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

Maram Hamid

SEMI-SUPERVISED LEARNING FOR PLANKTON IMAGE CLASSIFICATION

Master's Thesis

Examiners: Professor Heikki Kälviäinen
D.Sc. Tuomas Eerola

Supervisors: D.Sc. Tuomas Eerola
Professor Lasse Lensu
Professor Heikki Kälviäinen

ABSTRACT

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

Maram Hamid

Semi-supervised learning for plankton image classification

Master's Thesis

2019

57 pages, 20 figures, 9 tables, 0 appendices.

Examiners: Professor Heikki Kälviäinen
D.Sc. Tuomas Eerola

Keywords: Plankton, classification, semi-supervised learning, plankton imaging, convolutional neural network, label propagation, label spreading, pseudo labeling

The production of big data from plankton populations has become feasible with the use of imaging devices. This opens up the possibility of testing key characteristics in planktonic systems. The manual labeling of images is complicated as well as time-consuming. On the other hand, semi-supervised learning makes it possible to automate the process by performing the classification of large unlabeled data with the available labeled data. The classification of plankton images using semi-supervised methods including label propagation, label spreading, and Pseudo labeling utilizing CNN based image features has been implemented in this study. The comparison of the accuracy of the semi-supervised methods and the CNN method using different datasets shows that the semi-supervised learning methods were more accurate than the baseline method where the amount of data available to train the model was sufficient and fully supervised CNN was able to extract the correct features. However, sometimes the semi-supervised learning methods have not been more accurate than CNN, when CNN has struggled to find the right image features.

PREFACE

First of all, I thank Allah for giving me the insight and strength to finish this study. In fact, I could not have achieved my current level of success without strong support, assistance, and guidance from the supervises D.Sc. Tuomas Eerola, Professor Lasse Lensu, and Professor Heikki Kälviäinen during the research process. And secondly, my parents, who supported me with love and understanding. Thank you all for your unwavering support.

Lappeenranta, June 20, 2020

Maram Hamid

CONTENTS

LIST OF ABBREVIATIONS

CEAL	Cost-effective active learning
CNNs	Convolutional neural networks
DT	Decision Trees
DTs	Decision Trees kernels
GRF	Guided Random Forest
GRRF	Guided Regularized Random Forest
K-NN	K-Nearest Neighbors
LP	Label Propagation
LS	Label Spreading
MAP	Maximum a Posteriori
MLP	Multi-Layer Perception
MNIST	Standards and Technology database
RBF	Radial Basis Function
RF	Random Forest
RRF	Regularized Random Forest
SVM	Support Vector Machine
VPR	Video Plankton Recorder

1 INTRODUCTION

1.1 Background

Planktons are a varied assortment of aqueous micro-organisms characterized typically by their microscopic size and incapability of withstanding water streams. Planktons are small water-floating organisms which are the core of marine ecosystems. Planktons can be categorized into approximately three sub-collections based on the trophic type [1]: producers, decomposers, and consumers. Phytoplankton, a class of planktons belongs to the producers hence goes through the process of photosynthesis, Zooplanktons are plankton consumers who feed other plankton, and Bacterioplankton are nutrient decomposers.

Phytoplanktons triggers the marine food chain and are essentially the primary producers in the ocean and environmental primary producers in the world [2]. Phytoplanktons often contribute to the biogeochemical process of biogenic resources including carbon, nitrogen, and phosphorus. They also play an important role in aquaculture, ecological, and form the foundation for the food web that connects the atmosphere with the deep ocean. However, marine phytoplanktons cause several dangerous environmental anomalies including red tides and canola, especially in coastal places such as China. This leads to infinite oxygen deprivation and the eventual destruction of hydrous cultures. Concurrently, phytoplanktons are considered to be a major donor of the oxygen to the atmosphere layer of the Earth [3].

Planktons require comprehensive and continuous observations to grasp aspects of physical and biological activity that govern their composition. Recent technological developments have enabled the creation of automated plankton imaging tools to capture a huge number of images (up to ten thousand) per hour. A submarine flow cytometer also called Flow-Cytobot [4] is used to capture the ocular characteristics of the individual plankton cells as they move in laser rays. This technique facilitates unceasing long-term observations for the construction of the plankton population images. It also provides a huge amount of details of the cellular shapes and dimensions that help in getting a closer insight into the mechanism and interactions of plankton life-cycle regulation [5]. Through this, sufficient images from such systems can be captured to provide a unique opportunity to design and understand plankton ecosystems in a sufficient time frame [6]. Figure 1 shows a collection of several phytoplankton images [7].

Previously, phytoplankton ecosystem research required scientists to observe and calculate manually by microscopic device, the large portion of which relates to non-in-situ (moved from its original place) evaluation techniques. The manual classifications of such a large set of images are considered to be inefficient. Moreover, the annotation role is challenging due to the size of the applicable classes, the difference and the similarity between the classes. This conventional method is time-consuming and requires a lot of workers with a high level of professional experience. Furthermore, it is very difficult to classify plankton within an appropriate duration which challenge arises from the variation in species distribution of plankton [2].

Automatic classification is one of the challenges in machine learning algorithms that trains the classifier to assign input data to the unlabeled output classes. There are varieties of techniques for classifying images, such as decision tree [8], random forest [9], and neural network [10]. Plankton datasets include a limited number of classified images and numerous images that have not been labeled. This brings a possibility of using datasets to enhance the classification of minimal or small labeled images data [11].

The supervised method means that sample data are provided as labeled data are given to the learning classifier. The other approach is unsupervised learning where sample data are provided as observations without a label. Unsupervised learning uses techniques to try to find the patterns in the datasets. This method does not have external information for identifying the sample pattern, and it is the sole responsibility of the learning classifier to identify the important details. Training datasets are fully labeled in supervised learning and none of the training datasets are labeled in unsupervised learning. Semi-supervised learning is the technique used to classify the dataset that contains labeled and unlabeled data which can offer excellent classification output through the use of unlabeled data [12].

The semi-supervised methods intended to train from both unlabeled and labeled datasets, usually assuming that they are sampled from certain or related distributions. Such methods depend on the information to be obtained from the unlabeled datasets [11]. The plankton images system faces real problems in collecting plankton data across hundreds of micro-organisms. This method is very useful when most of the dataset is not labeled.

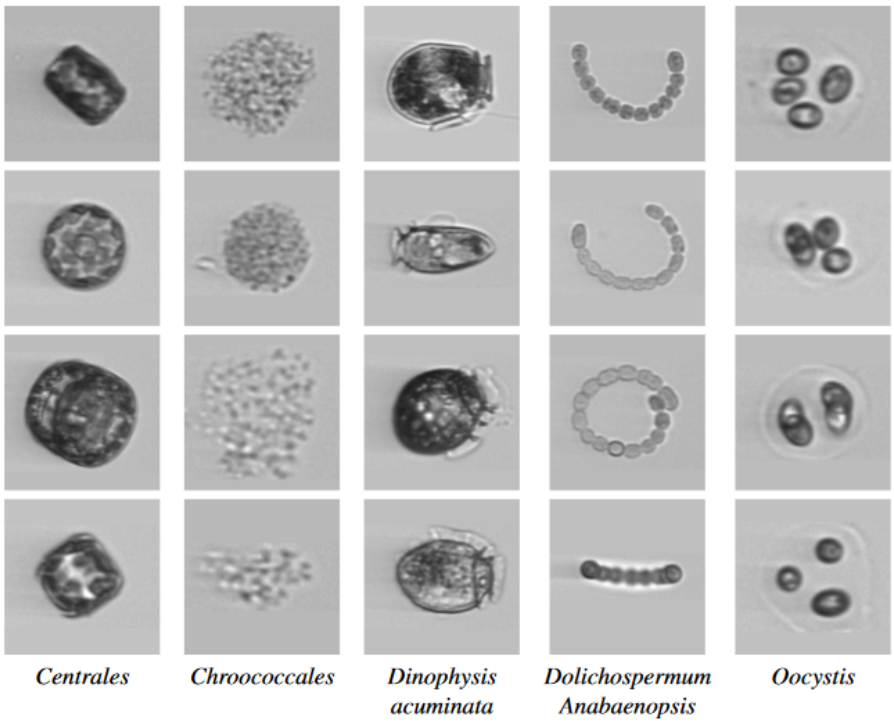

Figure 1. Examples of the resized phytoplankton images [7].

1.2 Objectives and delimitations

The aims of this research are as follows:

1. To implement and to train semi-supervised image classification methods for plankton recognition.
2. To assess the performance of the implemented methods concerning the number of labeled training images per class, as well as, their ability to handle large class imbalance.
3. To compare the classification accuracy of semi-supervised approaches with the classification approach trained using labeled images only.

The main limitations of this research are as follows:

1. Only phytoplankton species found in the Baltic Sea were considered.
2. Images were captured using the same device called Imaging FlowCytobot.

This research focuses on plankton images but, however, the classification methods used in this research can be utilized for any datasets with the same properties.

1.3 Structure of the thesis

Chapter 2 provides an overview of plankton imaging, plankton recognition, the automated methods for recognizing plankton. Chapter 3 discusses semi-supervised learning principles and semi-supervised deep learning concepts. Chapter 4 includes a pipeline for semi-supervised methods, supervised CNN for feature extraction, semi-supervised methods of learning such as label propagation, label spreading, and Pseudo labeling. Chapter 5 describes the data, the pre-processing steps used, the experiments with these methods, and the results of those experiments. Chapter 6 addresses the results, and outlines potential future works. Finally, the conclusions are given in Chapter 7.

2 PLANKTON RECOGNITION

2.1 Plankton imaging

The development of plankton-imaging technology and progress in the timely extraction of helpful information from the image dataset brings increased opportunities for aquatic biota research. Usually, microscopic analysis of stored dataset contains subsampling, numbering, and filtering into many taxonomic classes. A standardized optical micrometer is also used to assess and to provide taxonomic information of the plankton. These operations take time and result in a long delay in collecting, analyzing, and interpreting the image dataset. The analysis requires a human expert who can often discern morphological characteristics. Furthermore, increasing the speed of the process can lead to exhaustion and increased error rates [13].

Silhouette photography [14], considered as an original study of the design of a constant history of plankton, the data from this device could be processed utilizing a microscope or a computer-aided device. Silhouette imaging has lately been updated by the use of a flatbed scanner to build a digital images silhouettes [15]. A Video Plankton Recorder (VPR) was the first plankton-imaging tool used for capturing images in aquatic biota research. It was the former plankton detection that automatically recognizes and defines the plankton taxonomic group, which is to classify plankton within a larger system based on its similarities and variation. It maps a very large amount of plankton and spread their behaviors quantitatively in time with high-resolution [16]. Many technologies have been created using holographic imaging which captures perceptible images of the constituents providing unique details that cannot be obtained by traditional systems. They also allow imaging technologies to gather information without touching the target plankton [17].

Imaging flow cytometry is another technique used for constructing plankton images [5]. There are many flow cytometry applications to capture plankton images among which are the Imaging FlowCytobot [4] and FlowCAM [18]. It is now possible to generate measurements that can both steadily and selectively track, quantify and record the finest plankton organisms in the aquatic habitat. Flow cytometer (FlowCytobot) tracks the optical characteristics of actual dandling plankton the moment they cross laser beam. Imaging FlowCytobot can be performed unattended to for two months, track results and adjust sampling processes if necessary. Such integrated approaches allow for sustained long-term observations of the nature of the plankton population over a wide spectrum of cell types and sizes. They help to clarify the mechanism and relationships that govern the individual

species throughout their life cycle.

Another technique is ZooCAM [19] which is suitable for the measurement of a micro-organism longer than 300 micrometers. The ZooCAM performance is better for a large population of organisms and practical community studies than microscopes or other imaging systems. ZooCAM is used for unperceived datasets, an enormous amount of data, an automatic filtering process, and is particularly effective for fish eggs. ZooCAM could deliver real-time results of plankton populations for evolutionary sampling or targeted ecosystem studies.

2.2 Automated plankton recognition methods

2.2.1 Automatic plankton recognition

Plankton organisms are very important to aquatic ecosystems. Scientists use imaging-focused techniques to investigate certain creatures in their normal behavior. Images from such kind of systems offer a rare chance to model plankton ecosystems and understand them better. The dataset from these systems can be huge making it difficult to manually classify. Also, the annotation process is difficult due to the wide range of related classes, internal class variation, and similarity between classes. Automated plankton recognition is a computer algorithm that uses machine learning techniques to recognize a plankton type based on a database rather than a professional human expert. Various types of automated methods are available, but often their accuracy is lower than that of professional human experts [6].

2.2.2 Plankton recognition methods utilizing handcrafted features

This subsection discusses briefly the common automated classification methods of plankton organisms. The handcrafted features indicate the features (such as size and shape) derived from the different methods used to discover the knowledge from the image itself [20]. In hand-crafted features, the feature extraction process is manually designed with the aim that the features extracted should have strong differences in the objects, the features are particularly successful in case of a small amount of sample in datasets. Though hand-crafted methods have some inherent disadvantages. For example, the extracted features sometimes have very weak features, so they are not suitable for classifying

the object [21].

The prominent machine learning algorithms with handcrafted feature extraction are briefly explained next. Those methods are mostly used for image recognition with a small dataset, this way the features can be extracted from the images insufficient time.

A Support Vector Machine (SVM) [22] is a supervised machine learning algorithm, commonly used in classification problems. SVM is based on the concept of finding a hyperplane that separates a dataset into two groups. The dataset that is far from the hyperplane is more confident and has been correctly classified. To selecting the right hyperplane, the distance between the hyperplane and the closest point in the training set (margin) must be large. Choosing the right hyperplane offers a higher probability of proper classification of new data [23]. Figure 2 shows the concept of SVM method [24].

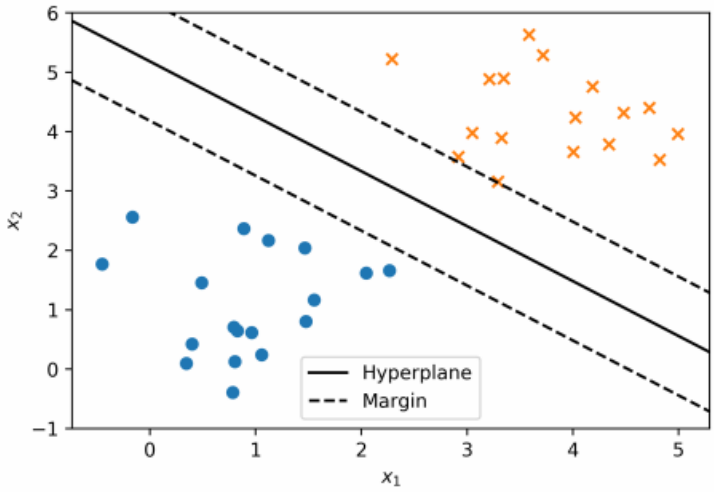

Figure 2. Support Vector Machine method, the continuous line is hyperplane and dash lines are the margin [24]

Multilayer Perceptron (MLP) [25] is a neural network containing more than one perceptron. The perceptron consisting of the input layer, the hidden layer, and the output layer. The MLP train on the collection of input and output data, and the model learns the dependencies between these inputs and outputs. MLP uses two parameters (weights and bias) to reduce the error. Backpropagation is used to make these weight and bias improvements by error, this error can be calculated in different ways, such as root mean squared error (RMSE). MLP has forward and backward movements. In the forward, the information flows from the input layer to the output layer, as well as the output layer prediction is evaluated against the ground truth labels. The backward use back-propagation and derivatives of the error function, this differentiation gives us a gradient. Figure 3

shows the concept of MLP method [26].

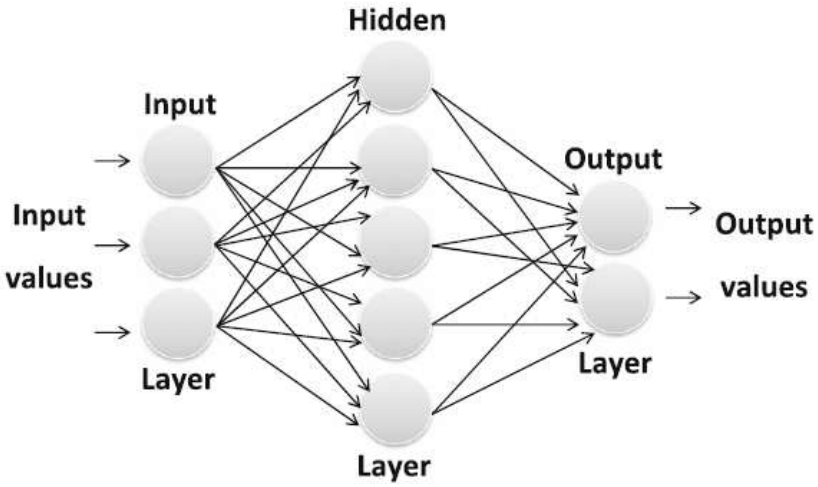

Figure 3. Multilayer Perceptron method architecture [26].

The decision tree (DT) [27] is a classification model built in the shape of a tree structure where the internal node is a feature, the branch is a decision rule, and the leaf is the result. The DT classifies the samples by ordering them from the root to the leaf where the node of the leaf produces the result of classification. The general steps to classify the sample using DT are to select the best feature, create a decision node, break the dataset to smaller subsets, and then start building the tree by repeating these steps until all tuples conform to the same attribute value.

Random Forest (RF) [28] is a supervised learning approach. The Forest constructs a collection of the DTs, usually trained using a baggage approach. Bagging approach means that the overall result is increased by a combination of learning models. The RF consists of thousands of DT trains by different datasets, with a tiny difference in observations, and then divides the nodes in each tree, taking into consideration a limited number of features. The result of the random forest is the mean of each tree's predictions. RF can be utilized to assess which of the features are most useful for classification [29].

A regularized random forest (RRF) is one method of regulated tree proposed for feature selection by constructing a single set. In RRF, the features are assessed on the data training part of each 3 node. The RRF needs to apply the tree regularization structure to RF and can pick a compact feature. RRF is not sensitive to outliers, so they need a few pre-processing, such as normalization. The Guided Regularized Random Forest (GRF) is an enhanced RRF that is driven by the significant results from normal random forest [30].

K Nearest Neighbors (K-NN) [31] is a simple method of classification that classifies the data depending on the measure of similarity. K-NN select K value, which is any positive number, locate k data points nearest to the unidentified data point depending on the distance, vote for the most repeated label, and then allocate the unknown data to the class with the maximum number of neighbors. K-NN is conceptually simple and has the advantage of being simple and needs a few parameters (distance metric and k).

Cabell S Davis et al. [16] and [32] implemented the plankton classification method with images taken by a device called the plankton video recorder. The SVM algorithm applied had an accuracy of 71 % for the classification to seven classes. In [33], the accuracy of six methods MLP classifiers, K-NN with k equal to 5, SVM and RBF, DT, and RF was evaluated with classes from 5 to 35. The number of digital images for individual objects has balanced classes.

Using ZooScan for automated investigation of plankton images, Kramer et al. [34] evaluated individual classification methods and also combined different classification methods. The combination includes K-NN with vector forest, and especially a mixture of linear discriminating analysis (a method for finding a linear subset of features that divides two or more classes) with learning vector quantization and RF. Vector quantization is a traditional signal processing quantization technique that permits the design of probability density function using the paradigm vector distribution [35]. The combination of linear discriminant with random forest and learning vector quantization achieves an accuracy of 75 % for 29 plankton organisms. In the study the classification of plankton images, classifiers like SVM, RF, DT, and the neural network of sequence correlations were evaluated. The SVM has the best classification efficiency for 6 and 7 classes, with 90 % and 75 % respectively.

In [36], a quantitative analysis of the results of the classification of phytoplankton organisms by six proposed classification techniques was examined. The techniques were Guided Random Forest (GRF), Regularized Random Forest (RRF), Guided Regularized Random Forest, SVM, RF, and K-NN, and each classifier was assessed with four-fold cross-validation. The outcome obtained indicated that RF was the most robust in the classification of phytoplankton organisms. Andrhé Bigand et al. [36], offered a quantitative comparison result of the different classification approaches used in different types of applications. In comparison, the same classifiers were applied using different types of features for the same target, and the different classifiers applied to the same features. In particular, the researchers analyzed the extracted features and determined the attribute that affected the focus variable [36].

2.2.3 Convolutional neural networks for image classification

Convolutional neural networks (CNNs) are modern algorithms used to classify images. The model classifies the data by taking the input (images) and then train the model and the output (class) is output. CNN uses deep learning techniques to derive features from data, unlike traditional machine learning techniques that use handcrafted features [37]. Typical CNN involves layers of different types that convert the input (images) to output (classes). The CNN architecture typically starts with a combination of convolutional layers, pooling layers, and several fully connected layers. The human vision system inspired CNN, and different kinds of layers could be seen as the same as several neurons. [7]. Figure 4 shows an example of the CNN architecture [38].

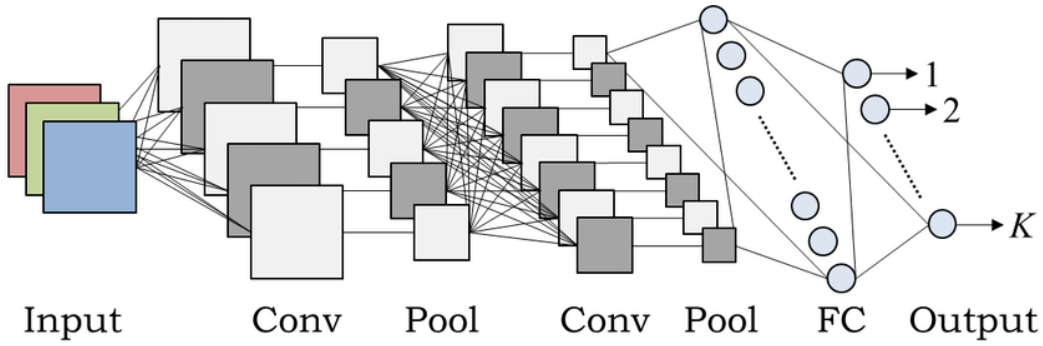

Figure 4. Architecture for convolutional neural networks consists of convolution, pooling, and fully connected layers [38].

The parameters of the convolutional layer are defined as a series of learnable filters. Moving forward, every single filter slide through the height and width of the input volume and calculates the dot products at every point between the inputs and entries of the filter. The two-dimensional activation map which provides the filter's response at each spatial point is generated. Logically, the network can learn how to excite the filters when it sees some kind of visual feature, such as an edge, a color block, or a more complex structure on higher layers of the network [39].

In the CNN architecture, a pooling layer is commonly used for downsampling feature maps from a convolutional layer. This leads to shrinking of the network size to remain just the most beneficial details and makes the network more invariant in shifts [40]. Pooling can be classified across two common types: average pooling and max pooling. The max-pooling layers could perform better than the average pooling layers, since the most significant features can be considered locally by the largest local activation of the convolutional layers.

Fully connected layers are conventional neural networks containing all neurons fully connecting to each other and receiving input value from the activation of the previous layer. It performs the classification task based on the features of the images by complete linkage to the last layers of CNN. Connections within a fully-connected layer include learnable weights and the trainable biases in the output nodes. In a fully connected layer, the output value of an output node is the weighted sum of the bias and its inputs [41].

In CNN, the deep feature extraction technique is supreme to the classical feature extraction technique. The automated deep-feature extraction process is capable of determining useful potentially features for specific classification cases. For other classification approaches, CNN convolutional layers could be used to extract features from the image dataset. The idea behind the convolutional layers is to execute the feature extraction process to give the map of the input feature [42]. Figure 5 shows the feature extraction process using CNNs [43].

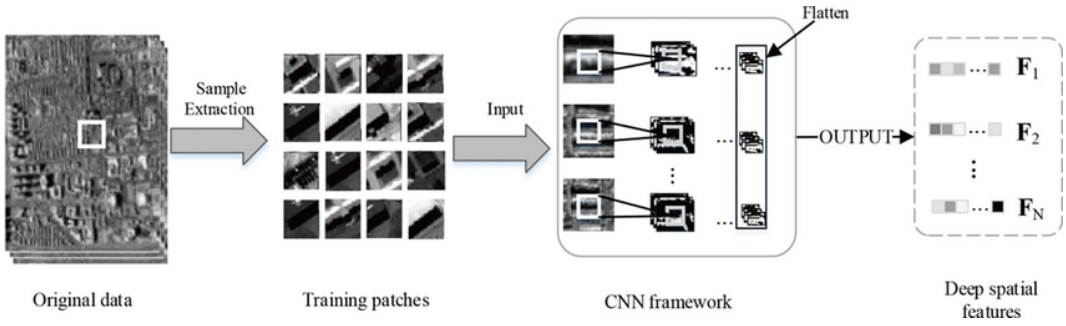

Figure 5. Feature extraction process using CNN architecture [43].

Image classification is among the most common use of CNN. CNN training can be done by sequentially calculating the difference between the target and the expected output of the network (bias) is computed using the loss function, then modify the CNN parameters (weights) to minimize this bias. Realistically in classification, applying loss function means to reduce the divergence that comes from measuring how specific the values of probability vary from other values. [44]. Figure 6 shows a full graph of the CNN architecture for the process of features extraction and classification to find the class prediction. The feature extraction passed the outcome through a batch normalization function, a pooling function and an activation function [45].

The gradient descent algorithm the gradient descent algorithm can be defined as the foundation method to decrease the minimum value of the differentiable functions. Backpropagation is a technique for effective calculation of the gradient's partial derivative for network biases and weights in terms of a loss function [46].

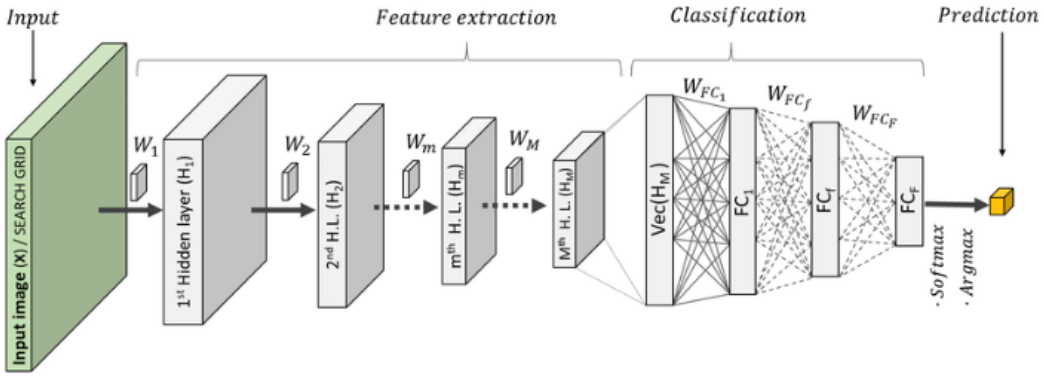

Figure 6. A full graph of the CNN architecture for the process of features extraction and classification, W_M are filters for features extraction process, W_{FC_F} are weight matrices for classification process, hidden layers are H_M [45].

2.2.4 convolutional neural networks based methods for plankton recognition

This subsection introduces classification studies of plankton images using convolutional neural networks (CNN).

In [47], the CNN-based image classification was used to classify the plankton dataset by conducting three experiments. The dataset in this study consisted of 160736 images and 121 different classes. The dataset was passed through the convolutional layers, sampled and the size of the data was reduced. The extracted features map was small, so there was no need to add a subsampling layer when the features were entered in the MLP classifier. The first experiment compared the simple machine learning model like the MLP model to the CNN model, and the CNN method proved to be appropriate for the plankton dataset with an accuracy of 77.36 %. The second study tested how many layers are suitable for the experiment 3-layer, 4-layer, or 5-layer. CNN model showed that the suitable number of layers for the dataset provided was four layers and the model produced better results compared to using three or five layers. The third study tested the ability of the noise resistance and the result showed that the CNN model could withstand some level of noise, although when the percentage of noise in the dataset is higher than 5%, the impact of the added noise progressively becomes apparent in the accuracy of the model.

In [48], the CNN classification method for plankton was proposed to solve the imbalanced class problem (number of samples per class is not balanced), which is more than 90% of images with only 5 image classes. This problem was resolved by transferring CNN pre-training models by reducing the amount of data for a class with a large sample size. Then the model is fine-tuned by retraining the model with the actual data to recognize the

class distribution in order to avoid bias. The study proposed a variable, accurate CNN classification model for a large database. Moreover, the study considered reducing the class bias of classes that have small-sized, class-normalized data constructed by the data for large classes. The proposed model improved performance for small classes and also retained good accuracy with learning transfer which was compared to the classifier trained with original data.

In [49], difficulty in classifying the microalgae organism because limited characteristics of microalgae, such as size and shape, have been addressed. The result of a deep learning model (part of machine learning that artificial neural networks and human brain-inspired mechanisms learn using a huge amount of data) with CNN to classify the datasets of microalgae images collected using the FlowCAM device was provided. To improve the results, an augmentation method was used to increase the data for the training process. The increase in training data showed an improvement in accuracy of approximately 17 % points. The method demonstrates its advantage, since the results of the model have given much better accuracy than the accuracy achieved by humans.

The study in [50] explored a new deep learning network method, ZooplanktonNet for automated and efficient classification of zooplankton. Many traditional and common learning models are utilized to classify zooplankton, and some significant aspects in networks such as depth and width are investigated to demonstrate that. ZooplanktonNet model can improve performance when it takes into account the loss, time of training, and difficulty of the model. The result showed that ZooplanktonNet with 11 layers for 13 classes with 9460 images was successful in solving the problem of classification with an accuracy of 93.7%.

In [51], the CNN input layer followed by five convolutionary layers, and two fully connected layers were used to classify the dataset. The convolutional layer separated into three parts, called filter, nonlinearity, and sub-sampling layers. The nonlinearity layer contains the ReLU activation function that has been used for all neurons and accelerated convergence. Comprehensive experiments and assessments of CNN accuracy were conducted using three different plankton datasets chose from the SIPPER dataset. The results of the experiments using SIPPER datasets containing 750000 samples and 81 different classes [51] showed a high accuracy of 98.20% in classification performance.

In [52], the design of the CNN was based on the AlexNet and has been adapted to fulfill the criteria of the plankton dataset. ReLU activation is executed after each layer, except the last layer consisting of softmax activated output. The ability of the cost-effective

active learning (CEAL) method to train the CNN for classifying plankton dataset was verified. Active learning is a case of machine learning in which the process of recognizing the data is necessary to be labeled in order to have the greatest impact on the model. For CEAL, a professional expert only needs to explain a fraction of the samples to get a high-performance of CNN. The method thus helps to reduce the divergence between automated large-scale image data collection systems and the real explanation of biological application data by automatically deduce the necessary information.

3 SEMI-SUPERVISED LEARNING

3.1 Principles of semi-supervised learning

Semi-supervised methods [53] are methods between unsupervised methods (training without labeled data) [54] and supervised methods (training with only labeled data) [55]. A common concept is that semi-supervised methods cluster the dataset using an unsupervised method and then utilize the existing labeled dataset to give a label to the remaining unlabeled dataset [53].

Semi-supervised learning solves the difficulty of having a huge amount of unlabeled data by learning with partly labeled datasets. The unlabeled datasets of training provide useful details on the estimation problem at hand which can lead to a good performance of the predictive function. It is therefore helpful to rely on both labeled and unlabeled instances when learning a predictive function [56].

Moreover, semi-supervised learning has the advantage that human labor is being reduced and that performance is being increased. It is theoretically possible to use the labeled and unlabeled data to execute greater outcomes compering to supervised learning. Alternatively, semi-supervised learning can achieve the same level of accuracy as supervised learning, however, but with less labeled dataset [56–58]. Due to the high performance shown by semi-supervised methods, particularly in large datasets, semi-supervised learning methods are recommended when compared to supervised and unsupervised learning methods [11].

The semi-supervised learning has some assumptions such as smoothness, clustering, and manifolds [56]. The fundamental assumption in semi-supervised learning is smoothness which means that when two points are near in high-density areas, they should have similar labels. The second assumption is the cluster assumption, implying that if two instances are in the same group, they probably belong to a similar class. It is unlikely to have two instances in the same group corresponding to different classes. The third assumption is manifolds, the data is based on a manifold of lower dimension than that of the initial data domain. A manifold uses both labeled and unlabeled dataset can eliminate a dimensionality of the data. Learning can progress by using the distances and densities described in the manifold. Many of semi-supervised methods depend on those assumptions. Figure 7 [56] shows an illustration of the cluster and manifold assumption.

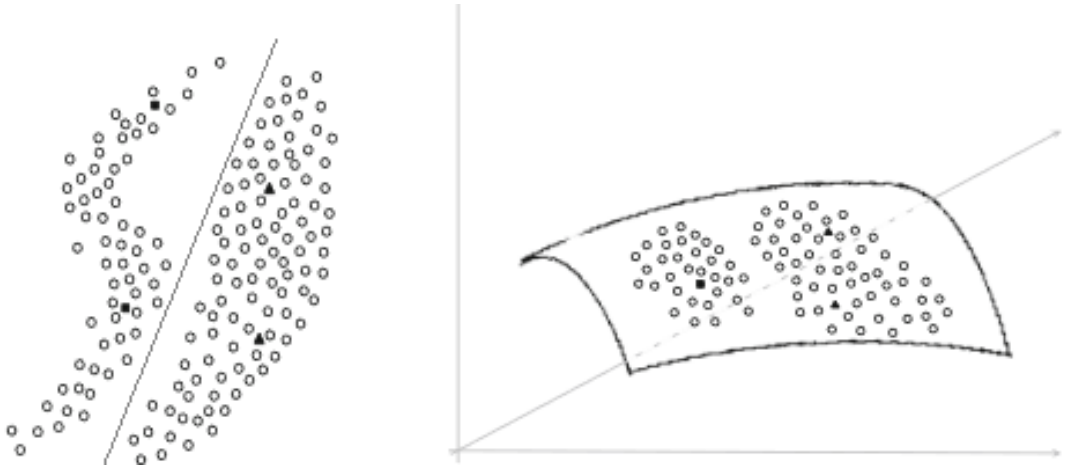

Figure 7. The explanation of clusters and manifold assumption. Unlabeled samples are defined by an open circle, and labeled samples of various classes are defined by a filled square and a filled triangle. Left: The decision boundary passes cross low-density regions. Right: low dimensional topological areas or geometric manifolds [56].

Several semi-supervised learning approaches have emerged in the literature. These approaches include self-training [11], semi-supervised support vector machines [59], pseudo labeling [60], co-training [61, 62] and graph-based [63] methods.

3.2 Semi-supervised learning methods

This subsection describes the main semi-supervised deep learning methods such as Generative models [64], self-training [53], graph-based [63], co-training [61, 62] and Pseudo labeling [60].

Self-training [53] is a widely used technique for semi-supervised learning. The key concept behind self-training is to train a classifier with labeled data first. The classifier is implemented to estimate labels of unlabeled data. The selection of the best-unlabeled data with confident predictions over unlabeled examples and their estimated labels are selected and applied to the training set. In the new training collection, the classifier is retrained and the process is repeated. In self-training, the classifier is training itself by using its predictions. The cycle of the retraining of the classifier by itself is called self-teaching or bootstrapping. Figure 8 shows an example of the self-trained learning architecture for a Medical radiology reports dataset. The dataset contains two classes which were normal class for the radiology reports without any fracture or dislocation and abnormal class for the radiology report with fracture or dislocation [65].

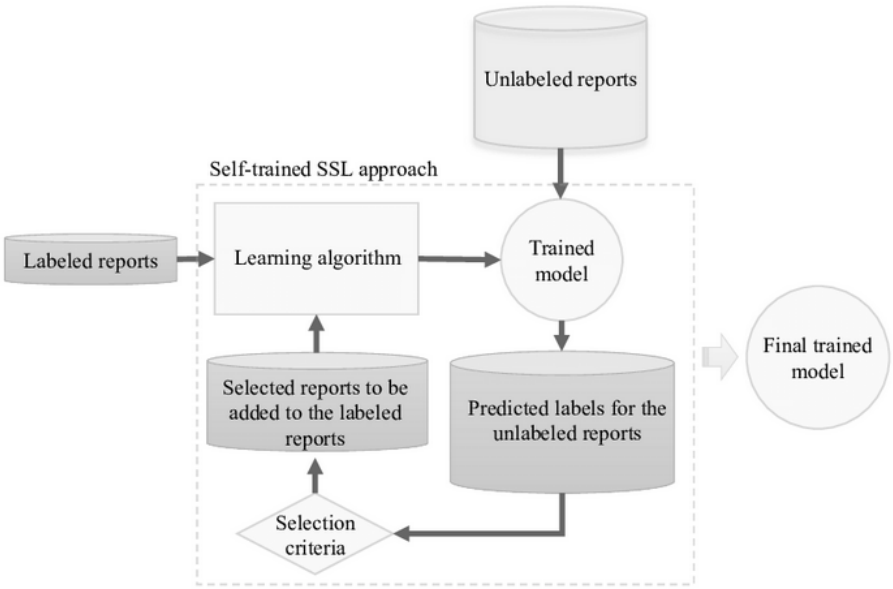

Figure 8. An example of the self-trained learning architecture for a Medical report dataset [65].

Co-training [61, 62] is a semi-supervised method that requires dual data views, meaning that each instance is described using two set of features. This gives different complementary details of the instance. Dual views are ideally conditionally independent, given that each feature case is conditionally independent and each view is adequate. From each point of view, the class of the instance can be precisely estimated. Co-training begins by learning the model for each view using samples with a label. The best estimate of the model for the unlabeled data is used to iterate more labeled training data. The classifier is trained with a set of features and used to classify unlabeled data. Another classifier allows iterative use of the best estimates of each classifier on unlabeled data as the labeled training data [53].

The work [66] combines an unsupervised method with a supervised method in deep neural networks. The proposed model demonstrated how a simultaneous unsupervised method was able to enhance CNN and MLP methods to achieve state-of-the-art of some semi-supervised learning using a small amount of labeled data. The proposed method implements the Ladder network for MLP networks and convolutional networks then standard rectifiers with batch normalization used to each pre-activation process. The result has shown that the proposed method can improve unsupervised learning. Nevertheless, the model also achieves noticeable improvement when compared to the supervised model with labels sample in the Modified National Institute of Standards and Technology database (MNIST) with 60000 samples in 10 classes. In suggested model the training is fast, convergence is quick, and also a greater improvement in performance was noticed

in the proposed models with the amount of dataset compared to the number of obtainable labeled data.

In [67], the semi-supervised learning using the minimum entropy regularizer (applied to encode learning priorities) in the supervised learning system was shown. The regularizer is a functionality that can be used to encode priors for learning. It is also used to generalize the methods to continuous input spaces, or as a standard for learning parameters of weight function in the form of transduction on training manifolds. The regularizer is inspired by the theory that indicates that the unlabeled samples are most helpful when classes get a small overlap.

The maximum a posteriori (MAP) system [67] offers the possibility of managing the weight of unlabeled cases while avoiding optimism when unlabeled data appear to cause failure to the classification. As a specific case, the proposal involves self-learning, as minimizing entropy that enhances confidence in the efficiency of the classifier. The results of the Grandvalet and Bengio experiments suggested that the limited regularization of entropy could be a serious contender for generative systems (models apply to the joint density of patterns and class). The systems compared in three cases: for a small number of samples, where the generative model cannot completely benefit from the information provided by the appropriate mutual process; when the distribution is incorrect; and when the unlabeled data appears to be uninformative about class probabilities.

In [58], a method of semi-supervised learning for solving classification problem is presented. The method known as Self-Organizing Map for semi-supervised learning which cluster the dataset without knowing the label of the input data. During the training process, the Self-Organizing map swap dynamically among the supervised and unsupervised methods depending on the available label of the class. The method showed substantial changes in the outcomes of classification when the size of the label data is small. Self-Organizing Map for semi-supervised learning compared with traditional methods such as MLP and SVM, the result using the REAL-WORLD dataset showed Self-Organizing Map achieving good accuracy 0.935 % when using 100% of the labeled data.

One of the most common methods for semi-supervised learning among recent years is graph-based technique [68]. The method begins with creating a graph from the training dataset. It further describes the graph in which the nodes are described in the dataset as the labeled and unlabeled samples, and the edges show the similarity among the samples. This technique generally assumes the smoothness of the label over the graph and no parameters were needed for the application of this method.

The graph-based technique implies that both labeled and unlabeled data contained in a low-dimensional manifold which means the data can logically be described in a graph. In a weighted graph, a vertex is described for each sample, with the weights giving a measure of vertex. Figure 9 shows an example of a Graph-based approach to addressing a semi-supervised learning problem [63]. The steps for this approach are as follows:

- Graph creation if no graph exists.
- Inserting seed labels on a subset of points.
- Labels the unlabeled points in the graph [69].

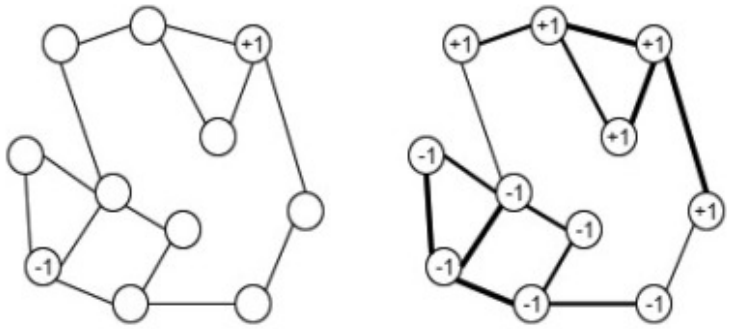

Figure 9. Graph-based classification of dataset. Left: Before classification, Right: After using Graph-based for classification [63].

4 Semi-supervised plankton classification

4.1 Semi-supervised learning pipeline

This subsection describes the ideas and steps for the method used in classifying plankton images in this research. The pipeline consists of the following steps (see Figure 10):

1. Pre-processing: This is to ensure the original image is appropriate for use by resizing the images to a target size, converting the images to grayscale, and applying data augmentation.
2. Feature extraction: Here, the features in the image are extracted using convolutional neural network architecture (CNNs).
3. Classification: Applying a semi-supervised method such as label propagation, label spreading, or Pseudo labeling using the extracted feature as an input of the methods.

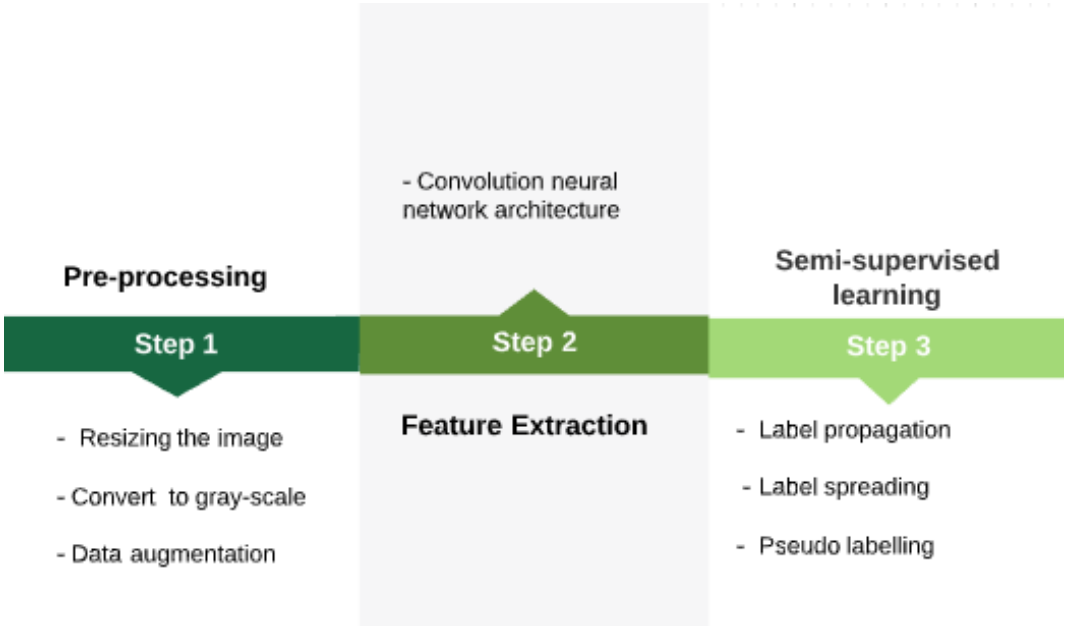

Figure 10. The pipeline for implementing semi-supervised plankton image classification.

4.2 Supervised CNN for feature extraction

This subsection outlines the procedure and concepts associated with feature selection and learning the data through CNN.

Feature extraction is the method for obtaining representative features such as color, texture, histogram features from the pre-processed data. Generally, feature extraction is a way for reducing the dimensional of the input dataset. It has several advantages such as reducing the cost of calculation related to the classification problem and playing a key role in improving the accuracy and visualization of the dataset. Sometimes feature extraction helps to reduce the over-fitting risk when the model learns the data almost completely. In particular, feature extraction helps to have an overview of the data structures and to get a better understanding of the model [70].

The proposed CNN consists of three conventional layers, max pool layers, and tow full output layers. The first step of the feature extraction process is that each image should split into equal size groups called patches. These patches overlap by sharing certain common parts of the image required for feature extraction [71]. The convolutional layer produces a number of feature maps. For one feature map, the neurons are sharing similar weights. At the same time, hidden units cover different patches, each patch of the image is connected with several feature maps. In various feature maps, the neurons extract different features from a similar patch [72]. Max-pooling layer takes the highest value in each window which exposes invariance to local transformations (because pooled value remains similar in the neighborhood). It decreases the representation of inputs such as the images and the hidden layer output matrix. Finally, the output of the convolutional layer gives features of the input image [72].

4.3 Semi-supervised learning methods

This section explains the semi-supervised methods used in predicting the class according to the labeled and unlabeled data after the features extraction process.

4.3.1 Label propagation

Most of semi-supervised methods assume that nearest data points belong to have the same or similar labels of class in a way similar to k-Nearest-Neighbor (K-NN) in classical supervised learning. Though such methods are propagating labels by a lot of unlabeled data areas of the problem. In a specific model of label propagation, the labels of the node propagate to all nodes depending on their proximity. At the same time, the method fixes the labels on the labeled dataset group. Consequently, the labeled data act as a source that pushes the labels via unlabeled data [73].

Zhu and Ghahramani [74] presented the Label propagation algorithm (LP). It is one of the most common semi-supervised methods that follow a transductive methodology with two assumptions about the information. The first assumption, smoothness is that if there are two data points near to each other, the labels of these points are probably to be similar. The second assumption, cluster is that if two data points have a similar cluster, they are probably to have a similar label.

In the LP algorithm, each node starts with a unique label for the group. Each iteration label propagates across the network and every single node that creates a new label depends on the maximum number of its neighbors. LP achieves convergence when all node has a label of a popular neighbor label. The algorithm stops whenever the convergence or the highest number of iterations is reached. The densely linked node groups agree on a unique label, only a little number of labels can remain. In the end, nodes with a similar convergence group classify the same class [73].

Label propagation does not work well when the data have a high dimension or the data have a very curved shape. To solve the problem of dimensionality, the labels (Y) representation can be changed to the soft format (low-dimension), then estimate the result of the initial estimations of labels Y using the probability distributions. With LP, the edges (E) are weighted to describe the relationship between nodes, the highest weights representing the highest similarity. Such edges can be represented using a matrix weight (W). The nodes that are labeled are used to propagate knowledge to all nodes. The node has a greater weight used to cause this propagation, the more frequent the label occurs. Since the estimated label is in a soft labeling format, LP considers the unlabeled node that has a hard label to be the one with the highest confidence.

In Algorithm 1, D is the diagonal degree matrix, $\hat{Y}$ represent estimated label, Y_l and $\hat{Y}_l$ refers to label and estimated label of node l, and W refers to the affinity(probabilistic)

matrix given by the Gaussian kernel, the kernel width is σ and the datasets are x_i and x_j. Measurement of W according to the follows:

$$W_{i,j} = \exp - \frac{||x_i - x_j||^2}{2\sigma^2} \tag{1}$$

The LP algorithm is represented in Algorithm 1 [75]:

Algorithm 1 Label propagation [75]

Result: Labeled data (data x_i labeled by the sign of $\hat{Y}_i^{(\infty)}$)
Calculate W from Eq 1
Calculate D, $D_{ii} \longleftarrow \sum_j W_{ij}$
Create $\hat{Y}^{(0)} \longleftarrow (y_1, ..., y_l, 0, ..., 0)$
Iterate
    I. $\hat{Y}^{(t+1)} \longleftarrow W\ (D)^{-1}\ \hat{Y}^{(t)}$
    II. $\hat{Y}_l^{(t+1)} \longleftarrow Y_l$
until convergence reaches $\hat{Y}^{(\infty)}$

The propagation process of LP is executed sequentially until each node gets the most common label of its neighbors. As well, the groups in LP are specified as node classes that carrying similar labels. LP has a striking characteristic that is less costly to implement, the complexity closer to linear time complexity. The drawback of this method is that LP is not always stable, the method is responsive to the order of updating of node labels in each stage. Thus, the results and their corresponding values can somehow have slightly different in diverse runs [76].

To propagate the labels needs to clamping the data that have labeled and incrementally normalizing the interpretation of probability. Clamping the outcome guarantees that after normalization the initial labels have not changed. It also needs to build the label distribution which is an array with label probability before doing the iteration. Each column in the array reflects the index of each label. Through iteration determines the probability for every element, and updates the distribution. This will be achieved by calculating the dot product of the array of similarity graph and existing distribution, and the result is then normalized [77]. Figure 11 shows the idea behind how the LP method works [73].

The label distribution should be normalized and clamped until all nodes have a similar label as most of their neighbor-nodes, until the process converges. Figure 12 demonstrates

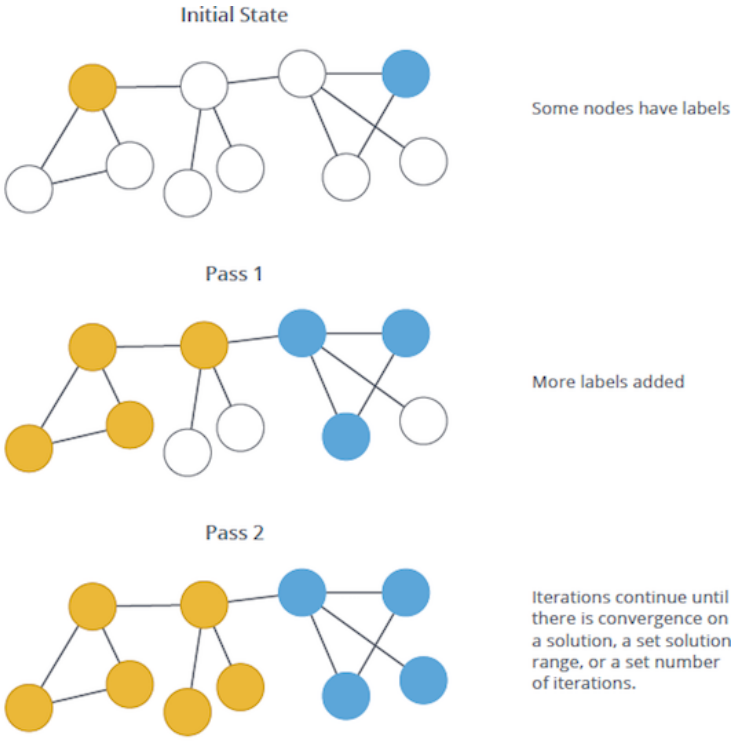

Figure 11. Label propagation semi-supervised learning method [73].

how the distribution of the label shifts to a certain degree after the various amount of iterations [77].

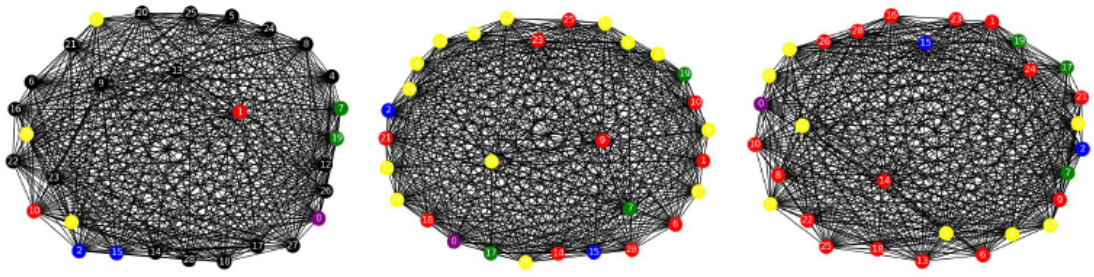

Figure 12. The distribution of the label after various iterations [77].

4.3.2 Label Spreading

In [78], Zhou et al. introduced a new method of semi-supervised learning call Label spreading (LS). The Label spreading method is similar to label propagation. The difference between the LS method and the LP method is the LS the design of probabilistic transition matrix (matrix related to the variation of the base for vector space) depending on the normalized Laplacian graph while in the LP depends on the Laplacian graph. This indicates that LP utilizes a direct similarity matrix built from data with no modifications, while LS reduces a loss function which has regularization features, which also makes it easier to be resilient to noise [78].

In LS, in each step, the node (i) gets the participation of its neighbors (j), weighted by the normalized edge weight (i, j), and a small contribution provided by the initial value. In Algorithm 2, D is the diagonal degree matrix, $\hat{Y}$ represent estimated label, L is normalized Laplacian graph, and W refers to the affinity matrix given in Eq 1. The LS algorithm presented in Algorithm 2 [75]:

Algorithm 2 Label Spreading [75]

Result: Labeled data (data x_i labeled by the sign of $\hat{Y}_i^{(\infty)}$
Initialization
Calculate the affinity matrix W for $W_{ii} \longleftarrow 0$ and $j \neq i$.
Calculate the diagonal matrix D, $D_{ii} \longleftarrow \sum_j W_{ij}$.
Calculate the normalized Laplacian graph $L \longleftarrow D^{\frac{-1}{2}} W D^{\frac{-1}{2}}$.
Create $\hat{Y}^{(0)} \longleftarrow (y_1, ..., y_l, 0, ..., 0)$.
Select $\alpha \in [0, 1)$.
Repeat the iterate $\hat{Y}^{(t+1)} \longleftarrow \alpha L\hat{Y}^{(t)} + (1 - \alpha)\hat{Y}^{(0)}$, until reach convergence at $\hat{Y}^{(\infty)}$.

The label spreading method operates by describing a data as a point and then identifying the other points nearest to it, so the method figures out how to label the unlabeled data regardless of how near they are to the training set. Figure 13 shows an example of learning using the label spreading with the K-Nearest Neighbors (K-NN) kernel. The classes label lies within their distinct category, and then the labels are properly spread across the circle [79].

In both LS and LP, the graph construction is built from the matrix of feature selection, which depends on data reprocessing. Each value in the matrix of the features is repre-

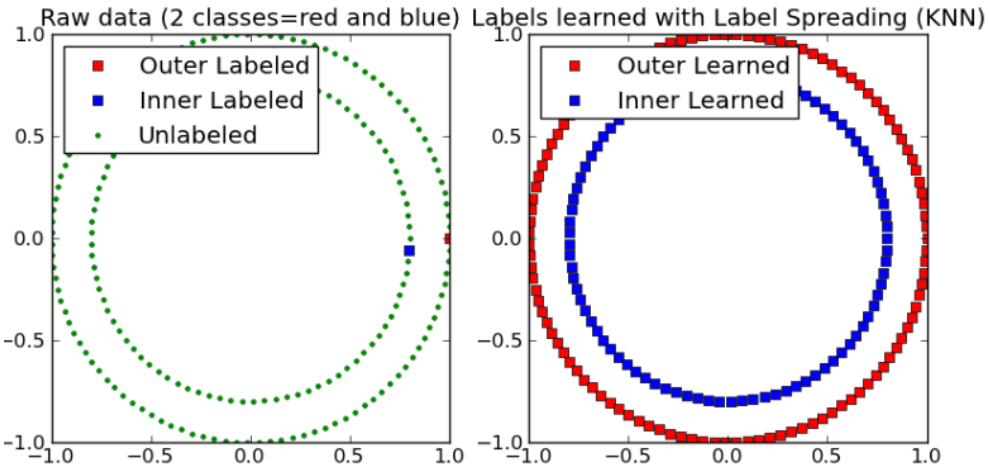

Figure 13. Label Spreading semi-supervised learning with K-NN kernel [79] .

sented as a vertex (node) in the graph and has weighted edges depending on the resemblance of the data [80].

There are several ways of measuring the similarity between the data sample, the K-Nearest Neighbors (K-NN) and Radial Basis Function Kernel (RBF kernel) are two common methods. The K-NN kernel determines the distance by selecting a small positive value for K. After that, the method discovers the K nearest neighbors of the data to be classified, and the dataset labels are the same as the common neighbor class. It is essential to select K value appropriately, as this value can render the algorithm to under-fit or over-fit [80]. The RBF kernel is described as

$$R(N1, N2) = \exp\left(-\gamma \, ||N1 - N2||^2\right) \tag{2}$$

where $||N1 - N2||^2$ is the square of the Euclidean distance between the vector feature in the first node $N1$ and the vector feature in the second node $N2$ and γ are float number between [0-1). The RBF kernel generates a completely connected graph that is described by an array. This array could be very big when merged with the expense of performing a complete array multiplication calculation for every iteration of the method, this leads to exorbitantly long times for running [79].

4.3.3 Pseudo labeling

The Pseudo label [60] is one of the easy, simple, and most efficient methods of semi-supervised learning for deep neural networks. Essentially, this network is trained concurrently in a supervised manner using unlabeled and labeled data. For unlabeled results,

pseudo-labels can be used as if they were real labels. Figure 14 shows how the Pseudo labeling technique works [81].

As a result, this is equivalent to entropy regularization. This prefers a division of low density among classes which is a general assumption of prior art for semi-supervised learning.

Pseudo labeling classifies the data according to the follows steps:

- The classifier utilizing only the training dataset,
- estimating the labels for the unlabeled test dataset,
- add confident estimated class to the training dataset,
- constructing a model using both original training dataset and the confident estimated dataset,
- using the new model to classify the test dataset [82].

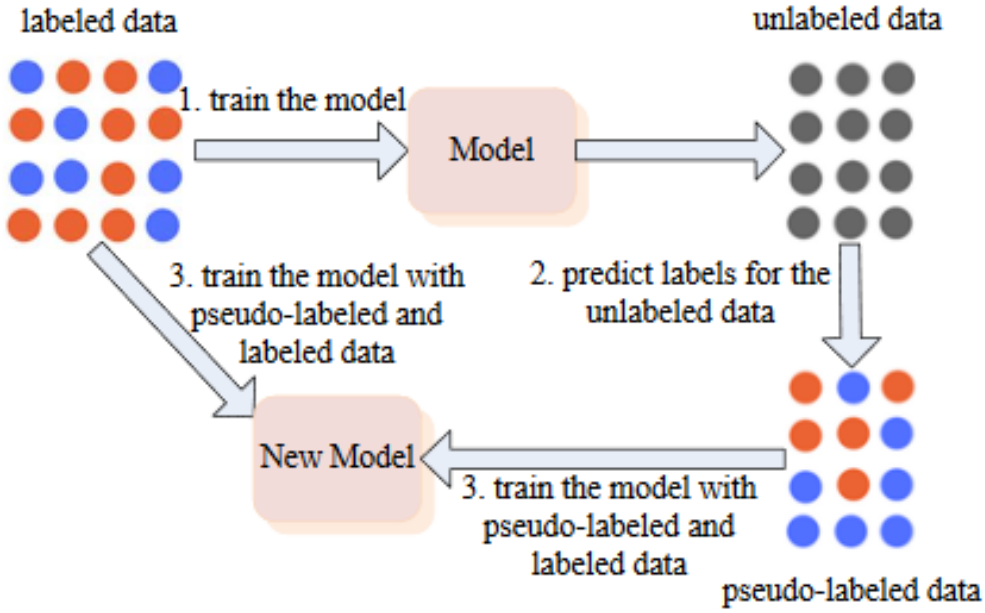

Figure 14. Pseudo labeling process [81].

Pseudo labeling is take the class which has highest predicted probability for any unlabeled sample as follows [82]:

$$y_i^{'} = \begin{cases} 1 & \text{if} \quad i = \arg\max_i, f_{i'} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where $y_i^{'}$ is pseudo labeling and $f_{i'}$ is unlabeled data.

The pre-trained framework is trained concurrently in a supervised manner. For unlabeled data, each weight recalculated by a Pseudo label is used for a similar loss function as a supervised method. Since the overall number of labeled data and unlabeled data are different and the training equilibrium between them is critical for the output, the loss function can be calculated as

$$L = \frac{1}{n} \sum_{m=1}^{n} \sum_{i=1}^{C} L(y_i^m, f_i^m) \; + \; \alpha(t) \frac{1}{n^{'}} \sum_{m=1}^{n^{'}} \sum_{i=1}^{C} L(y_i^{'m}, f_i^{'m}) \tag{4}$$

where n is number of the mini batches for label dataset and $n^{'}$ for unable, f_i^m is the output of m label data, y_i^m is label data, $f_i^{'m}$ is unable data, $y_i^{'m}$ is pseudo labeling and α for balancing [82].

Several works [60, 83, 84] are devoted to the discovering of an optimum labeling scheme. Figure 15 shows Pseudo labeling approaches [84]. The approaches can be summarized as follows:

1. All-in-one [83] is implemented by searches for the simplest way for creating labels. This essentially adds additional new class and combines all unlabeled data directly into this group without taking into account any differences that might occur among all images. Across the training phase, unlabeled data are equipped with this set new class label.

2. One-hot [60] is a method based on the all-in-one, One-hot considers the difference of all images produced and proposes to presume that every sample belongs to a current class. Pseudo-labels are given by considering the highest value of the probability estimation for each class.

3. Distributed [84], this way of Pseudo labeling also expands the one-hot scheme and claims that the label should be spread as shown in Figure 15.c for an unlabeled data.

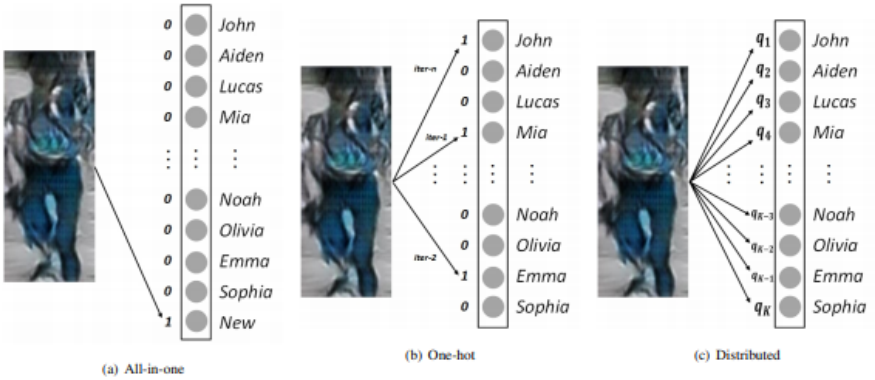

Figure 15. Pseudo labeling approaches [84]: (a) All-in-one; (b) One-hot; (c) Distributed methods [84].

5 EXPERIMENTS

This chapter describes the data used in this research, the data preprocessing steps, the experiments carried out, the evaluation criteria for the classifiers, and the results.

5.1 Data

The plankton dataset used in the research was provided by the Marine Research Center of the Finnish Environment Institute, and is related to the FastVision project [85]. These images have been captured using an automated submarine imaging flow cytometer called Imaging FlowCytobot. This device capable to capture images with a resolution of 3.4 pixels per micrometer in sizes between 10 to 100 micrometers and transformed the captured images from analog (in photomultiplier tubes) to digital (in devices) [4].

The dataset consists of 86 different classes. The classes have a large imbalanced class distribution, with samples ranging from one image per class to more than 4,000 images per class. The Dataset images have been described in 2-dimensional objects. Table 1 shows the name of the class and the number of samples in each class. One of the classes is called the unclassified class. The class which contains a large number of samples that the expert was not able to label.

More than half of the classes contain less than 100 samples per class which is not sufficient to train the CNN model. To obtain more samples, the dataset needs to be increased using data augmentation processes. A sample of the original dataset is shown in Figure 16. Table 2 shows the dataset groups for the experiments.

Some of these classes can be compiled in one class because those classes belong to the same category. These classes are Chroococcales with Chroococcus-small, Ciliata with Ciliata-strawberry, Cryptophyceae-Euglenophyceae with Cryptophyceae-small, Dinophyceae-under20 with Dinophyceae-over20, Dino double div and Dino small funny, Dolichosper Anaba with Dolichos Anaba round, Euglenophyceae with Euglenophyceae-big, Mesodinium-rubrum with Mesodinium-rub-shrun, Pennales-sp-basic with Pennales-sp-boxy and Pennales-sp-curvy, and Snowella-Woro-sp-den with Snowella-Woro-sp-loos.

Table 1. Number of samples per class for the whole dataset.

Class name	size	Class name	size	Class name	size
Unclassified	82028	Eutreptiella-sp	236	Pennales-sp-curvy	28
Oscillatoriales	4402	Heterocyte	234	Binuclearia-lauterborn	23
Dino-small-funny	3088	Prorocentrum-cord	229	Gonyaulax-verior	22
Snowella-Woro-sp-den	2275	Pennales-sp-boxy	171	Chaetoceros-similis	29
Pyramimonas-sp	1602	Cyst-like	150	Aphanothece-parallel	29
Dolichosper-Anaba	1550	Gymnodinium-like	150	Akinete	19
Skeletonema-marinoi	1517	Peridiniella-cat-chain	144	Amylax-triacantha	19
Heterocapsa-triquetra	1464	Cymbomonas-tetram	132	Euglenophyceae-big	17
Dinophyceae-under20	1412	Ciliata-strawberry	126	Scenedesmus-sp	14
Thalassiosira-levanderi	1164	Pauliella-taeniata	119	Cryptophyceae-Euglen	13
Teleaulax-sp	1132	Beads	100	Apedinella-radians	13
Aphanizom-flosaquae	1072	Cyclotella-choct	99	Chaetoceros-thronds	12
Snowella-Woron-loose	1042	Dino-double-div	99	Nostocales	8
Chaetoceros-sp	952	Chlorococcales	96	Chaetoceros-resting	8
Peridiniella-cat-single	875	Chroococcales	93	Dinophyceae-over20	7
Pseudopedinella-sp	835	Uroglenopsis-sp	92	Dinobryon-balticum	7
Dolichos-Anaba-round	817	Ciliata	91	Pauliella-res-stage	5
Chroococcus-small	791	Dinophysis-acumin	89	Chaetoceros-subtilis	5
Pennales-sp-basic	777	Nodularia-spumigena	84	Chaetoceros-danicus	4
Heterocapsa-rotundata	624	Merismopedia-sp	79	Aphanizomenon-sp	3
Mesodinium-rubrum	610	Euglenophyceae	76	Melosira-arctica-res	3
Oocystis-sp	597	Cluster-A	72	Dinophysis-norvegica	3
Cryptomonadales	422	Nitzschia-paleacea	65	Dinophysis-sp	2
Mesodinium-rub-shrun	352	Licmophora-sp	63	Nodularia-spumig-het	2
Centrales-sp	338	Katablepharis-rem	54	Gymnodinium-sp	2
Cryptophyceae-small	316	Gymnodiniales	50	Coscinodiscus-granii	2
Monoraph-contortum	303	Melosira-arctica	43	Rotifera	2
Cryptophyceae-drop	265	Ceratoneis-closterium	39	Amoeba	1
		Protoperidinium-bipes	1	Flagellates	1

Table 2. The dataset groups for the experiments.

Dataset	Class size	Number of classes
Dataset 1	Classes with $\geq$ 1000 images	5
Dataset 2	Classes with $\geq$ 100 images	38
Dataset 3	Classes with $\geq$ 3 images	78

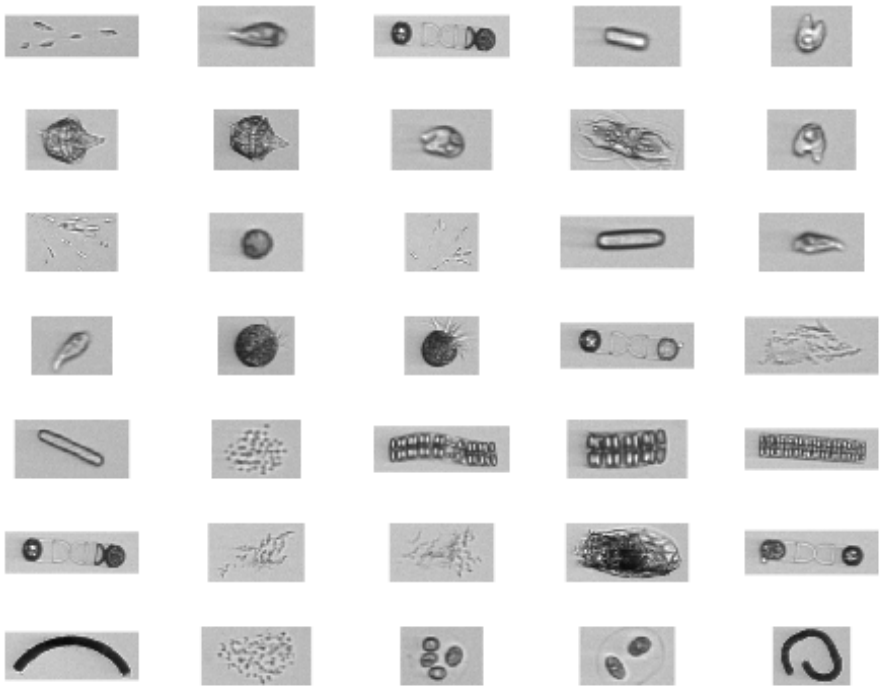

Figure 16. Phytoplankton images dataset samples.

5.2 Data preprocessing

Before using the image dataset, the images need to be adapted to the CNN method. The input images had to be in the same size. Some images had large size need to reduce to fit in the specific frame, and others are small need resize to fit the frame. In this study, the images were scaled to be in size 100×100 of the input images.

Data augmentation techniques were used to increase the amount of data in classes containing a small number of samples. The process of increase was to rotate the image to a certain degree, such as ± 10 degrees, as well as horizontal flip, vertical flip, and crop large area from the image randomly. The data augmentation process was carried out after the data were split into sets. The augmentation techniques enable a large amount of data to be produced that was useful for training. Figure 17 shows the dataset of the images after resize and augmentation processes.

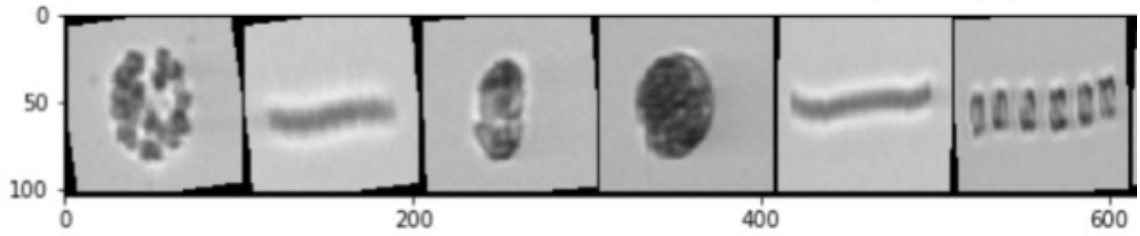

Figure 17. Image dataset after resizing and augmentation.

Figure 17 shows that the process of augmentation has made changes to the images, such as the black border in the background image. These changes may increase the potential for misclassification. In order to solve this problem, it is necessary to fill the black pixel in the border of a new image to the same color of the image background. Figure 18 shows the images after they have been filled.

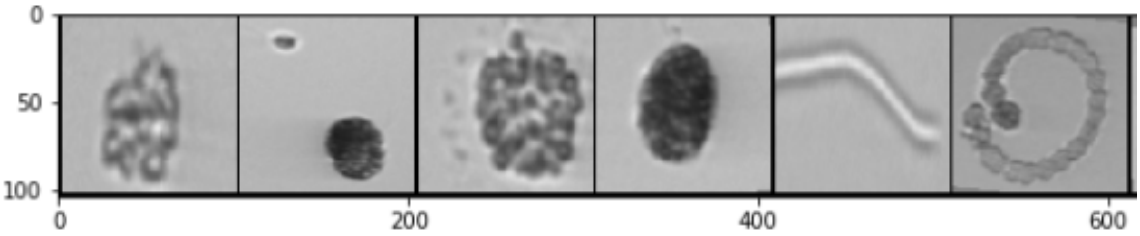

Figure 18. Images after an adjustment process.

5.3 Description of experiments

The research focused on three semi-supervised methods to classify the plankton images dataset. The proposed methods were Label propagation (LP), Label spreading (LS), and Pseudo labeling for the semi-supervised methods. Moreover, fully supervised CNN was used as a baseline.

The CNN method was used to extract the features and to classify the plankton images in a supervised manner. The proposed CNN-based feature extraction used to extract the features, and consistently optimize the CNN parameters using Adam optimizer. Adam optimizer is a learning optimization technique, specifically designed to train deep neural networks. The batch normalization was then applied to the output of each convolutional layer in order to avoid saturation in the layers. The batch normalization had the impact of stabilizing the training process and significantly decreasing the number of epochs.

The used CNN architecture network shows in Table 3 consists of an input layer followed by three 2D-convolutional layers, 2D-Max-Pooling with 2×2 filters, and the two fully connected layers. The first convolutional layers involve one input channel, 16 output channels, and the kernel size was 3. The second layer consists of 16 input channels, 8 output channels, and a kernel size of 2. The third layer consists of 8 input channels, 8 output channels, and a kernel size was 3. For the fully connected layers, the first layer included 11 ×11 × 8 input features and 50 output features, while the second layer included 50 input features and the output features equal to the number of classes used.

Table 3. A table about CNN architectural parameters.

Convolutional layers			
Layers	**Input channel**	**Output channel**	**kernel size**
CONV 1	1	16	3
CONV 2	16	8	2
CONV 3	8	8	3
Fully connected layers			
Fully connected layers	**Input feature**	**Output feature**	
FC 1	11 ×11 × 8	50	
FC 2	50	number of classes	
Max polling layer			
Layer	**Kernel size**		
Max pooling	2		

The CNN network was trained to use the Adam optimizer to reduce the loss of classification. In the experiment, a mini-batch technique was used to improve the learning rate, with a batch size of 64 images. The learning rate was also one of the key factors in the training of the model if the learning rate was high then the pace of each update became high, but if the rate was low then the model would be trapped at a minimum speed and would never again be eligible. The model proposed learning rate was 0.01.

Other important factors included dropout (a regularization approach for the decreasing over-fitting in neural), ReLU activation function, window size, and number of epochs. These factors have been significant and had an impact on final accuracy. Factors analyzed as follows: Dropout plays a significant role in adjusting the over-fitting, the drop-out ratio was set at 0.2. Rectified linear units (Relu) were used as an activation function for CNN. The Window size was 2×2, and 20 training periods were used. The images should be trained multiple times to ensure that the model was fully convergent.

The LP and LS were implemented using the ready package given in the scikit-learn library [79]. The plankton dataset and its related labels were transmitted through the LS and LP algorithms that construct the graph. The LP and LS algorithms used two kernels (K-NN and RBF) to classify the data. The K-NN and RBF kernel used various neighbors, γ, and maximum iterations.

To implement the experiments, the dataset was divided into three groups. The first group with a small number of classes which was 5 classes (Dataset 1), the second experiment with a large number of classes, all the classes that had more than 100 images per class (Dataset 2), and the last experiment with all dataset (Dataset 3). Then each group was split randomly into training sets, support set, and testing sets.

The parameters for implemented experiments were described as follows: The LS method with K-NN kernel used 50 maximum iterations, the α was 0.01 and the number of neighbors was 20. While the maximum iteration for the RBF kernel was 1000 and γ was 0.03. The LP method with K-NN kernel used 1000 maximum iterations, the α was 0.001, and the number of neighbors was 50 and 100 depending on data size. While the maximum iteration for the RBF kernel was 2000 and γ was 0.03.

CNN parameters value such as learning rate, dropout ratio and LP and LS parameters such as α, γ, maximum number of iterations, number of neighbors selected by running iteration with different increments to select the appropriate parameters to use.

Experiment 1: balanced class distribution experiment using Dataset 1 with the following five classes: Dino small funny shaped, Dolichospermum Anabaenopsis, Oscillatoriales, Skeletonema marinoi, and Snowella Woronichinia sp-dense with a large number of samples in each class. The steps used for the experiment are shown as follows:

(A) The experiment when the number of samples in each class was the same for all support, test, and train sets. Each set contains 1500 images.

(B) The effect of the different sizes of the support set of 1500 samples, 1000 samples and 500 samples, and a fixed training set of 1500 samples.

(C) The effect of the different sizes of the training set of 1500 samples, 1000 samples, and 500 samples with a fixed support set size of 1500 samples.

Experiment 2: imbalanced class distribution experiment using Dataset 1 with a large support set size of 7500 samples and a large training set size of 7248 samples.

Experiment 3: balanced class distribution experiment using Dataset 2. The data augmentation processes were used to increase the data. The results showed the accuracy and the comparison before and after the augmentation processes.

Experiment 4: imbalanced class distribution experiment using Dataset 3. The dataset divides into three-set, 40% of the data used for training, 40% for support set, and 20% of the dataset was used for the test set. Then the data augmentation process was used to increase the data.

The results were evaluated by calculating the accuracy of all data. Accuracy measures the percentage of data that correctly predicted between all datasets.

The accuracy is defined as follows in Eq 5 [86]:

$$\begin{aligned} Accuracy &= \frac{Correctly\ classified\ samples}{Total\ amount of samples} \times 100\% \\ &= \frac{(TP+TN)}{(TP+TN+FP+FN)} \times 100\% \end{aligned} \tag{5}$$

where TP represents the number of true positives values, FN is false negatives values, FP is false positives values, and TN and true negatives values [86].

5.4 Results

This subsection shows the results of four experiments being carried out. The classifiers trained with the dataset and cross-validation were used to provide sufficient confidence in the results and to check how the model was sensitive to the new dataset.

The Four experiments run at least three times to check the sensitivity of CNN's extraction feature model to dataset change. The experiment found that the difference between the results was very small, which means that the model used was not sensitive to the dataset.

The classification accuracy of all methods used in Experiment 1.A is shown in Table 4. The highest accuracy was achieved by the LS (K-NN) method followed by the LP (K-NN), LP (RBF), LS (RBF), Pseudo label and the worst accuracy for the CNN method. The semi-supervised methods improved accuracy compared to the CNN method in the plankton image dataset when a balanced class and enough data were used to train the model.

Table 4. Comparison between different semi-supervised methods in Experiment 1.A using balanced class distribution. The bold font refers to highest accuracy.

Methods	CNN	Pseudo	LP (K-NN)	LP (RBF)	LS (K-NN)	LS (RBF)
Accuracy	81.5 %	81.9 %	82.8 %	82.7%	**83.0** %	82.7%

Table 5 and Figure 19 demonstrates the accuracy of applying Experiment 1.B. Figure 19 shows that increasing the size of the support set improves the accuracy of classification in the semi-supervised methods, which means using more samples in the support set helpful to increase the performance. Using 500 samples, the highest accuracy achieved by LP (RBF) and the worst accuracy occurred when LS (K-NN) was used. For 1000, the overall accuracy improved, and the highest accuracy was achieved by the LS (K-NN) and the worst accuracy occurred when the Pseudo label was used.

Table 5. Evaluation of the impact of different support set sizes with a fixed size of training set in Experiment 1.B. The bold font refers to highest accuracy.

Support set size	500	1000	1500
CNN	81.4%	81.5 %	81.5 %
Pseudo	80.9 %	81.3 %	81.8 %
LP(K-NN)	80.7 %	82.0 %	82.8 %
LP(RBF)	**81.7** %	82.06 %	82.7%
LS(K-NN)	80.6 %	**82.2** %	**83.0** %
LS(RBF)	81.6 %	82.0 %	82.4%

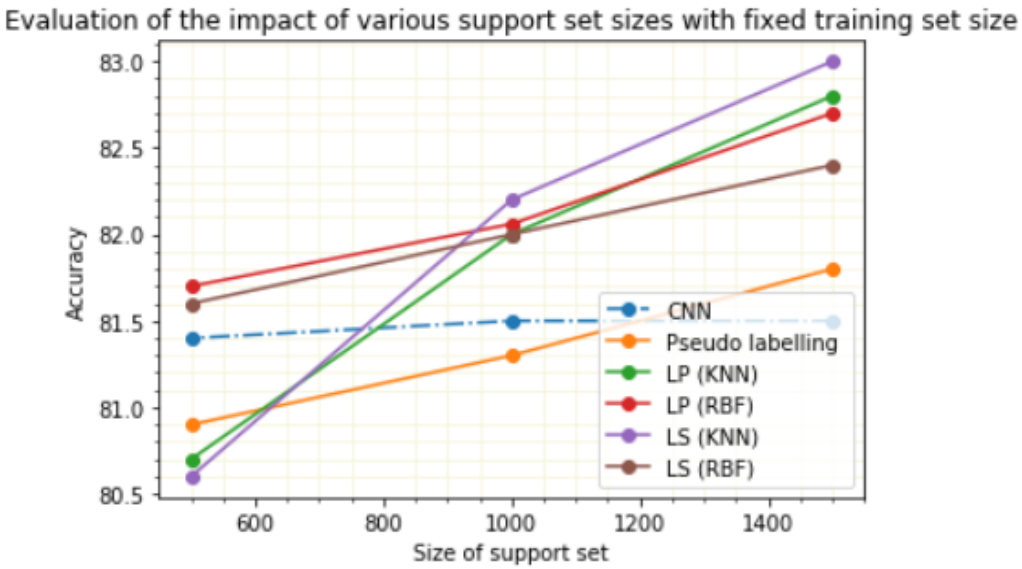

Figure 19. Evaluation of the impact of the different support set sizes with a fixed training set size in experiment 1.B. The bold font refers to highest accuracy.

Table 6 and Figure 20 demonstrates the accuracy of applying Experiment 1.C. The result shows that increasing the size of the training set improves the accuracy of the classification of all methods. In particular, the LP algorithm was more sensitive to the size of the training data, particularly when using the RBF kernel. The reduction occurs mainly

because the accuracy does not take into account the unequal distribution of the class and only demonstrates high accuracy when most of the data is of a similar class.

Table 6. Evaluation the effect of different sizes of the training set with a fixed size of the support set in Experiment 1.C. The bold font refers to highest accuracy.

Training set size	500	1000	1500
CNN	80.3 %	81.1 %	81.5 %
Pseudo	**80.9** %	81.5 %	81.8 %
LP(K-NN)	78.0 %	78.3 %	82.8 %
LP(RBF)	68.4 %	71.5%	82.7%
LS(K-NN)	81.0 %	**82.2** %	**83.0** %
LS(RBF)	76.0 %	80.19 %	82.4 %

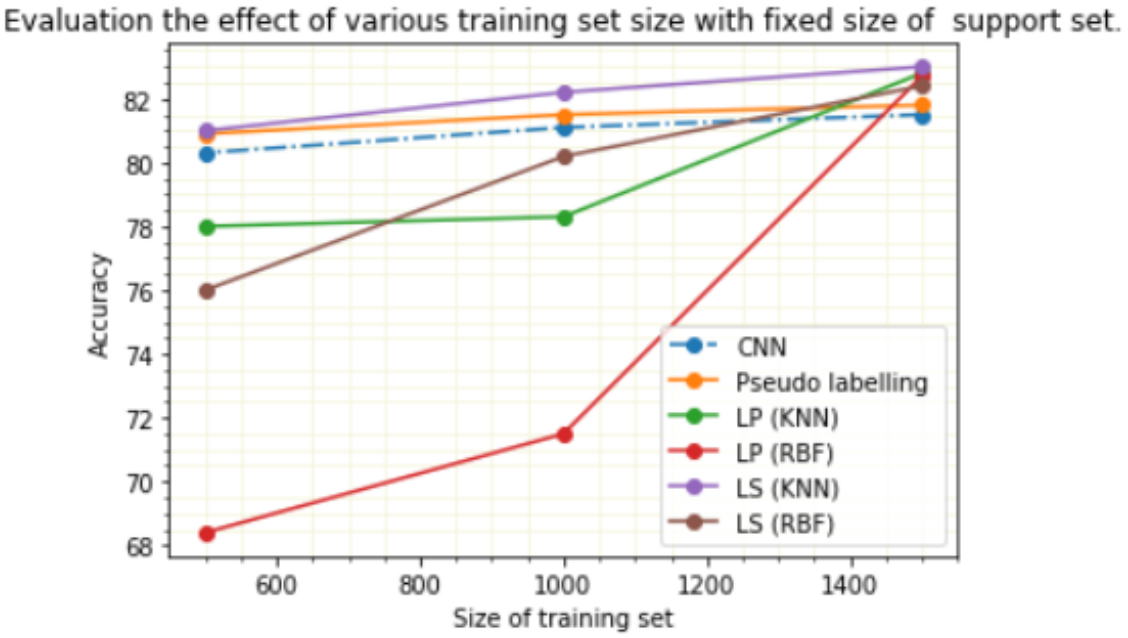

Figure 20. Evaluation of the impact of different sizes of the training set with a fixed support set size in Experiment 1.C. The bold font refers to highest accuracy.

Table 7 shows the result of implemented Experiment 2 using imbalanced five classes. Experiment 2 achieved a high performance for all methods by using imbalanced large dataset sizes, and the semi-supervised method showed improvement in accuracy compared to the CNN-supervised method. The highest accuracy achieved by LP (RBF) and the worst accuracy occurred when CNN was used.

Table 7. Comparison between various semi-supervised methods in Experiment 2. The bold font refers to highest accuracy.

Methods	CNN	Pseudo	LP (K-NN)	LP (RBF)	LS (K-NN)	LS (RBF)
Accuracy	89.3 %	89.9 %	90.3 %	**90.5**%	90.1%	90%

Table 8 shows the result of implemented Experiment 3 using balanced classes and the effect of data augmentation processes. The results show that the semi-supervised methods failed to achieve high accuracy when the number of samples per class was small. The reduction occurred mainly because the CNN model struggled to find the right features for the dataset. Then data augmentation was applied to increase the datset, and after the augmentation process, the accuracy improves by approximately 10% points than before.

Table 8. Accuracy of the different methods used before and after data augmentation processes in Experiment 3. The bold font refers to highest accuracy.

Methods	Accuracy before augmentation	Accuracy after augmentation
CNN	**45.2** %	**55.7** %
Pseudo	38.2 %	49.3 %
LP(K-NN)	30.3 %	39.0 %
LP(RBF)	30.0 %	39.5 %
LS(K-NN)	34.5 %	42.1 %
LS(RBF)	33.06 %	43.8 %

The result of Experiment 4 shows in Table 9 for all dataset and the accuracy before and after augmentation. Pseudo labeling had a higher accuracy between the rest of all methods used. Accuracy is declined because in classes that contain a small amount of data. The model over-fits training data that reduces performance. Also, some of the semi-supervised learning methods failed to achieve more accuracy than CNN, when CNN struggled to find the right dataset features, so cluster assumptions did not work.

Table 9. Accuracy of the different methods used before and after data augmentation in Experiment 3. The bold font refers to highest accuracy.

Methods	Accuracy before augmentation	Accuracy after augmentation
CNN	48.5 %	54.0%
Pseudo	49.0 %	55.2 %
LP(K-NN)	46.9 %	51.8%
LP(RBF)	47.0 %	52.3 %
LS(K-NN)	47.05 %	50.1 %
LS(RBF)	**50.0** %	**57.4** %

6 DISCUSSION

6.1 Current study

The research focused on classification plankton images using semi-supervised methods and comparing the result with the supervised baseline method CNN. The research was carried out on Label Propagation, Label Spreading, and Pseudo labeling semi-supervised methods. The methods were compared to CNN supervised method.

Experiment 1 showed that using balanced class distribution, the highest accuracy achieved is 83% by using LS with the RBF kernel. The LP (K-NN) came next with an accuracy of 82.8%. The LS (RBF) and LP (RBF) achieved an accuracy of 82.7% and Pseudo labeling method's accuracy was 81.9%. CNN had the least accuracy of 81.5%. The semi-supervised methods improved the accuracy comparing to the CNN method. . Reducing the size of training data and supporting data reduces accuracy. Among the four methods, the LP algorithm was more sensitive to the size of the training set, particularly when using the RBF kernel. The reason of high sensitivity because the accuracy does not take into consideration the unequal distribution of the class, and only shows high accuracy if most of the data had a similar class.

In LP and LS algorithms, parameters such as maximum iteration, α and γ, and a number of neighbors have a direct effect on accuracy. Choosing the correct parameters, depending on the details of the data, is a key reason for improving the accuracy of the classification. The number of maximum iterations needed to perform the experiments depends mainly on the value of the parameters, in particular the number of neighbors and γ. For K-NN kernel, a high number of neighbors required a high maximum iteration while in the RBF kernel a high number of γ required a high maximum iteration. Also, the maximum number of iterations depends on the features that have been transferred to the function.

The LS and LP algorithms can reach the limit of iteration before they have finished classifying all data. In this case, the accuracy does not improve even if the number of training data increases. In general, the RBF kernel needs more iterations than the K-NN kernel.

Experiment 2 used imbalanced class distribution. The result showed that the highest accuracy was achieved by using LP (RBF) which was 90.5%, followed by the LP (K-NN) with an accuracy of 90.3 %. The LS (K-NN), LS (RBF), and pseudo achieved accuracies 90.1%, 90.0%, and 89.9 respectively. The worst accuracy was with the CNN method

with an accuracy of 89.3%. The experiment showed an improvement in the classification results compared to experiment 1 using the same classes with more data.

Experiment 3 implemented for balanced class distribution. The result was between 30.0-45.2% for all methods. Sharp decrease in accuracy occurred because the number of samples per class was not sufficient to train the CNN used for feature extraction. The highest accuracy was achieved using the CNN method which was 45.2% followed by Pseudo labeling, was 38.2% and 30.3%, 30.0%, 34.5%, and 33.06% for LP (K-NN), LP (RBF), LS (K-NN), and LS (RBF) respectively. The semi-supervised methods did not work well because of CNN struggled to find the right features for the dataset. These features were not good enough to separate the classes, so the cluster assumption did not work. Data augmentation was applied to increase the amount of data per class, and after the augmentation process the accuracy increase by approximately 10% points.

Experiment 4 was applied for an imbalanced dataset using all available data. The result showed that the highest accuracy achieved by LS (RBF) which was 50.0% followed by 49.0 for Pseudo labeling and 48.5%, 47.0%, 47.05%, and 46.9% for CNN, LS (KNN), LP (KNN) and LP (RBF). The implemented methods faced problems that the samples in the classes were not equal which led to an imbalanced class problem, and also more than half of the dataset contains less than 100 samples per class which has led to over-fitting problems. This problem was solved by increasing the classes using the data augmentation process which helps to improve the accuracy. Another challenge was the size of the image that differs from one image to another in the dataset which was resolved by resizing the image to the same target size.

6.2 Future work

Additional work can be done by applying a deep CNN method such as Alexnet to enhance the extracted features. It can also be performed to improve the selection process for the LP and LS parameters in order to achieve better results. Focusing more on the data augmentation process can help improve the outcome for the dataset with a small amount of samples. Applying deep semi-supervised learning methods across a wide range of plankton datasets, then compare the outcome to different methods. Also, applying the methods proposed to various datasets and comparing the results between the datasets.

7 CONCLUSION

The research focused on studying the classification of plankton images using semi-supervised methods and the result compared to the CNN supervised baseline method. The dataset used was collected by the Marine Research Center of the Finnish Environment Institute. The dataset contained 86 different classes, with varying numbers of images per class. The LP, LS, and Pseudo labeling semi-supervised methods were used in this study and compared to fully supervised CNN. The images features were extracted from the images using the CNN architecture.

Increasing the size of the support set or training set improved the accuracy of the classification. The data augmentation was applied and the result showed an increase in accuracy by approximately 10% points compared to the original dataset accuracy.

The study found that semi-supervised learning was more accurate than the CNN-supervised method, when the available dataset was good at extracting the correct feature, and the amount of data available to train the model was sufficient. At the same time, some of the semi-supervised learning methods failed to achieved accuracy better than CNN, when CNN struggled to find the right dataset features, then the cluster assumption did not work.

REFERENCES

[1] Diane K Stoecker. Conceptual models of mixotrophy in planktonic protists and some ecological and evolutionary implications. *European Journal of Protistology*, 34(3):281–290, 1998.

[2] Iain Suthers, David Rissik, and Anthony Richardson. *Plankton: A guide to their ecology and monitoring for water quality*. Commonwealth Scientific and Industrial Research Organisation publishing, 2019.

[3] 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. *International Council for the Exploration of the Sea Journal of Marine Science*, pages 2–7, 2019.

[4] Heidi M Sosik, Robert J Olson, Michael G Neubert, Alexi Shalapyonok, and Andrew R Solow. Growth rates of coastal phytoplankton from time-series measurements with a submersible flow cytometer. *Limnology and Oceanography*, 48(5):1756–1765, 2003.

[5] Robert J Olson and Heidi M Sosik. A submersible imaging-in-flow instrument to analyze nano-and microplankton: Imaging flowcytobot. *Limnology and Oceanography: Methods*, 5(6):195–203, 2007.

[6] Eric C Orenstein, Oscar Beijbom, Emily E Peacock, and Heidi M Sosik. Whoi-plankton-a large scale fine grained visual recognition benchmark dataset for plankton classification. *arXiv preprint*.

[7] Osku GrOnberg. Plankton recognition from imaging flow cytometer data using convolutional neural networks. *Master thesis, LUT University*.

[8] S Rasoul Safavian and David Landgrebe. A survey of decision tree classifier methodology. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(3):660–674, 1991.

[9] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002.

[10] Thierry Denoeux. A neural network classifier based on dempster-shafer theory. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 30(2):131–150, 2000.

[11] Xiaohua Zhai, Avital Oliver, Alexander Kolesnikov, and Lucas Beyer. Self-supervised semi-supervised learning. *arXiv preprint*, 2019.

[12] Akinori Fujino, Naonori Ueda, and Kazumi Saito. A hybrid generative/discriminative approach to semi-supervised classifier design. In *Association for the Advancement of Artificial Intelligence*, pages 764–769, 2005.

[13] Mark C Benfield, Philippe Grosjean, Phil F Culverhouse, Xabier Irigoien, Michael E Sieracki, Angel Lopez-Urrutia, Hans G Dam, Qiao Hu, Cabell S Davis, Allen Hansen, et al. Rapid: research on automated plankton identification. *Oceanography*, 20(2):172–187, 2007.

[14] Peter B Ortner, Shailer R Cummings, R Paul Aftring, and Harold E Edgerton. Silhouette photography of oceanic zooplankton. *Nature*, 277(5691):50–51, 1979.

[15] Mark C Benfield, Andone C Lavery, Peter H Wiebe, Charles H Greene, Timothy K Stanton, and Nancy J Copley. Distributions of physonect siphonulae in the gulf of maine and their potential as important sources of acoustic scattering. *Canadian Journal of Fisheries and Aquatic Sciences*, 60(7):759–772, 2003.

[16] Cabell S Davis, Qiao Hu, Scott M Gallager, Xiaoou Tang, and Carin J Ashjian. Real-time observation of taxa-specific plankton distributions: an optical sampling method. *Marine Ecology Progress Series*, 284:77–96, 2004.

[17] Andrew Remsen, Thomas L Hopkins, and Scott Samson. What you see is not what you catch: a comparison of concurrently collected net, optical plankton counter, and shadowed image particle profiling evaluation recorder data from the northeast gulf of mexico. *Deep Sea Research Part I: Oceanographic Research Papers*, 51(1):129–151, 2004.

[18] Christian K Sieracki, Michael E Sieracki, and Charles S Yentsch. An imaging-in-flow system for automated analysis of marine microplankton. *Marine Ecology Progress Series*, 168:285–296, 1998.

[19] F Colas, M Tardivel, J Perchoc, M Lunven, B Forest, G Guyader, MM Danielou, S Le Mestre, P Bourriau, E Antajan, et al. The zoocam, a new in-flow imaging system for fast onboard counting, sizing and classification of fish eggs and metazooplankton. *Progress in Oceanography*, 166:54–65, 2018.

[20] Loris Nanni, Stefano Ghidoni, and Sheryl Brahnam. Handcrafted vs. non-handcrafted features for computer vision classification. *Pattern Recognition*, 71:158–172, 2017.

[21] Lu Jin, Shenghua Gao, Zechao Li, and Jinhui Tang. Hand-crafted features or machine learnt features? together they improve rgb-d object recognition. In *IEEE International Symposium on Multimedia*, pages 311–319. IEEE, 2014.

[22] Dmitriy Fradkin and Ilya Muchnik. Support vector machines for classification. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 70:13–20, 2006.

[23] Kai-Bo Duan and S Sathiya Keerthi. Which is the best multiclass svm method? an empirical study. In *International Workshop on Multiple Classifier Systems*, pages 278–285. Springer, 2005.

[24] A. Mir and Jalal A. Nasiri. Lighttwinsvm: A simple and fast implementation of standard twin support vector machine classifier. *Journal of Open Source Software*, page 1252, 03 2019.

[25] Matt W Gardner and SR Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14-15):2627–2636, 1998.

[26] Ghazi Al-Naymat, Mouhammd Al-kasassbeh, Nosaiba Abu-Samhadanh, and Sherif Sakr. Classification of voip and non-voip traffic using machine learning approaches. *Journal of Theoretical and Applied Information Technology*, 3192, 11 2016.

[27] Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

[28] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[29] Tin Kam Ho. Random decision forests. In *3rd International Conference on Document Analysis and Recognition*, volume 1, pages 278–282. IEEE, 1995.

[30] Houtao Deng and George Runger. Gene selection with guided regularized random forest. *Pattern Recognition*, 46(12):3483–3489, 2013.

[31] Padraig Cunningham and Sarah Jane Delany. k-nearest neighbour classifiers. *arXiv preprint*, 2020.

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

[33] Gaby Gorsky, Mark D Ohman, Marc Picheral, Stéphane Gasparini, Lars Stemmann, Jean-Baptiste Romagnan, Alison Cawood, Stéphane Pesant, Carmen García-Comas, and Franck Prejger. Digital zooplankton image analysis using the zooscan integrated system. *Journal of Plankton Research*, 32(3):285–303, 2010.

[34] Tong Luo, Kurt Kramer, Dmitry B Goldgof, Lawrence O Hall, Scott Samson, Andrew Remsen, and Thomas Hopkins. Recognizing plankton images from the shadow image particle profiling evaluation recorder. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 34(4):1753–1762, 2004.

[35] Robert Gray. Vector quantization. *IEEE Assp Magazine*, 1(2):4–29, 1984.

[36] Andrhé Bigand Alain Lefebvre Thi-Thu-Hong Phan, Émilie Poisson Caillault. Comparative study on supervised learning methods for identifying phytoplankton species. In *IEEE Sixth International Conference on Communications and Electronics (ICCE)*, pages 283–288, 2016.

[37] 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.

[38] Akinori Hidaka and Takio Kurita. Consecutive dimensionality reduction by canonical correlation analysis for visualization of convolutional neural networks. In *ISCIE International Symposium on Stochastic Systems Theory and its Applications*, volume 2017, pages 160–161, 2017.

[39] Andrej Karpathy et al. Cs231n convolutional neural networks for visual recognition. *Neural Networks*, 1:1–2, 2016.

[40] Dominik Scherer, Andreas Müller, and Sven Behnke. Evaluation of pooling operations in convolutional architectures for object recognition. In *International Conference on Artificial Neural Networks*, pages 92–101. Springer, 2010.

[41] Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. *arXiv preprint*.

[42] Anibal Pedraza, Gloria Bueno, Oscar Deniz, Gabriel Cristóbal, Saúl Blanco, and María Borrego-Ramos. Automated diatom classification (part b): a deep learning approach. *Applied Sciences*, 7(5):460, 2017.

[43] Wenzhi Zhao and Shihong Du. Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach. *IEEE Transactions on Geoscience and Remote Sensing*, 54(8):4548–4549, 2016.

[44] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[45] Sebastian Avalos and Julian M Ortiz. Geological modeling using a recursive convolutional neural networks approach. *arXiv preprint*.

[46] Yann LeCun, Patrick Haffner, Leon Bottou, and Yoshua Bengio. Object recognition with gradient based learning. In *Shape, contour and grouping in computer vision*, pages 319–345. Springer, 1999.

[47] An Zheng and Mingyang Wang. Convolutional neural networksbased plankton image classification system. 2015.

[48] Hansang Lee, Minseok Park, and Junmo Kim. Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning. In *IEEE International Conference on Image Processing (ICIP)*, pages 3713–3717, 2016.

[49] Iago Correa, Paulo Drews, Silvia Botelho, Marcio Silva de Souza, et al. Deep learning for microalgae classification. In *IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 20–25, 2017.

[50] Jialun Dai, Ruchen Wang, Haiyong Zheng, Guangrong Ji, and Xiaoyan Qiao. Zooplanktonet: deep convolutional network for zooplankton classification. In *OCEANS 2016-Shanghai*, pages 1–6. IEEE, 2016.

[51] Hussein A Al-Barazanchi, Abhishek Verma, and Shawn X Wang. Intelligent plankton image classification with deep learning. *International Journal of Computational Vision and Robotics*, 8(6):561–571, 2018.

[52] Erik Bochinski, Ghassen Bacha, Volker Eiselein, Tim JW Walles, Jens C Nejstgaard, and Thomas Sikora. Deep active learning for in situ plankton classification. In *International Conference on Pattern Recognition*, pages 5–15, 2018.

[53] V Jothi Prakash and Dr LM Nithya. A survey on semi-supervised learning techniques. *arXiv preprint*, 2014.

[54] Zoubin Ghahramani. Unsupervised learning. In *Summer School on Machine Learning*, pages 72–112. Springer, 2003.

[55] Erik G Learned-Miller. Introduction to supervised learning. *I: Department of Computer Science, University of Massachusetts*, 2014.

[56] Massih-Reza Amini and Nicolas Usunier. *Learning with Partially Labeled and Interdependent Data*. Springer, 2015.

[57] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Lectures on Artificial Intelligence and Machine Learning*, 3, no. 1:1–8, 2009.

[58] Pedro HM Braga and Hansenclever F Bassani. A semi-supervised self-organizing map for clustering and classification. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

[59] Olivier Chapelle, Vikas Sindhwani, and Sathiya S Keerthi. Optimization techniques for semi-supervised support vector machines. *Journal of Machine Learning Research*, 9(Feb):203–233, 2008.

[60] Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, International Conference on Machine Learning*, volume 3, pages 2–4, 2013.

[61] Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Eleventh Conference on Computational Learning Theory*, pages 92–100, 1998.

[62] Wei Wang and Zhi-Hua Zhou. Analyzing co-training style algorithms. In *European Conference on Machine Learning*, pages 454–465. Springer, 2007.

[63] Shrutika S Sawant and Manoharan Prabukumar. A review on graph-based semi-supervised learning methods for hyperspectral image classification. *The Egyptian Journal of Remote Sensing and Space Science*, 2018.

[64] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014.

[65] Hamed Hassanzadeh, Mahnoosh Kholghi, Anthony Nguyen, and Kevin Chu. Clinical document classification using labeled and unlabeled data across hospitals. In *AMIA Annual Symposium Proceedings*, volume 2018, page 545. American Medical Informatics Association, 2018.

[66] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Advances in Neural Information Processing Systems*, pages 3546–3554, 2015.

[67] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. In *Advances in Neural Information Processing Systems*, pages 529–536, 2005.

[68] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. Learning from labeled and unlabeled data on a directed graph. In *International Conference on Machine Learning*, pages 1036–1043, 2005.

[69] Amarnag Subramanya and Partha Pratim Talukdar. Graph-based semi-supervised learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, pages 1–125, 2014.

[70] Mark Nixon and Alberto Aguado. *Feature extraction and image processing for computer vision, 4th edition*. Academic Press, 2019.

[71] Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3642–3649. IEEE, 2012.

[72] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in Neural Information Processing Systems*, pages 3123–3131, 2015.

[73] Mark Needham and Amy E Hodler. *Graph Algorithms: Practical Examples in Apache Spark and Neo4j*. O'Reilly Media, 2019.

[74] Zhu Xiaojin and Ghahramani Zoubin. Learning from labeled and unlabeled data with label propagation. *Tech. Rep., Technical Report CMU-CALD-02–107, Carnegie Mellon University*, 2002.

[75] Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. *Label Propagation and Quadratic Criterion*. Massachusetts Institute of Technology Press, semi-supervised learning edition, 2006.

[76] Xin Liu and Tsuyoshi Murata. Advanced modularity-specialized label propagation algorithm for detecting communities in networks. *Physica A: Statistical Mechanics and its Applications*, 389(7):1493–1500, 2010.

[77] Bonnie Hardin and Upulee Kanewala. Using semi-supervised learning for predicting metamorphic relations. In *IEEE/ACM 3rd International Workshop on Metamorphic Testing (MET)*, pages 14–17. IEEE, 2018.

[78] Dengyong Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in Neural Information Processing Systems*, pages 321–328, 2004.

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

[80] Sebastian Raschka. *Python machine learning*. Packt Publishing Ltd, 2015.

[81] Gongwen Xu, Zhiqi Sang, and Zhijun Zhang. Cross-media retrieval based on pseudo-label learning and semantic consistency algorithm. *International Journal of Performability Engineering*, 14(9):2219–2229, 2018.

[82] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. *arXiv preprint*, 2019.

[83] Yan Huang, Jingsong Xu, Qiang Wu, Zhedong Zheng, Zhaoxiang Zhang, and Jian Zhang. Multi-pseudo regularized label for generated data in person re-identification. *IEEE Transactions on Image Processing*, 28(3):1391–1403, 2018.

[84] Guodong Ding, Shanshan Zhang, Salman Khan, Zhenmin Tang, Jian Zhang, and Fatih Porikli. Feature affinity-based pseudo labeling for semi-supervised person re-identification. *IEEE Transactions on Multimedia*, 21(11):2891–2902, 2019.

[85] FastVision project, Lappeenranta-Lahti University of Technology (LUT), Finland. http://www2.it.lut.fi/project/fastvision/index.shtml, 2020. [Online; accessed June, 11, 2020].

[86] Spyros Makridakis. Accuracy measures: theoretical and practical concerns. *International Journal of Forecasting*, 9(4):527–529, 1993.