sciences*, 7(8):753, 2017.

[32] Qiao Hu and Cabell Davisl. Automatic plankton image recognition with cooccurrence matrices and support vector machine. *Marine Ecology Progress Series*, 295:21–31, 2005.

[33] T. Phan, E. P. Caillault, and A. Bigand. Comparative study on supervised learning methods for identifying phytoplankton species. In *Proceedings of IEEE Sixth International Conference on Communications and Electronics (ICCE)*, pages 283–288, 2016.

[34] I. Correa, P. Drews, S. Botelho, M. S. de Souza, and V. M. Tavano. Deep learning for microalgae classification. In *Proceedings of 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 20–25, 2017.

[35] Tuomas Eerola, Kaisa Kraft, Osku Grönberg, Lasse Lensu, Sanna Suikkanen, Jukka Seppälä, Timo Tamminen, Heikki Kälviäinen, and Heikki Haario. Towards operational phytoplankton recognition with automated high-throughput imaging and compact convolutional neural networks. *Ocean Science Discussions*, 2020.

[36] Yan J., Li X., and Z Cui. A more efcient cnn architecture for plankton classifcation. In *Proceedings of Chinese Conference on Computer Vision (CCCV)*, volume 773, pages 198–208. Springer, 2017.

[37] Jaroslav Bureš. Classification of varying-size plankton images with convolutional neural network. Master's thesis, Brno University of Technology, Czech Republic, 2020.

[38] Jaroslav Bureš, Tuomas Eerola, Lasse Lensu, Heikki Kälviäinen, and Pavel Zemčík. Plankton recognition in imageswith varying size. In *Proceedings of ICPR International Workshops and Challenges, ICPR 2021*, volume 12666, pages 110–120. Springer Lecture Notes in Computer Science, 2021.

[39] Al-Barazanchi H. A., Verma A., and Wang X. S. Intelligent plankton image classifcation with deep learning. *International Journal of Computational Vision and Robotics*, 8(6):561–571, 2018.

[40] L. Xing and Y. Qiao. Deepwriter: A multi-stream deep cnn for text-independent writer identification. In *Proceedings of International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 584–589, 2016.

[41] Nanne van Noord and Eric Postma. Learning scale-variant and scale-invariant features for deep image classification. *Pattern Recognition*, 61:583–592, 2017.

[42] J. S. Ellen, C. A. Graff, and M. D. Ohman. Improving plankton image classifcation using context metadata. *Limnology and Oceanography: Methods*, 17(8):439–461, 2019.

[43] J. Hu, J. Lu, and Y. Tan. Discriminative deep metric learning for face verification in the wild. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1875–1882, 2014.

[44] H. O. Song, S. Jegelka, V. Rathod, and K. Murphy. Deep metric learning via facility location. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2206–2214, 2017.

[45] D. Yi, Z. Lei, S. Liao, and S. Z. Li. Deep metric learning for person re-identification. In *Proceedings of International Conference on Pattern Recognition*, pages 34–39, 2014.

[46] J. Lu, J. Hu, and J. Zhou. Deep metric learning for visual understanding: An overview of recent advances. *IEEE Signal Processing Magazine*, 34(6):76–84, 2017.

[47] R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 1735–1742, 2006.

[48] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 539–546, 2005.

[49] E. Hoffer and N. Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, volume 9370, page 84–92, 2015.

[50] Eric Craeymeersch. One shot learning, siamese networks and triplet loss with keras. https://medium.com/@crimy/one-shot-learning-siamese-networks-and-triplet-loss-with-keras-2885ed022352. Accessed: 2021-2-18.

[51] H. Nguyen and L. Bai. Cosine similarity metric learning for face verification. In *Asian Conference on Computer Vision (ACCV)*, pages 709–750. Springer Berlin Heidelberg, 2011.

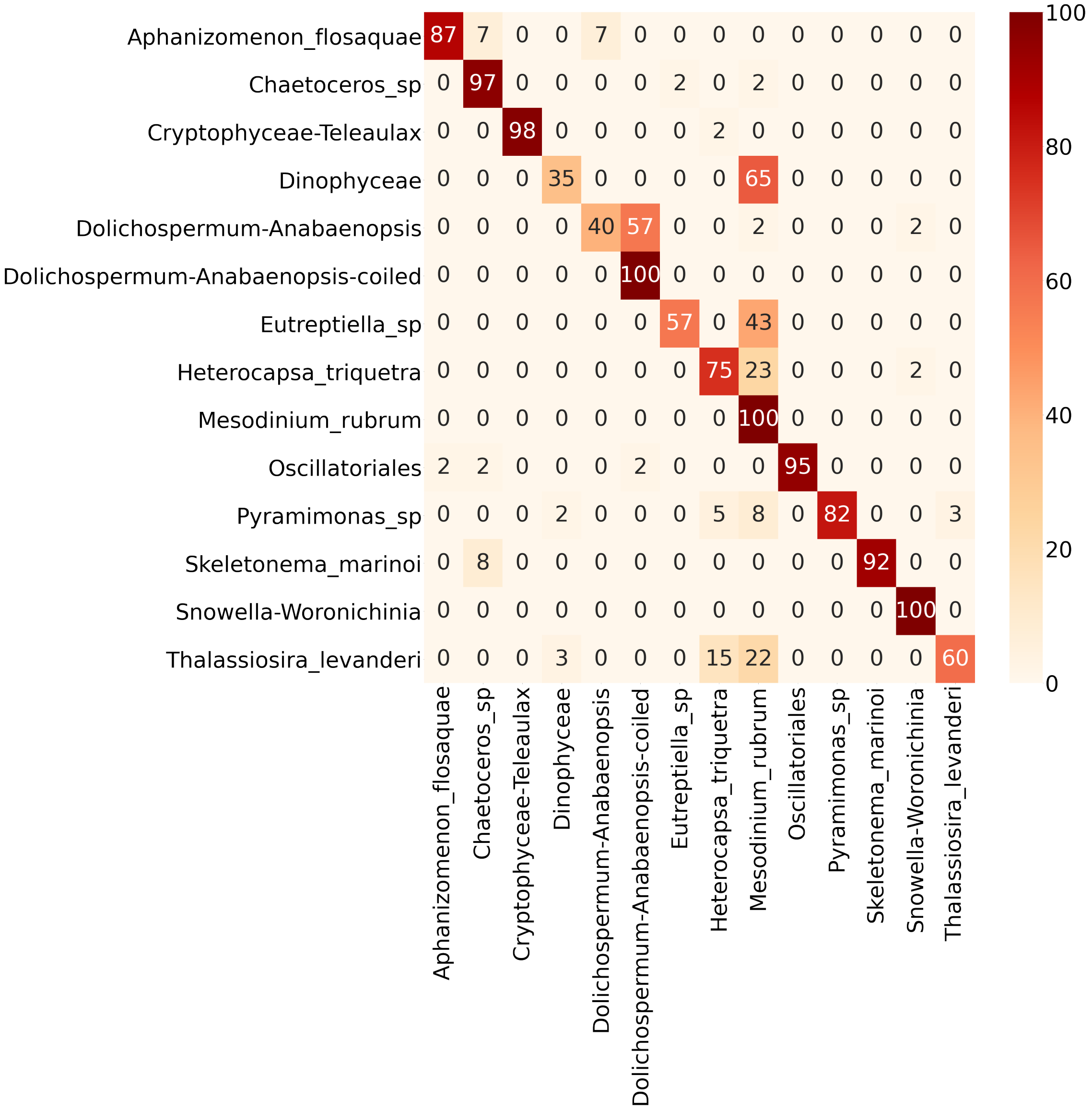
[52] L.K. Weinberger, K.Q.; Saul. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research*, 10(0):207–244, 2009.

[53] P. Paclik, J. Novovicova, and R. P. W. Duin. A trainable similarity measure for image classification. In *Proceedings of 18th International Conference on Pattern Recognition (ICPR)*, volume 3, pages 391–394, 2006.

[54] Finnish environment institute (syke). https://www.syke.fi/en-US. Accessed: 2021-5-25.

Appendix 1. Confusion matrix of the experiment on Subset 1

The top-1 accuracy(%) on Subset 1 using VGG19 CNN architecture based model.

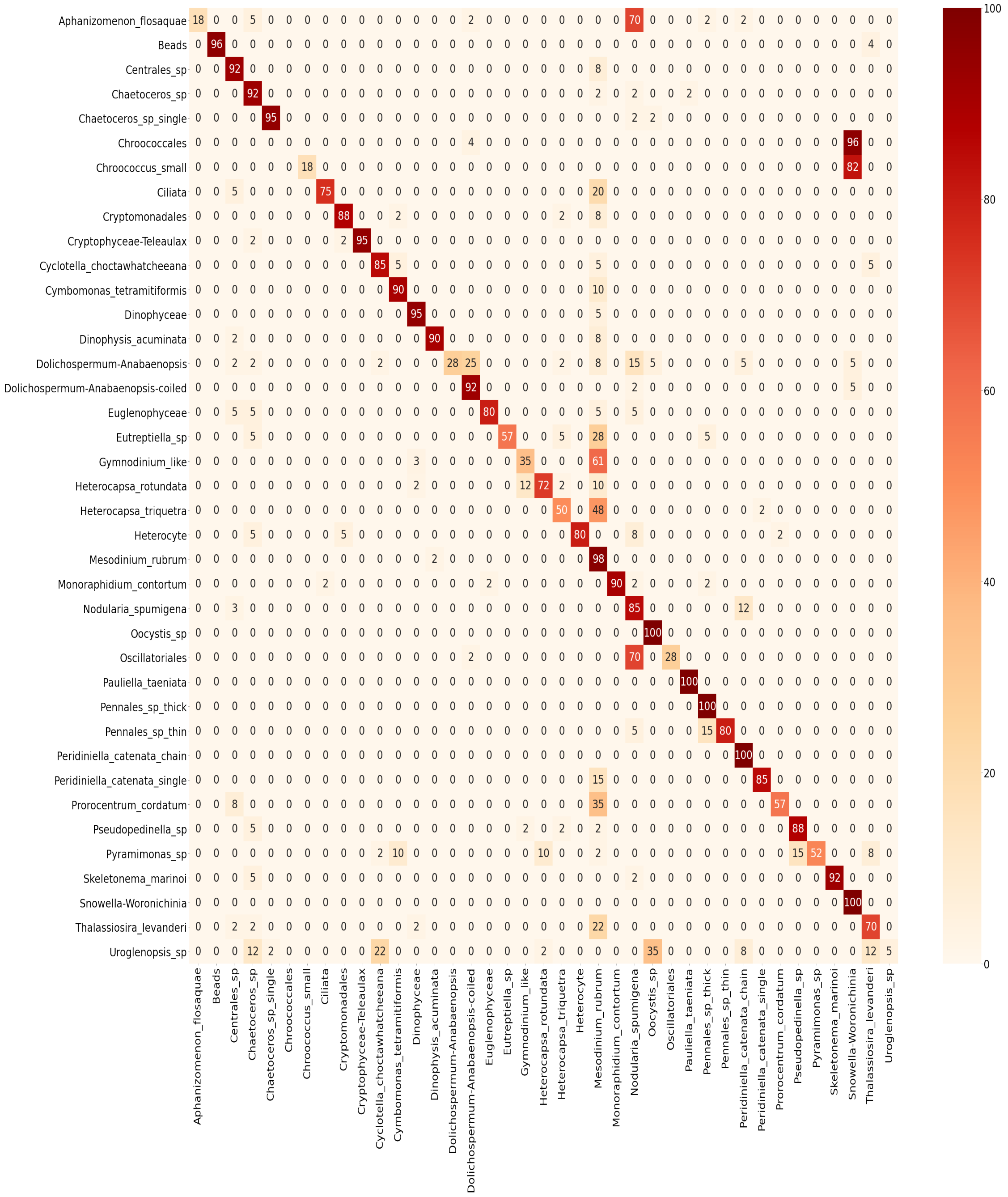


	Aphanizomenon_flosaquae	Chaetoceros_sp	Cryptophyceae-Teleaulax	Dinophyceae	Dolichospermum-Anabaenopsis	Dolichospermum-Anabaenopsis-coiled	Eutreptiella_sp	Heterocapsa_triquetra	Mesodinium_rubrum	Oscillatoriales	Pyramimonas_sp	Skeletonema_marinoi	Snowella-Woronichinia	Thalassiosira_levanderi
Aphanizomenon_flosaquae	87	7	0	0	7	0	0	0	0	0	0	0	0	0
Chaetoceros_sp	0	97	0	0	0	0	2	0	2	0	0	0	0	0
Cryptophyceae-Teleaulax	0	0	98	0	0	0	0	2	0	0	0	0	0	0
Dinophyceae	0	0	0	35	0	0	0	0	65	0	0	0	0	0
Dolichospermum-Anabaenopsis	0	0	0	0	40	57	0	0	2	0	0	0	2	0
Dolichospermum-Anabaenopsis-coiled	0	0	0	0	0	100	0	0	0	0	0	0	0	0
Eutreptiella_sp	0	0	0	0	0	0	57	0	43	0	0	0	0	0
Heterocapsa_triquetra	0	0	0	0	0	0	0	75	23	0	0	0	2	0
Mesodinium_rubrum	0	0	0	0	0	0	0	0	100	0	0	0	0	0
Oscillatoriales	2	2	0	0	0	2	0	0	0	95	0	0	0	0
Pyramimonas_sp	0	0	0	2	0	0	0	5	8	0	82	0	0	3
Skeletonema_marinoi	0	8	0	0	0	0	0	0	0	0	0	92	0	0
Snowella-Woronichinia	0	0	0	0	0	0	0	0	0	0	0	0	100	0
Thalassiosira_levanderi	0	0	0	3	0	0	0	15	22	0	0	0	0	60

(continues)

Appendix 2. Confusion matrix of the experiment on Subset 2

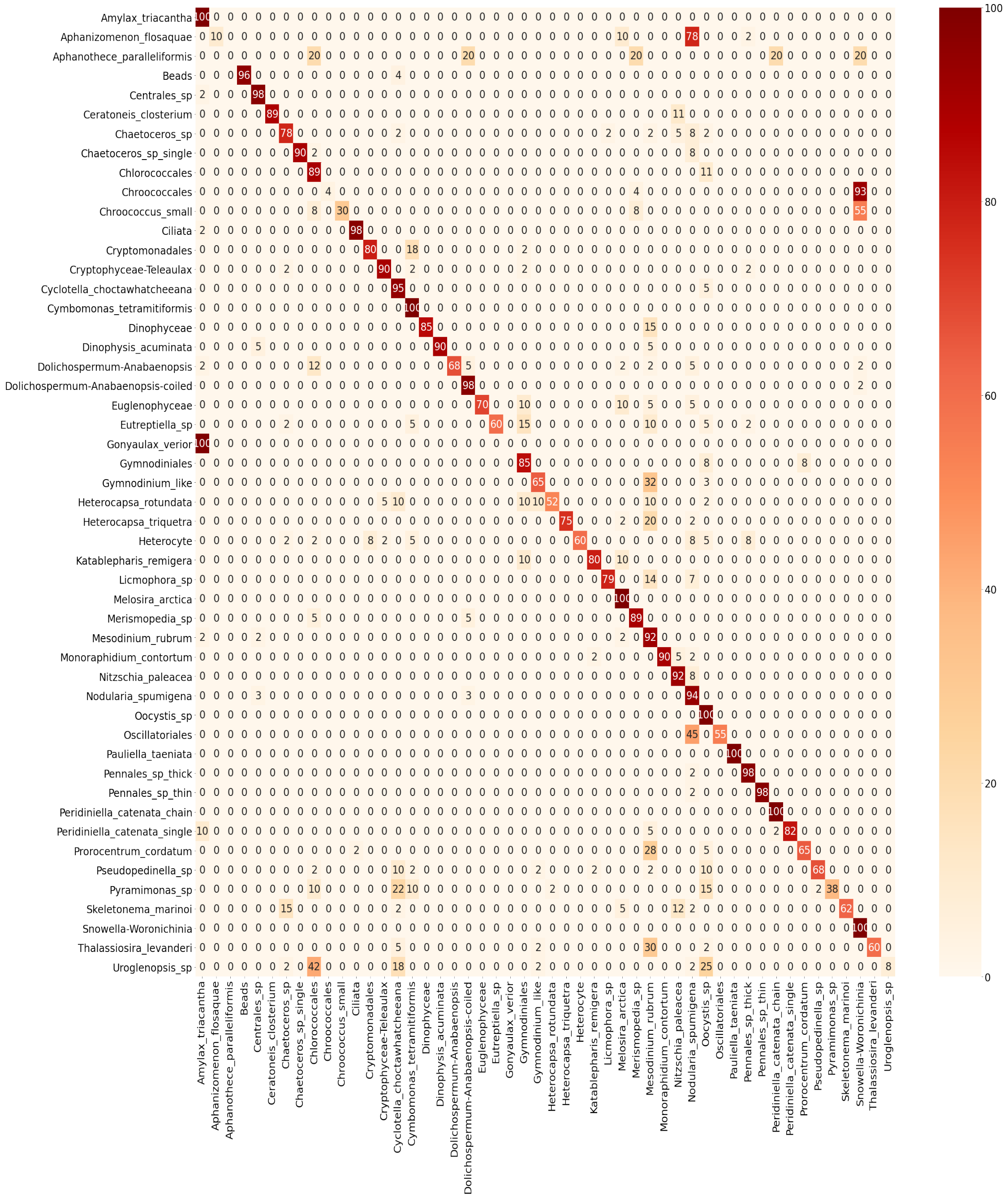
The top-1 accuracy (%) on Subset 2 using VGG19 CNN architecture based model.



(continues)

Appendix 3. Confusion matrix of the experiment on Subset 3

The top-1 accuracy(%) on Subset 3 using VGG19 CNN architecture based model.



Appendix 4. Accuracy of each single class

The top-1 accuracy(%) of each class using VGG19 CNN architecture based model.

Class name	**Samples**	**Accuracy(%) in each Iteration**						**MAX**	**Mean**	**STD**	**MIN**
Amylax triacantha	3	100	100	100	100	100	100	100	100	0	100
Cyclotella choctawhatcheeana	20	100	100	100	100	100	100	100	100	0	100
Melosira arctica	8	100	100	100	100	100	100	100	100	0	100
Pauliella taeniata	23	100	100	100	100	100	100	100	100	0	100
Peridiniella catenata chain	38	100	100	100	100	100	100	100	100	0	100
Snowella Woronichinia	40.0	95.0	100	97.5	100	100	100	100	98.8	1.91	95.0
Mesodinium rubrum	40	95.0	97.5	90.0	100	97.5	95.0	100	95.8	3.12	90.0
Pennales sp. thin	40	95.0	90.0	97.5	100	100	90.0	100	95.4	4.19	90.0
Oocystis sp.	40	92.5	97.5	95.0	90.0	95.0	95.0	97.5	94.2	2.36	90.0
Cymbomonas tetramitiformis	39	92.3	89.7	94.9	94.9	94.9	97.4	97.4	94	2.42	89.7
Nodularia spumigena	33	97.0	94.0	94.0	94.0	91.0	91.0	97.0	93.4	2.08	90.9
Dolichospermum Anabaenopsis coiled	40	90.0	92.5	90.0	90.0	95.0	92.5	95.0	91.7	1.86	90.0
Pennales sp. thick	40	97.5	85.0	92.5	97.5	87.5	87.5	97.5	91.3	4.95	85.0
Peridiniella catenata single	40	85.0	95.0	80.0	87.5	92.5	87.5	95.0	87.9	4.88	80.0
Beads	25	84.0	88.0	84.0	84.0	88.0	88.0	88.0	86.0	2.00	84.0
Nitzschia paleacea	13	61.5	92.3	92.3	92.3	92.3	76.9	92.3	84.6	11.8	61.5
Dinophysis acuminata	40	82.5	82.5	87.5	90.0	82.5	82.5	90.0	84.6	3.03	82.5
Gymnodiniales	13	76.9	92.3	76.9	76.9	84.6	84.6	92.3	82.1	5.7	76.9
Chaetoceros sp. single	40	82.5	72.5	82.5	85.0	85.0	82.5	85.0	81.7	4.3	72.5
Monoraphidium contortum	40	80.0	80.0	82.5	82.5	80.0	85.0	85.0	81.7	1.86	80.0
Skeletonema marinoi	40	72.5	72.5	80.0	90.0	90.0	82.5	90.0	81.3	7.21	72.5
Cryptophyceae teleaulax	40	90.0	80.0	75.0	65.0	80.0	75.0	90.0	77.5	7.51	65.0
Ceratoneis closterium	9	66.7	88.9	66.7	77.8	88.9	66.7	88.9	75.9	9.97	66.7
Cryptomonadales	40	72.5	80.0	72.5	77.5	80.0	72.5	80.0	75.8	3.44	72.5
Chaetoceros sp.	40	72.5	80.0	60.0	85.0	72.5	77.5	85.0	74.6	7.83	60.0
Licmophora sp.	14	64.3	71.4	64.3	78.6	71.4	71.4	78.6	70.2	4.91	64.3
Pseudopedinella sp.	40	70.0	77.5	60.0	57.5	75	77.5	77.5	69.58	8.09	57.5
Gymnodinium like	31	64.5	58.1	67.7	71.0	74.2	71.0	74.2	67.7	5.31	58.1
Chlorococcales	19	78.9	52.6	78.9	68.4	73.9	47.4	78.9	66.7	12.4	47.4
Eutreptiella sp.	40	70.0	70.0	60.0	67.5	60.0	22.5	70.0	58.3	16.6	22.5
Dolichospermum anabaenopsis	40	57.5	57.5	55.0	45.0	57.5	62.5	62.5	55.8	5.34	45.0
Oscillatoriales	40	57.5	40.0	47.5	72.5	65.0	52.5	72.5	55.8	10.7	40.0
Katablepharis remigera	10	60.0	60.0	40.0	60.0	60.0	50.0	60.0	55.0	7.64	40.0
Centrales sp.	40	60.0	52.5	62.5	35.0	50.0	62.5	62.5	53.8	9.66	35.0
Euglenophyceae	20	45.0	75.0	50.0	50.0	5.00	50.0	75.0	53.33	9.86	45.0
Thalassiosira levanderi	40	55.0	45.0	37.5	65.0	42.5	37.5	65.0	47.1	9.94	37.5
Ciliata	40	30.0	35.0	52.5	37.5	32.5	62.5	62.5	41.7	11.8	30.0
Heterocapsa triquetra	40	40.0	27.5	40.0	22.5	55.0	60.0	60.0	40.8	13.4	22.5
Dinophyceae	40	47.5	42.5	47.5	10.0	15.0	75.0	75.0	39.6	21.9	10.0
Heterocapsa rotundata	40	20.0	35.0	25.0	45.0	42.5	25.0	45.0	32.1	9.40	20.0
Heterocyte	40	47.5	32.5	10.0	17.5	30.0	42.5	47.5	30.0	13.1	10.0
Aphanizomenon flosaquae	40	57.5	27.5	10.0	37.5	15.0	27.5	57.5	29.2	15.5	10.0
Merismopedia sp.	19	21.1	42.1	31.6	31.6	26.3	10.5	42.1	27.2	9.81	10.5
Chroococcus small	40	17.5	15.0	45.0	17.5	10.0	10.0	45.0	19.2	11.9	10.0
Prorocentrum cordatum	40	37.5	22.5	02.5	10.0	30.0	05.0	37.5	17.9	13.1	02.5
Pyramimonas sp.	40	12.5	02.5	12.5	05.0	15.0	05.0	15.0	8.75	4.73	02.5
Gonyaulax verior	4	00.0	25.0	00.0	25.0	00.0	00.0	25.0	8.33	11.8	00.0
Chroococcales	28	7.14	3.57	00.0	7.14	3.57	00.0	7.14	3.57	2.91	00.0
Uroglenopsis sp.	40	00.0	00.0	00.0	10.0	02.5	02.5	10.0	02.5	3.54	00.0
Aphanothece paralleliformis	5	00.0	00.0	00.0	00.0	00.0	00.0	00.0	00.0	00.0	00.0