O. SIDDIK

AUTOMATIC SPIRULINA DETECTION USING IMAGE PROCESSING TECHNIQUES

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

OTHMAN SIDDIK

DOCTOR OF PHILOSOPHY THESIS
IN
MODELING AND DESIGN OF ENGINEERING SYSTEMS
(MAIN FIELD OF STUDY: COMPUTER ENGINEERING)

JULY 2019

ATILIM UNIVERSITY 2019

AUTOMATIC SPIRULINA DETECTION USING IMAGE PROCESSING TECHNIQUES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

BY

OTHMAN SIDDIK

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
MODELING AND DESIGN OF ENGINEERING SYSTEMS
(MAIN FIELD OF STUDY: COMPUTER ENGINEERING)

JULY, 2019

Approval of the Graduate School of Natural and Applied Sciences, Atilim University.

_____

Prof. Dr. Ali KARA
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of **Doctor of Philosophy in Modeling and Design of Engineering Systems (MODES) Atilim University.**

_____

Assoc. Prof. Dr. Ender KESKİNKILIÇ
Head of Department

This is to certify that we have read the thesis AUTOMATIC SPIRULINA DETECTION USING IMAGE PROCESSING TECHNIQUES submitted by OTHMAN SIDDIK and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Doctor of Philosophy.

_____

Asst. Prof. Dr. Atila BOSTAN
Supervisor

Examining Committee Members:

Assoc. Prof. Dr.Murat KOYUNCU
Information Systems Engineering, Atılım University _____

Asst. Prof. Dr. Atila BOSTAN
Computer engineering, Atilim University _____

Asst. Prof. Dr. Hasan Umut AKIN
Industrial Engineering, THK University _____

Asst. Prof. Dr. Erol ÖZÇELİK
Department of Psychology, Cankaya University _____

Asst. Prof. Dr. Güzin TİRKEŞ
Computer engineering, Atilim University _____

Date: 09.07.2019

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : Othman SIDDIK

Signature :

ABSTRACT

AUTOMATIC SPIRULINA DETECTION USING IMAGE PROCESSING TECHNIQUES

SIDDIK, Othman

PHD, Department of Modeling and Design of Engineering Systems

Supervisor: Atila, BOSTAN

July 2019, 67 pages

In this thesis, a study on automatic detection of spirulina is presented. Spirulina is an algae microorganism with 4 species which are quite useful for the determination and monitoring of water quality. Thesis contribution is to develop an automatic process for helping the diagnosis Spirulina in water, most of the Spirulina can be diagnosed by the size and shape from microscopic images, all algae detection that has to be diagnosed in a fast and accurate way is very critical for the water quality, manual methods are used to detect spirulina. This can give rise to inaccurate results. It is also very tedious effort to detect algae within water microscopic images. Automatic detection of spirulina is a challenging task due to factors such as change in size and shape with climatic changes, growth periods and water contamination. Nowadays, the automated detection of spirulina is one of the most fervent topics in applied biology. On the other hand, Deep-Learning and Convolutional Neural Networks (CNN) is yielding better results and is a judiciously used technique for image classification and for a variety of problems. This thesis introduces CNN into the automated spirulina detection problem in order to demonstrate whether it would succeed in solving the spirulina detection problem. A comprehensive dataset was specifically prepared using an artificial image generation method out of original images that are collected from rivers and lakes in Turkey. In this study, a spirulina image data set was prepared using a customized

technique for artificial image generation. Consequently, a dataset covering different illumination conditions was computationally augmented to 1000 sample images. Original images were collected from rivers and lakes in Turkey. In this thesis, the background to the spirulina detection problem, the methodology used in the study and the results of image processing and feature extraction methods to locate and extract spirulina in a microscopic image are reported. Initially, the RGB image format with morphological operations were employed to detect spirulina in a microscopic image. As a result with a rate of 84% accuracy detection was observed. Afterwards, three different methods were experimented with for comparison purposes. The methods and their relative detection success rates were observed as follows: SURF 63%, FAST feature detection 67%, CNN 99% result accuracy rate, consequently, some future work is also suggested to improve the study further. In this thesis, we introduced CNNs into the automated spirulina detection problem. A CNN method used to solve 4 class spirulina detection problem. Observed results were discussed and compared with those of previous studies. To the best of our knowledge and survey results on the literature, this is the first study to employ CNNs in the automated spirulina detection problem.

Keywords: Spirulina, Automatic detection, Morphological feature, Feature analysis, Image Acquisition, Deep Learning, Convolutional Neural Networks, Classification, Spirulina detection.

ÖZ

GÖRÜNTÜ İŞLEME YÖNTEMLERİ KULLANARAK OTOMATİK SPIRULİNA TESPİTİ

SIDDIK, Othman
Doktora, Mühendislik Sistemlerinin Modellenmesi ve Tasarımı
Tez Yöneticisi: Atila, BOSTAN

Temmuz 2018, 67 sayfa

Bu tezde Spirulina'nın otomatik tespiti üzerine bir çalışma sunulmuştur. Spirulina 4 türe sahip bir tür alg mikroorganizmadır ve su kalitesinin tespitinde, takibinde ve gözlemlenmesinde oldukça yararlıdır. Bu tezin katkısı, Spirulina'nın sudaki teşhisine yardımcı olmak için otomatik bir sistem geliştirmek ve bu yöntemin uygulanabilirliğini göstermek olmuştur. Mikroskobik görüntülerde Spirulina'nın hızlı ve doğru bir şekilde teşhis edilebilmesi diğer tüm alg türlerinin tespiti için de çok önemlidir. Su kalitesinin belirlenmesinde, Spirulina tür ve miktarını tespit etmek için manuel yöntemler kullanılmaktadır. Ancak insan dikkat ve hatasına bağlı olan bu yöntem yanlış sonuçlara yol açabilmektedir. Mikroskobik su görüntülerinde algleri uzmanlar aracılığı ile, tespit etmek oldukça zor ve yoğun çaba gerektirmektedir.

Spirulina'nın otomatik tespiti boyut, şekili, iklimsel değişiklikleri, büyüme periyotları ve su kirliliği gibi sebepler nedeniyle zor bir işlemdir. Günümüzde spirulina'nın otomatik olarak tespiti, uygulamalı biyoloji alanında öncelikle ihtiyaç duyulan konulardan biridir. Diğer taraftan, Derin-Öğrenme ve Evrişimli Sinirsel Ağlar (CNN) sayısal görüntülerde nesne tanıma işlemlemlerinde daha iyi sonuçlar sağlamakta ve görüntü sınıflandırma problemlerinde yaygın olarak kullanılmaktadır. Bu tezde, mikroskobik görüntülerde Spirulina tespit etme problemini çözmek için başarılı olup olmadığını göstermek amacıyla, CNN yöntemi de test edilmiştir. Türkiye'deki nehir ve göllerden toplanan orijinal görüntüler arasından yapay görüntü yaratma yöntemi

kullanarak kapsamlı bir veri kümesi hazırlanmıştır. Orjinal mikroskobik görüntüler yapay görüntü yaratma yöntemi kullanılarak çoğaltılmış ve bu çalışma için özel bir set oluşturulmuştur. Sonuç olarak, görüntü seti, farklı aydınlatma koşullarını içerecek şekilde 1000 görüntüye çıkartılmıştır. Orijinal görüntüler Türkiye'deki nehir ve göllerden toplanmıştır. Bu tezde, bir mikroskobik-görüntüde Spirulina'nın tespitinde görüntü işleme ve özellik çıkarma sonuçları raporlanmıştır.

İlk olarak, Spirulina'nun mikroskobik-görüntüde tespit edilmesi için RGB görüntü formatı üzerinde morfolojik işlemler kullanılmıştır. Bu yöntemle %84 oranında tespit-başarısı gözlemlenmiştir. Daha sonra, karşılaştırma amacıyla üç farklı yöntem denenmiştir. Bu yöntemler ve ilgili tespit-başarı oranları ise; SURF 63%, FAST 67%, CNN 99% doğruluk oranı olarak belirlenmiştir. Doğal olarak, gelecekte yapılacak çalışmaların, bu çalışmaları daha ileri götüreceği görülebilmektedir. Bu tez çalışmasında, mikroskobik görüntülerde Spirulinanın otomatik olarak tespiti için CNN ilk defa denenmiştir. Spirulinanın dört farklı türünün tespiti için CNN yöntemi de kullanılmıştır. Gözlemlenen, sonuçlar karşılaştırmalı ve ayrıntılı olarak tartışılmıştır. Bilgimiz dahilindeki literatür ve araştırma sonuçları uyarınca bu çalışma, mikroskobik görüntülerde otomatik Spirulina tespiti probleminde CNN yöntemi kullanılan ilk çalışmadır.

Anahtar Kelimeler: Spirulina, Otomatik tespit, Morfolojik özellik, Özellik analizi, Görüntü elde etme, Derin öğrenme, Evrişimli sinirsel ağlar, Sınıflandırma, Spirulina tespiti.

ACKNOWLEDGMENTS

I would prefer to specific my special thanks to my supervisor Dr. Atila BOSATN gave me the nice chance to try and do this glorious project on the subject Spirulina detection using image processing in water, that conjointly helped me in doing a great deal of analysis and that I came to grasp regarding such a big number of new things I'm extremely appreciative to them.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

In this thesis, an automatic system for spirulina detection and classification is presented. Much research has been conducted in this field; however, there is still a considerable requirement in classification and detection of Spirulina in microscopic images. Spirulina is a type of algae microorganism with 4 species which are quite useful for water quality determination and monitoring. Spirulina is used in Green Oil, food resources and plastic manufacturing. The automatic identification of algae is substantially difficult due to a variety of reasons such as changes in size and shape, climatic changes, growth periods and contamination [1].

It is exceptionally troublesome and difficult to detect spirulina, some species of which are identical in shape, as expounded by [2]. Two hundred thousand diatom species have been determined to exist, albeit only 50% of which have been identified. Nevertheless, it appears to be appropriate to develop an automated classifier system to recognize and evaluate various examples of spirulina. This would enable accelerating the procedure of water quality assurance. Similarly, this type of advancement may not be connected to various cells identified with diatoms but rather to microscopy as a target [3].

Common evidence of spirulina existence detection typically requires a specific level of analysis, which requires explicit experience and long term coaching. Moreover, the description of morphological spirulina and frustule (the series of connecting, siliceous links attached to a pipe) vary in different parts within the image. Some regular classes of spirulina that were imaged for many years are currently divided into entirely separate species. Aim to build up a programmed framework ready for use from the microscopic slides to detect spirulina, we produced the automatic spirulina identification system. The input to the system is an image taken from a magnifying lens of a microscope and the automated system is expected to identify and classify spirulina in it. Such an automation would certainly mitigate the need for human

experts, speed-up the detection and classification process and reduce the number of misevaluations. In this thesis, the prospects of computerized spirulina detection in microscopic digital images are studied.

Thesis contribution is to develop an automatically process for helping the diagnosis Spirulina in water, most of the Spirulina can be diagnosed by the size and shape from microscopic images, all algae detection that has to be diagnosed in a fast and accurate way is very critical for the water quality, manual methods are used to detect spirulina. This can give rise to inaccurate results. It is also very tedious effort to detect algae within a water microscopic images. The results are dependent on the situation of the expert doing the analysis. Therefore, a reliable automatic and fast way is vital for determining Spirulina in water. The main purpose of this thesis is to describe the development and to present the results Spirulina image based process to help for diagnosis. We live in a globalized world in which a few facilities are thought to be obtainable such as water or power supplies. In any case, this does not apply for everyone, so any kind of innovative venture that may help these advantages. Among them, water was exceptionally huge, since it was important for each and every role to be arranged for the prosperity of humans, such as agriculture or farming. Having this as a top priority, some kind of value power over this water must be performed. Most notably of the most encouraging strategies to satisfy this goal is to think about some microscopic organisms known commonly as diatoms. These organisms occur in gatherings of unicellular green algae. This is expressed in worldwide archives, such as the UN Directive "Framework of Water Policy," the purpose of which is to secure, improve, and control the nature of waterways, lakes, and oceans inside the European Union [3]. In this presented study, we experimented with four methods to detect spirulina in water. Furthermore, the composition and generation of artificial microscopic image datasets is also presented. Original microscopic water images were provided by Assoc. Prof. Dr. Cengiz Akköz form the Biology Department of Selçuk University. An image taken by an optical magnifying instrument is shown in Figure 1.1,

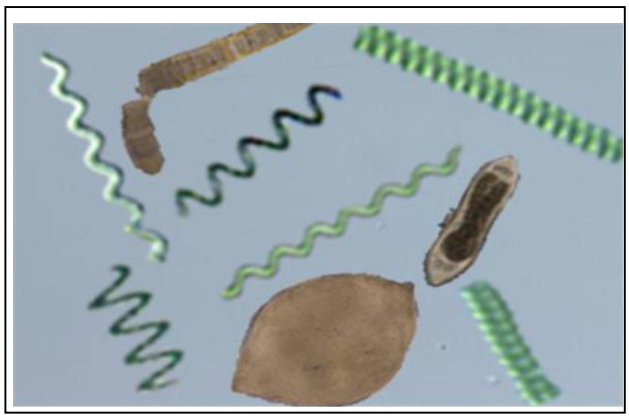


Figure 1.1 Spirulina in water

We had 60 original microscopic spirulina images and 40 additional images with other diatoms and non-spirulina objects. Since the number of genuine spirulina images was not satisfactory for the study, an artificial image generation was employed to generate supplementary spirulina images. The artificial image generation process generated 1000 images in a total of 100 original images which contained 60 original spirulina images. The method, exploited all possible combinations of spirulina types and non-spirulina objects. Based on color detection, the HSV (hue, saturation, value) format was used to extract the spirulina images. Initially, we counted the unique RGB (red, green, blue) color intensities of the pixels to find the color levels for the spirulina. Then we applied a plausible threshold onto the HSV color format and obtained a satisfactory detection rate in the ROI (region of interest). A morphological operation was also used to define the ROI. Speeded up robust features (SURF) and Features from Accelerated Segment Test (FAST) segmentation methods were also applied to detect any spirulina in water as all spirulina cells have a zigzag shape. Later in the study, we employed a method for corner detection using FAST and the Harries Corner methods. As mentioned in the international reference UN Water Policy Directive Framework [4],

Thesis Structure is present and organize as follows: Chapter 2 presents and introduces the related work of past research into spirulina characterization and detection as a general survey of grouping strategies. Chapter 3 presents the strategy of methodology advancement. In Chapter 4, the results are explained and validation tables are developed for each run of our system. Chapter 5 briefly presents thesis conclusion and future work.

CHAPTER 2

BACKGROUND

This chapter covers the most critical ideas and previous works related to diatom detection in digital images. Initially, the water quality and principles for identification of spirulina according to their shape and segments are presented in this chapter. From this point onward, a few ideas and works identified with diatom characterization are clarified. Finally, common characterization strategies (both customary picture handling and machine learning approaches) are detailed.

2.1 Literature Review

There are quite number of studies on object detection in microscopic images. Correspondingly, a significant proportion of them pertain to experimenting with algae detection. The topic of the challenge of automated detection and its achievements are described in this section. Edward Rosten and Tom Drummond proposed Diatom Identification and Classification in 2007. The system was designed for automated identification and classification of algae using digital microscopy. They used the artificial neural network method and were able to classify images of 23 different microalgae, Rosten and Drummond utilized the morphological operation dilate and erode and in segmentation operation, they were able to detect spirulina at 87% accuracy [5]. Yahui G. classified 12 types of circular diatoms, they reported success rates ranging between 43% and 86.5% accuracy [6] Bayer D. presented a Diatom Identification and Classification book in 2002. Their book contained RGB level formatted images in extracting diatoms at 83.4% accuracy [7].

Rosten and Drummond proposed the SIFT feature extraction for the detection of algae at detection speeds 300% greater than the feature detectors of Harris [8].

Bueno, G. and Deniz, O used feature detection to detect diatoms in water with an entire dataset of 24,000 segmented units which had 80 classes. The system reached 98%

efficiency of successful detection rate [9]. In 2012, a paper submitted by Dimitrovski et al. presented a 1093-image dataset with 55 classes which gave a rate of efficiency 96.17% [10]. In 2002, Du Buf, H. used the Bagging Tree classifier over 781 samples at 96.9% accuracy [11]. In 2003, Pappas, J. used the Multiple Discriminant Analysis classifier over 66 samples with 80.3% accuracy [12]. In 2016, Lai, Q. used the SVM classifier over 10,000 samples with 94.7% accuracy [13].

2.2 Image Processing

Image processing is the utilization of computer calculations to perform advanced picture handling on digital images. It allows for a more extensive scope of calculations so as to be connected to the information and a wider range of algorithms that are implemented to the input data. Moreover, it can avoid problems such as signal distortion noise reduction [14].

Digital image processing allows the use of substantially more unpredictable calculation, and subsequently, can offer both increasing and modern execution of basic tasks. Specifically, digital picture preparation is the main functional innovation for feature extraction, classification, pattern recognition, design, order and so on [15]. Moreover, a number of practical technologies are mentioned in the sections below.

2.3 Color Spaces

Color space is how colors are organized, which means that for testing the color space and the physical device, the color can be fixed analog or digital FIG. Color space can be defined by simply picking up some colors. For example, the PANTONE system simply takes a specific set of colors as a sample and then assigns a name and code to each color. It can also be based on rigorous mathematical definitions such as Adobe RGB and sRGB [16]. The color model is an abstract mathematical model that describes colors by a set of numbers; for example, RGB uses triples, and CMYK uses quadruples. If a color model has no mapping to the absolute color space, then it is a somewhat arbitrary color system with little connection to specific application requirements. If the establishment of a specific mapping function lies between the model and color of a specific reference color space, it will appear in the limited

“footprint” of the reference color space, also known as the color gamut. The color space is defined by the color model and the color gamut. For example, Adobe RGB and sRGB are based on RGB color models, but they are two different absolute color spaces [17].

The RGB (red, green and blue) color model is an additive color model that uses different levels of red, green, and blue primary colors which are added together to produce a wide variety of shades. (The shades of red, green and blue of the three primary colors cannot be synthesized with other monochromatic light) [Source Request]. The main purpose of the RGB color model is to detect, represent and display images in electronic systems, such as televisions, computers, and even in traditional photography. Before the electronic age, based on the human perception of color, the RGB color model had solid theoretical support. RGB is a device-dependent color space such that different devices detect and reproduce specific RGB values differently because color substances and their individual response levels to red, green and blue are manufactured, with the quotient varying, and even the same equipment varying in time. Based on any RGB color model is an additive color space belonging to the RGB color space [18]. The RGB color space of the RGB three primary colors of color definition can be defined whereby the respective color triangle generates other colors. The complete RGB color space definition also needs to give the white point chromaticity and a gamma correction curve [19]. The most common RGB color space at present is the sRGB color space.

The following is the numerical connection between HSI (hue, saturation, and intensity (HSI color space)) and RGB as showed in figure 2.1 [20]:

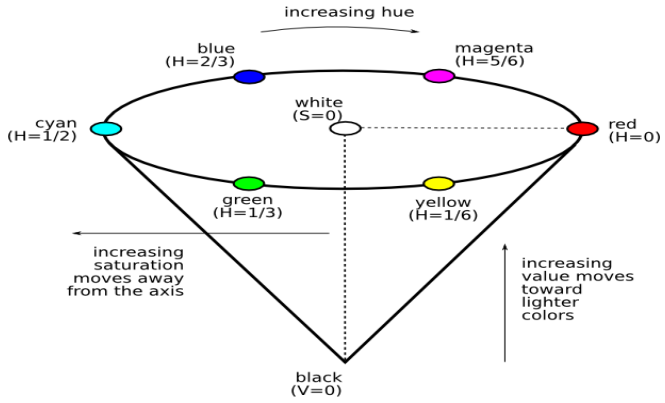


Figure 2.1 HSV color format [20]

Both HSL and HSV are representations of points in the RGB color model in a cylindrical coordinate system. These two representations attempt to be more intuitive than the geometry based on the Cartesian coordinate system of RGB.

HSL is hue, saturation, and brightness (alternatively, Hue, Saturation and *Lightness*). HSV is hue, saturation, and brightness (alternatively, Hue, Saturation, *Value*), also known as HSB, where B stands for brightness.

Hue (H) is the basic property of color, which is commonly referred to as the color name, such as red, yellow, and so on. Saturation (S) refers to the purity of the color. The high and low are level value of 0-100. Brightness (V) and brightness (L) values may range between values of 0% and100% [21]. Both HSL and HSV describe the color in a point in the cylindrical coordinate system. The central axis of the cylinder takes the value for black at the bottom to white at the top and gradations of gray between white and black. The angle around this axis corresponds to the hue. The distance to this axis corresponds to saturation and the height along this axis corresponds to brightness, hue or lightness.

These two representations are similar in purpose, but differ in method. Both are mathematically cylindrical, but HSV (hue, saturation, and lightness) can be conceptually considered to be an inverted cone of color (black dots at the lower vertex, white at the center of the upper bottom). HSL conceptually is a double cone and a sphere (white at the upper vertex, black at the lower vertex, and the center of the largest cross-section is half-grey). Note that although “hue” refers to an identical phenomenon in both HSL and HSV, their definitions of “saturation” differ significantly [22].

Because HSL and HSV are simple transformations of device-dependent RGB, the colors defined by the (h, s, l) or (h, s, v) triples depend on the particular red, green, and blue "additive primary colors" being used. Each unique RGB device is accompanied by a unique HSL and HSV space. However, the (h, s, l) or (h, s, v) triples become more explicit when constrained to a particular RGB.

The HSV model transformation of the three primary color modes, was founded in 1978 by Elvi Ray Smith as shown in figure 2.2 [23].

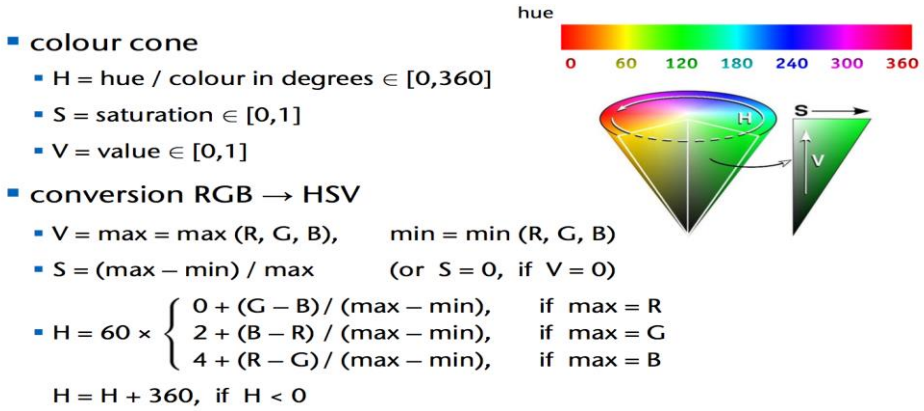


Figure 2.2 HSV transformations [23]

Grayscale image: In the computer field, a grayscale digital image is an image with only one sample color per pixel. Such images are typically displayed as the darkest black to the brightest white with gray between black and white although in theory the samples may be different shades of any color, or even different colors of different brightness. Grayscale images differ from black-and-white images. In the computer image field, black-and-white images are only black and white. A grayscale image has many levels of color depth between black and white. However, in the field of digital images, "black and white images" may also mean "grayscale images." For example, grayscale photos are often called "black and white photos." In some articles on digital images, a monochrome image is equivalent to a grayscale image, and in other articles, it is equivalent to a black-and-white image [24].

Grayscale images are often obtained by measuring the brightness of each pixel on a single electromagnetic spectrum, such as visible light [25].

Grayscale images for display are typically stored with a non-linear scale of 8 bits per sample pixel, which can have 256 shades of gray (8 bits is $2^8 = 256$). This precision barely avoids visible band distortion and is very easy to program. In the technical applications of medical and remote sensing imaging, more stages are often used to

make full use of the sensor accuracy of 10 or 12 bits per sample so as to avoid approximation errors in calculations. In such applications, 16 bits or 65,536 combinations (or 65,536 colors) are popular [27].

Binary images: A binary image is a digital image with only two possible values per pixel. Black and white, B&W, and monochrome images are often used to represent binary images, but they can also be used to represent any image with only one sample value per pixel, such as grayscale images [28].

Binary images often appear in digital image processing as image masks or in the results of image segmentation, binarization, and dithering. Some input and output devices, such as laser printers, fax machines, monochrome computer monitors, etc., can process binary images [29].

Binary images are often stored in bitmap format. A binary image can be interpreted as a two-dimensional integer lattice Z2, and the field of image deformation processing is largely inspired by this view [30].

2.4 Object Detection

Object detection is a machine technology in the field of computer vision and picture handling that manages identifying occurrences of semantic objects of a specific class (such as cars, buildings, or humans). Many researched topics of object detection include object detection and face detection. Moreover, many applications are used in several areas including video surveillance and image retrieval, and it is generally utilized in computer vision, such as face recognition, face verification and video object detection. It is additionally utilized in situations such as following a ball amid a football coordinate, following the development of a cricket bat, or following a human in a video, there are several feature detection algorithms, this thesis will present the specific feature detection algorithm for detection spirulina from water, the commonly feature detection (SIFT, HOG, SURF, FAST) as SURF more advance from SIFT and better for corner detection and HOG disadvantage that is very sensitive to image rotation we will use FAST and SURF [31].

Computer vision identifies and characterizes what is available in true pictures. To accomplish this, researchers have created a number of procedures, one of the primary methodologies utilized in writing has the following advances:

1. Applying a few descriptors to extract highlights from intriguing classes and pictures;
2. Using classifiers to determine which highlights have a place with each class; and
3. Developing indicators that find the articles present in an example picture.

Some of these object detection methods are mentioned in the sections below.

2.5 FAST Features

We saw a few element identifiers and huge numbers of them were large. In any case, when looking from a continuous application perspective, they were not sufficiently quick. One best example would be the SLAM (Simultaneous Localization and Mapping) mobile robot which has limited computational resources and a discovery strategy which could be utilized to concentrate highlight focuses and later used to track and guide protests in numerous PC vision errands [32]. The FAST corner detector was originally developed by Edward Rosten and Tom Drummond and published in 2006 [33]. The most encouraging favorable position of the FAST corner finder is its computational effectiveness. Moreover, when machine learning techniques are applied, superior performance in terms of computation time and resources can be realized. The FAST corner identifier is truly appropriate for continuous video preparation applications in view of its rapid execution.

Features from Accelerated Segment Test (FAST) is a corner identification strategy, and it is employed on extricate include indicates and following utilized step and guide protests in multiple image processing algorithms. The corner FAST locator was initially created by Tom Drummond and Edward Rosten.

Distributed on 2006, the usual encouraging leeway such as the FAST corner identifier is its computational effectiveness. Alluding to its sign, it is quick and without doubt quicker than many other techniques. For example, Distinction of Gaussians (DoG) is utilized by the SIFT, Harris and SUSAN indicators. The most encouraging improvement of the FAST corner detector is its computational performance. The FAST features method uses a circle group of 16 pixels to analyze whether a candidate point p is truly a corner such that individual pixels in the circle are labelled with integers 1 to 16 clockwise. A corner is detected if a set number of adjacent pixels in the circle (called N) is brighter in intensity than canter pixel P or if N is blacker in intensity than the canter pixel P [34].

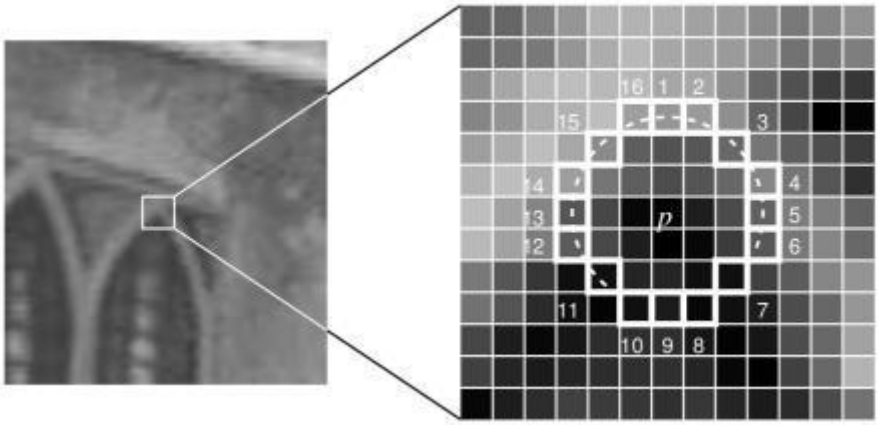


Figure 2.3 FAST feature filter

The pixel *p* is a corner if there exists many *n* adjoining pixels surrounding it (of 16 pixels as shown in Figure 2.3) which are altogether more brilliant than $I_p + t$, or everything darker than $I_p - t$ (appearing as white dash lines in the above picture). *N* was selected to be 12 [35].

A fast test was proposed to avoid an enormous number of non-corners. This test analyzes only the four pixels at 1, 9, 5 and 13 (initially 1 and 9 are tested for the possibility that they are excessively brighter or darker. We assume this to be the case, at point checks 5 and 13). In the event that *p* is a corner, at that point in any event three of these must all be more brilliant than $I_p + t$ or darker than $I_p - t$. If neither of these is the situation, the point *p* cannot be a corner. The full fragment test paradigm would then be able to be connected to the passed competitors by analyzing every pixel in the circle [36].

2.6 SURF Features

In computer vision, Speeded up Robust Features (SURF) is a patented local feature detector and descriptor. It can be used for tasks such as object recognition, image registration, classification or 3D reconstruction, SURF is better than Sift (Scale Invariant Feature Transform) in rotation invariant, blur and warp transform, SURF is 3 times faster than SIFT because using of integral image and box filter [37].

SURF was first presented by Herbert Bay, et al. at the 2006 European Conference on Computer Vision. The use of the calculation is protected in the United States [38].

To detect points of interest, SURF uses an integer approximation of the determinant of the Hessian blob detector, which can be computed with three integer operations

using a precomputed integral image. Its feature descriptor is based on the sum of the Haar wavelet response around the point of interest. These can also be computed with the aid of the integral image. Blob detection methods are aimed at detecting regions in a digital image that differ in in terms of properties, such as brightness or color, compared to the surrounding regions, by considering the scale-normalized determinant of the Hessian, and also referred to as the nonlinear second-order partial differential equation [39]. When determining the scale-space maxima of this operator, one obtains another straightforward differential blob detector with automatic scale selection which also responds to saddles [40]. SURF uses square-shaped filters as an approximation of Gaussian smoothing. (The SIFT approach uses cascaded filters to detect scale-invariant characteristic points in which the Difference of Gaussians (DoG) is progressively calculated on rescaled images.) Instead of using a different measure for selecting the location and scale of interest points (e.g., Hessian and DOG in SIFT), SURF uses the determinant of H to find both. [41]:

$$\det(H_{approx}^{SURF}) = L_{xx}^{'} L_{yy}^{'} - (0.9 L_{xy}^{'})^2 \quad (1)$$

SURF achieves a special blurring effect on the original image, called Scale-Space, and ensures that the points of interest are scale invariant. SURF descriptors are used to locate and recognize objects, people or faces.

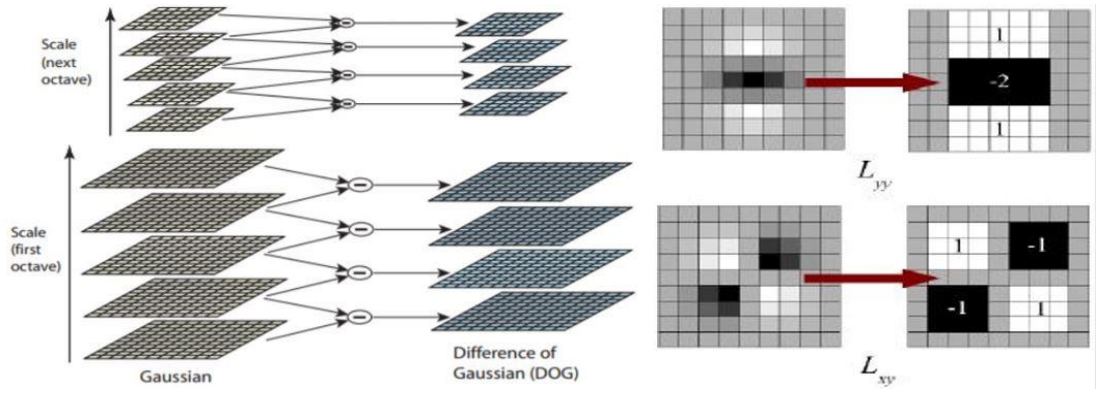


Figure 2.4 SURF feature filter [42]

SURF goes further and approximates LoG with the Box Filter, as shown in Figure 2.4. One significant advantage of this approximation is that the convolution with a box filter can be easily calculated with the help of integral images. Moreover, it can be done in parallel for different scales in addition to SURF relying on the determinant of the Hessian matrix for both scale and location [42].

2.7 Background Concepts about Diatoms and Water Quality

Spirulina is the most cultivated microalga worldwide. Every spirulina species is adjusted to certain autecological ranges, so it is conceivable to set up a good connection between the organization of spirulina networks and the synthetic parameters of the earth, as expressed in [43]. Hence, different productions since the 1970s have exhibited the adequacy of water quality control strategies dependent on diatoms as bio indicators. For instance, these merit referencing: [44], [43], [45], [46], [47] and [48]. Among the scientific categorizations proposed in [49] to use as bio indicators of water quality, it suggests the utilization of phytobenthic networks. Inside phytobenthic networks, the diatoms are commonly the most delicate gathering For instance, the extent of the greater part of the diatom species is somewhere in the range of 10 and 80 and one of them is spirulina, which will be discussed further in the section below. Moreover, their silica skeleton gives them incredible protection from warmth, acids and decay [50].

In the sixteenth century, the spirulina type diatom species was named *Tecuitlatl* by the Aztecs, meaning ‘stone’s fertilizer.’ Afterward, because of an outbreak of infectious diseases, it was not known until when humans had started to utilize microalgae. However, at present these assets can be alleged to be a “green tendency” [51]. Spirulina was removed by locals from the lake and rivers. It was dried into shapes and called ‘dihe’ in the ninth century. Spirulina shapes are shown below in Figure 2.5. The morphology of spirulina with 4 different zoom images are shown below in Figure 2.6 [52].

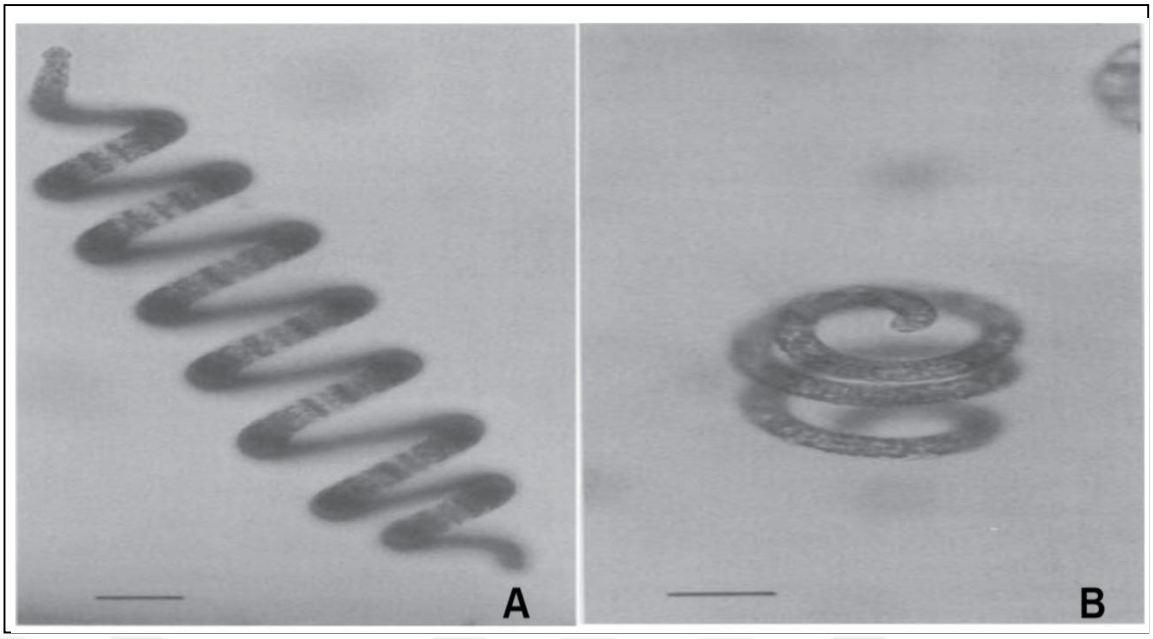


Figure 2.5 A. Light micrograph of spirulina. B. Light micrograph of spirulina platensis. Bar represents 20 μm

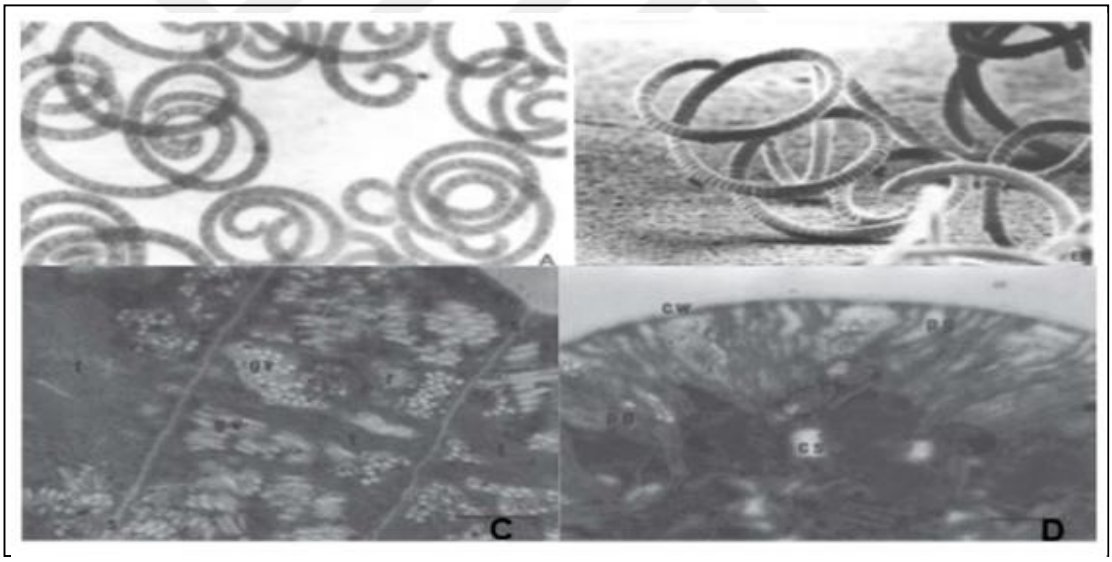


Figure 2.6 Morphology of spirulina with four zoom images.

The presentation of high goals of microphotography has permitted the orderly chronicling of numerous spirulina pictures. This permits the production of electronic accumulations for ordered automation and institutionalization purposes, as proposed in [53]. Traditional ordered conclusions are performed utilizing distinguishing proof keys or by visual correlations with model arrangement tests or reference iconography, as in [54]. There have been a few examinations with regard to the automation of the

distinguishing proof process with picture detection. For instance, the Automatic Diatom Identification and Classification (ADIAC) book published in 1991 [55] presented attractive outcomes by applying picture preparation and shape verification apparatuses to recognize the framework and ornamentation of each frustule, exploiting from that data to play out the distinguishing proof, as clarified in [56]. The framework created in the ADIAC venture permitted the distinguishing proof of 37 species at an exactness of 97%. Nonetheless, it did not consider different elements that can impact the distinguishing proof, including the life cycle of the examples or the water quality itself [57][58]. Moreover, geometry morphometric systems connected to photographic pictures of diatoms showed promising outcomes, as expounded in [59] and [60]. There are additionally later examinations on the use of elective characterization techniques in [61] and the proposition of new highlights such as surfaces in [62].

2.8 Neural Network

A neural network (NN) is a mathematical model similar to some of the characteristics found in brain functions. However, there is controversy as whether or not it is actually a simulation of the nervous system of the organism or that it may be called an artificial neural network (ANN) or something similar. In addition, differences are also being studied as a result of biology and progress.

A neural network refers to a general model in which an artificial neuron (node) is formed in a network by synapse connection changes. The strength of the synapse connection occurring through learning and has a problem solving ability. In a narrow sense, it sometimes refers to multilayer perceptions that use error back propagation, but this is incorrect usage. An artificial neuron in a general neural network, in fact, utilizes an extremely simplified operation of a living neuron [75].

A neural network may be divided into supervised learning that is optimized to a problem by inputting a teacher signal (correct answer), and into unsupervised learning that does not require a teacher signal. Nevertheless, essentially unsupervised learning and teacher learning are equivalent. It has been proved that a neural network with three or more layers can approximate a differentiable and continuous arbitrary function. For problems that cannot be separated linearly with multidimensional data, such as images

and statistics, good solutions can often be obtained with a relatively small amount of calculation. Currently, it is applied as image recognition, analogy of purchases based on customer data in the market, etc. (as well as pattern recognition and data mining). The forward propagation type neural network (feed forward neural network (FFNN)) was originally conceived as an artificial neural network model of a simple structure. It does not have a connection that loops in the network, but indicates a signal propagation in a single direction, such as input node → intermediate node → output node. The learning of neural networks by teacher signals is based on the law of synaptic plasticity proposed by the psychologist Donald Hebb in 1949 (Hebb's Law) [76].

The network connection between nerve cells becomes thick, and as a result, a signal transmission pathway to a specific cell is created (information flows easily), which is called learning. A perceptron divides a set into hyperplanes as a result of learning. This learning was proved by Marvin Minsky in 1969 to converge in a finite number of trials [77]. David Rumelhart et al. (1986) proposed and used backpropagation as a learning model of the multi-layer perceptron [78]. Back propagation is mainly used when the middle layer is one layer, and when there are two or more middle layers, it is called deep learning. The layer near the input is learned by the auto encoder and then stacked.

A neural network is simply a function that fits some data. In its simplest form, there is a single function fitting some data, as shown below. This structure is called a neuron, as shown in Figure 2.7 [79].

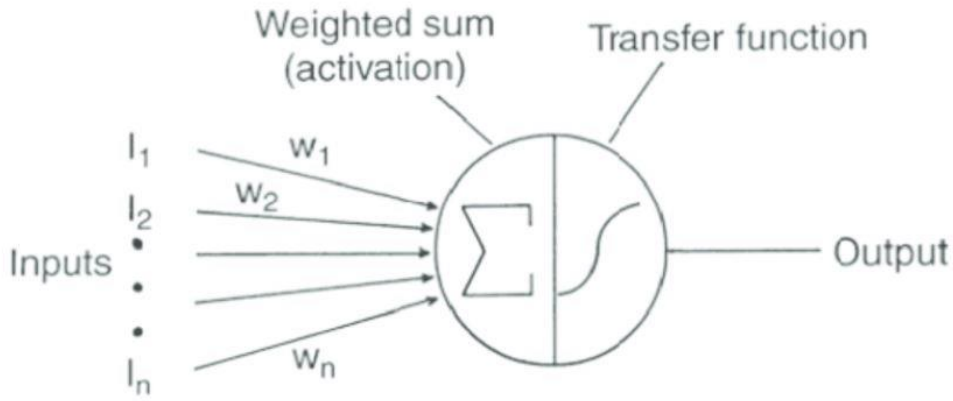


Figure 2.7 Schematic of a neuron [79]

The function can be anything, such as a linear function or a sigmoid function. Of course, a single neuron has no advantage over a traditional machine learning algorithm.

A neural network combines multiple neurons. We may think of neurons as the building blocks of a neural network and by stacking them, one is able to build a neural network, as in Figure 2.8.

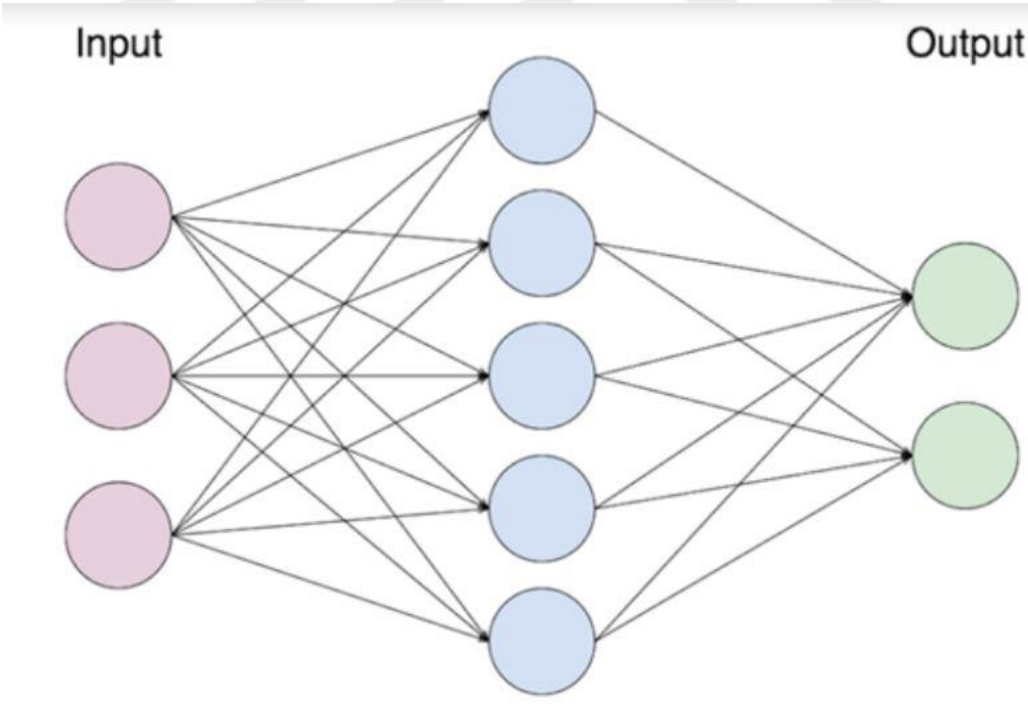


Figure 2.8 Schematic of a neural network

We can observe above how each input is fed into each neuron. The neural network will autonomously determine which function best fits the data. The only requirements are the inputs and the output.

The objective is to build a neural network that will take an image as an input and output whether or not it is a picture of cat. There should, of course, be number of images

For example, we have 300 images in the data set and we have 50 images for training. Each image is a square of width and height of 64 pixels. Moreover, we observe that the image has a third dimension of 3. This is due to the image being composed of three layers: a red layer, a blue layer and a green layer (RGB). Each value in each layer is between 0 and 255, and it represents the intensity level of red, blue, or green in that pixel, thereby generating a unique color for each combination

When selecting the activation function, one of the first steps in building a neural network is to find the appropriate activation function, such as when we wish to predict whether or not a picture has a cat. Therefore, this can be framed as a binary classification problem. Ideally, we would have a function that outputs 1 for a cat, and 0 otherwise.

Order is the manner of predicting a subjective reaction. Techniques utilized for characterization frequently predict the likelihood of every one of the classes of a subjective variable as being the reason for making the classification. With a particular goal in mind, they continue as regression techniques. With characterization, we can respond to questions such as: An individual has many side effects that could be credited to one of three ailments. Is an exchange deceitful or not, To answer such inquiries, various strategies are utilized as calculated regression, k-closest neighbors (knn), quadratic discriminant examination (QDA), direct linear discriminant investigation (LDA) and others. With regard to order, we determine the likelihood of whether a perception is a piece of a specific class. In this manner, we may wish to express the likelihood with an incentive somewhere in the range of 0 and 1. A likelihood near 1 implies the perception is likely to be a piece of that classification. In order to create values somewhere between 0 and 1, we express the likelihood utilizing this condition [80]:

$$P(X) = \frac{\exp(\beta 0 + \beta 1\, X)}{1+\exp(\beta 0 + \beta 1\, X)} \quad (2)$$

The condition above is characterized as the sigmoid capacity. When plotting this condition, we see that this condition dependably results in an S-molded bend bound somewhere between 0 and 1, as shown in Figure 2.9.

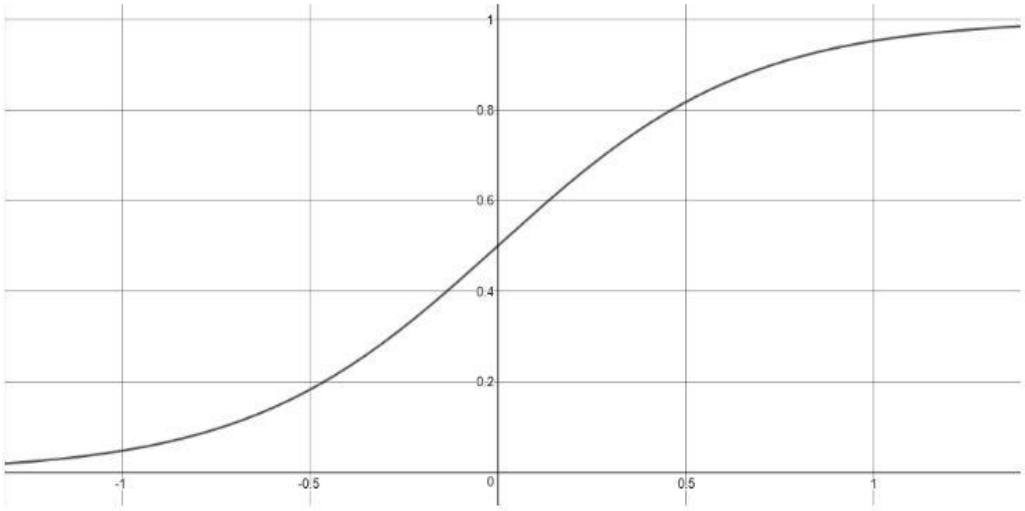


Figure 2.9 Logistic regression curve

The condition is known as the logit. It is obvious, that it is direct in *X*. Here, in the event that the coefficients are certain, at that point an expansion in *X* will result in a higher likelihood,

The intuition here is that we need coefficients to such an extent that the anticipated likelihood (signified with a punctuation in the condition above) is as close as conceivable to the observed state. Additionally, to direct a relapse, we utilize the p-esteem.to determine whether or not the invalid theory is rejected. The *Z*-measurement is additionally generally utilized. An enormous outright *Z*-measurement implies that the invalid theory is rejected. We need to in mind that for the invalid speculation states, there is no connection between the highlights and the objective.

Multiple logistic regression: Obviously, strategic relapses can, without much of a stretch, be extended to oblige more than one indicator:

$$Log\left(\frac{p(x)}{1-p(x)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p \quad (3)$$

Note that utilizing numerous strategic relapses may provide better outcomes since it can consider relationships among indicators, which is confounding. Additionally, a single indicator will rarely be adequate to make a precise model for a forecast [81].

2.9 Deep Learning

Deep Learning (called profound organized learning or various leveled learning) is a subset of a more extensive group of AI strategies dependent on neural systems. Learning can be administered, semi-directed or unsupervised [64] [65] [66]. Deep Learning structures, such as convolutional neural networks, recurrent neural networks, deep belief networks, and deep neural networks, have been used in many fields including machine translation, natural language processing, audio recognition, speech recognition, drug design, computer vision, medical image analysis and bioinformatics [67] [68] [69]. It is stimulated by the processing of the data and shared information nodes in biological systems. ANNs have differing contrasts from biological brains [70] [71] [72].

Most current profound learning models depend on other neural systems, the most explicit example being the Convolutional Neural Networks (CNN) [73].

In the application of an illustration recognition each raw data item may contain equal pixels of patterns, the first layer of CNN are the symbolic of encoding edges and pixels, the second layer may create methods and encoded edges. In layer three, the eye and nose will be encoded. In layer four, it may detect a face image. A deep learning method may determine which highlighted features to place in which level optimally by itself. (This seems not entirely to prevent the requirement for hand-tuning) this thesis presents deep neural network rather than neural network because of its hard to train and over fitting and computation time [74].

Most importantly, Deep Learning is a field in flux for the following primary reasons: first, the development of digital devices that had been created previously, including neural systems; second, expanding computational power occurring over a number of years.

2.10 Convolutional Neural Network

In machine learning, a convolutional neural network (CNN or ConvNet) is a type of forward propagation artificial deep neural network. It is a widely used model for image and video recognition.

CNN uses a variation of a multilayer perceptron designed to require minimal data pre-processing. The CNN is also called a shift invariant or space invariant artificial neural network (SIANN) based on its weight (matrix) shared structure and translation invariant properties.

Taking a previously trained model that has learned good general features and apply that to our own dataset, adapting the network weights for the problem at hand. The model used was a pre trained AlexNet model [82]. The connection pattern between neurons in the convolutional network was inspired by the arrangement of the visual cortex of an animal. Individual cortical neurons that respond only to stimuli in a limited area of the field of view [link needed correction] [aim needed ambiguity avoidance] are called receptive fields. The receptive fields of different neurons overlap partially to cover an entire field of view.

CNN uses relatively small pre-processing compared to other image classification algorithms. This means that CNN learns filters that were designed by human hands in the traditional algorithm. This prior knowledge in feature value design and independence from human effort are the great advantages of CNN.

The CNN is applied to image and video recognition, recommender systems [83] and natural language processing [84].

The Convolutional Neural Network (CNN) is a model for all in-depth learning courses and books. The power of the CNN is very great in image recognition. Many models for pattern recognition are also based on the CNN. Its architecture is based on

extensions and it is also worth mentioning that the CNN model is also a Deep Learning model established by a few references to human brain visual organizations. After learning the CNN, it also becomes helpful to learn other deep learning models, as shown in Figure 2.10:

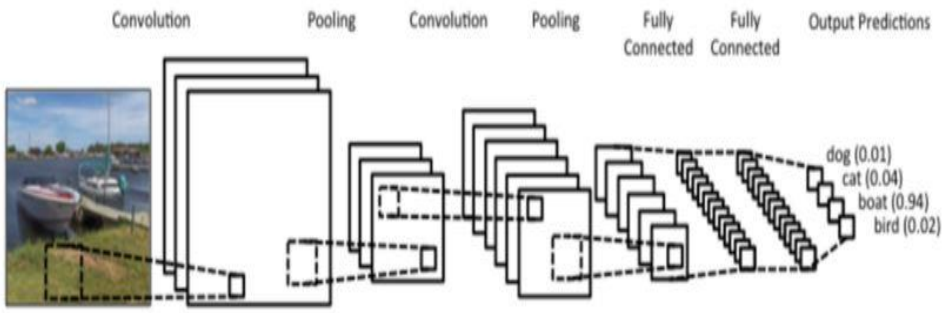


Figure 2.10 Deep learning layers [82]

Simply, the two images of Convolution, Pooling, and Fully Connected are the architecture of the CNN. Therefore, fully understanding the CNN occurs by understanding the contents of Convolution and Pooling.

The Convolution Layer, being the convolution operation, convolutes the original image with a specific Feature Detector filter. The convolution operation multiplies the two 3 × 3 matrices in the following figure, and then adds them, as seen in Figures 2.11 and 2.12.

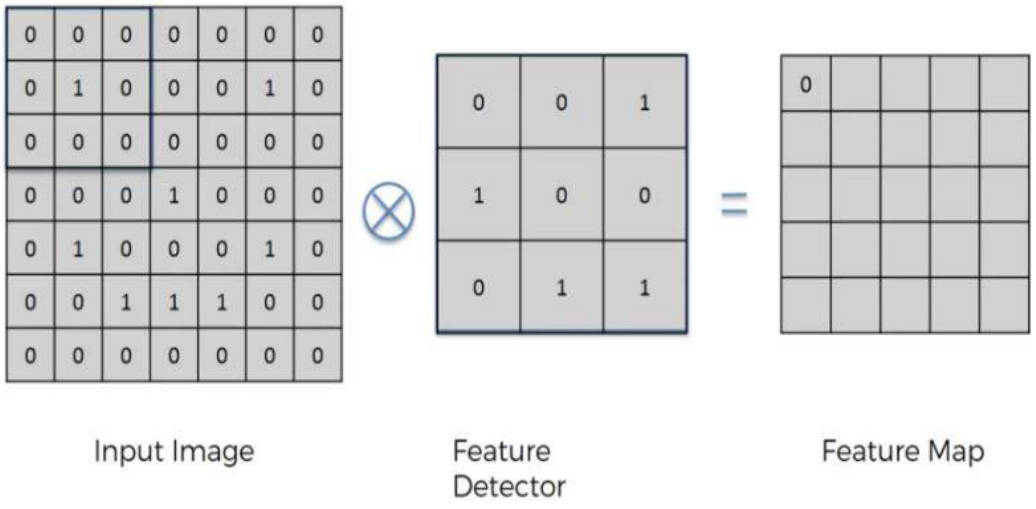


Figure 2.11 Convolution Layer Filter

Completing the table in sequence in Figure 2.12

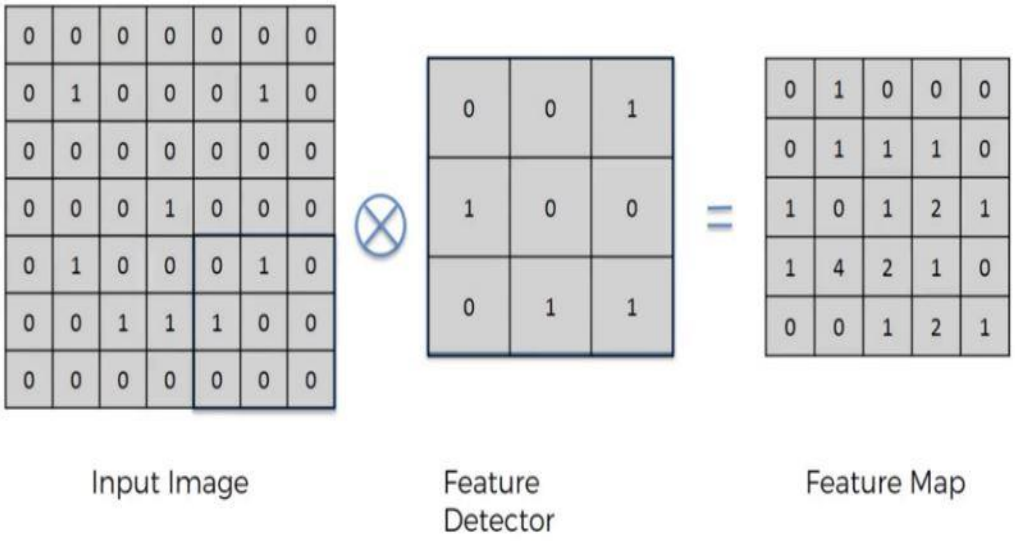


Figure 2.12 Convolution Layer Filter complete table

The middle Feature Detector (Filter) will randomly generate several kinds (e.g., 16 kinds). The purpose of the Feature Detector is to help to extract some features (e.g., the shape) in the image, similarly to how the human brain judges what a picture is based on its shape, as in Figure 2.13.

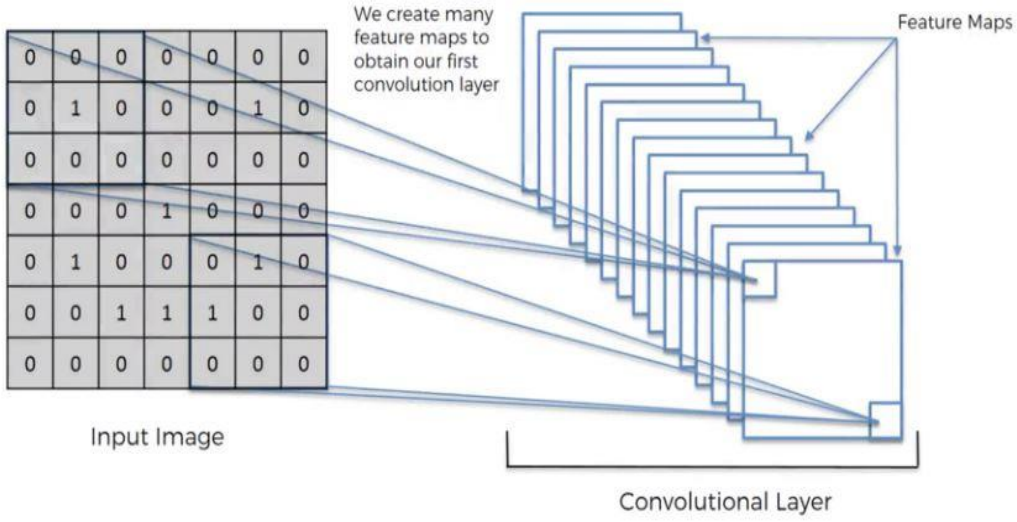


Figure 2.13 Convolution Layer Features

To extract the boundary of the object with the Feature Detector, we use the ReLU function to remove negative values and edges as in Figure 2.14 [85].

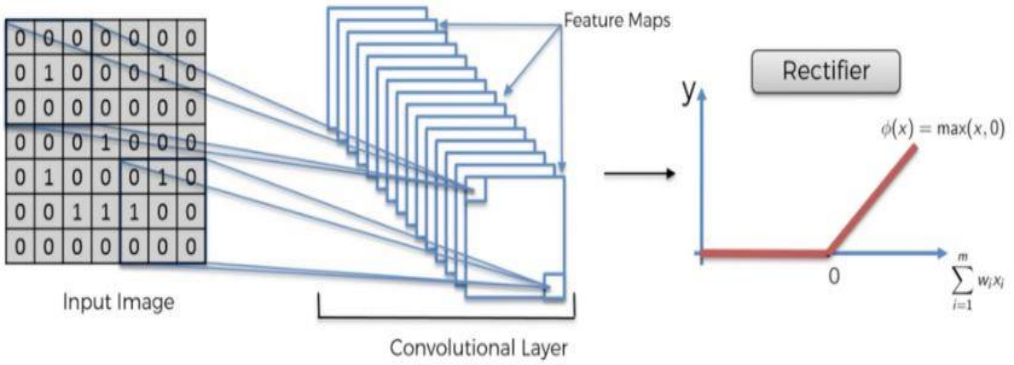


Figure 2.14 ReLU function

The sigmoid function is an S-shaped curve. The main reason for using the sigmoid function is because it exists between 0 and 1. Therefore, it is especially used for models where we have to predict probability as an output. Since the probability of anything exists only between the range of 0 and 1, the sigmoid is the most appropriate choice (Figure 2.15).

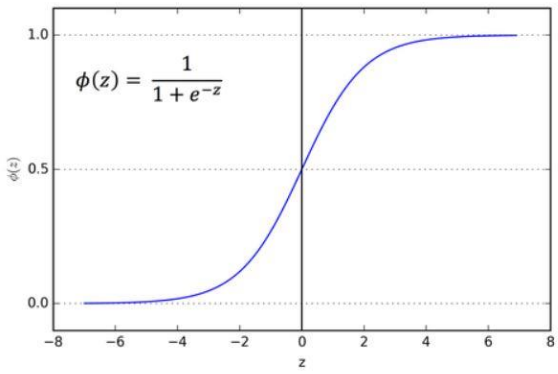


Figure 2.15 Sigmoid function

The ReLU is currently the most used activation function and it is used in almost every convolutional neural network and deep learning, as seen in Figure 2.16.

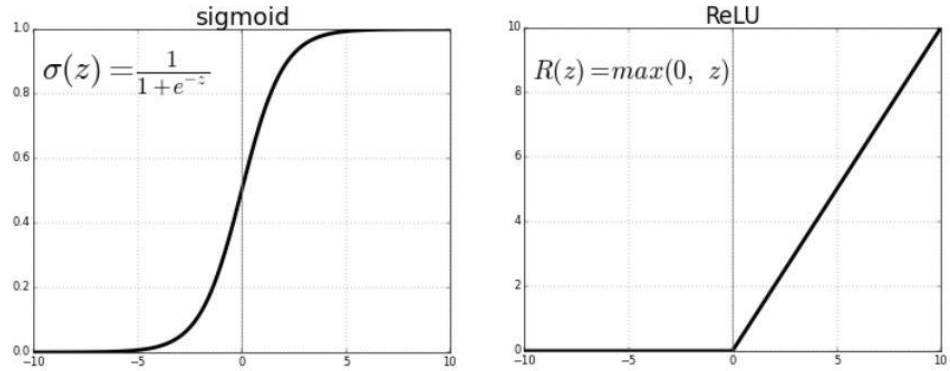


Figure 2.16 ReLU vs. Logistic Sigmoid functions

The ReLU curve is half rectified (from the bottom). R(z) is zero when z is less than zero and R(z) is equal to z when z is greater than or equal to zero. Over the range [0, ∞), both the function and its derivative are monotonic. However, the issue is that all the negative values become zero immediately, which decreases the ability of the model to fit or train from the data properly. This means that any negative input given to the ReLU activation function turns the value into zero immediately in the graph, which in turn affects the resulting graph by not mapping the negative values appropriately.

Pooling Layer: In the Pooling Layer, Max Pooling is mainly used. The concept of Max Pooling is very simple [86]. We only need to select the maximum value in the matrix. The main advantage of Max Pooling is that when the image is translated by a few

pixels, it will not affect the judgment, as in Figure 2.17 and Figure 2.18. Moreover, Max Pooling has a good anti-noise function [87].

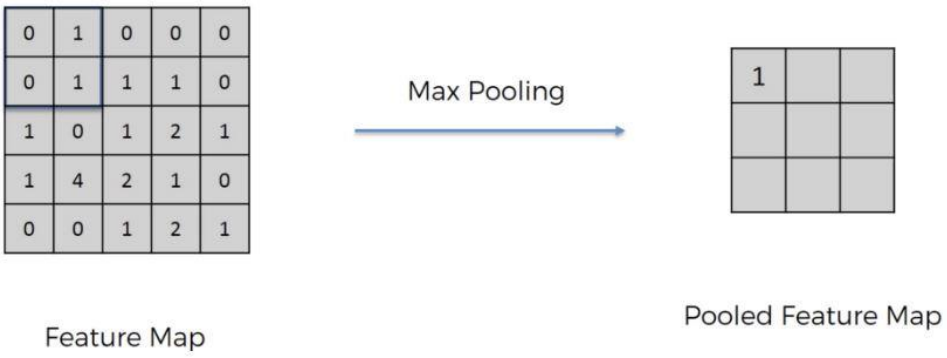


Figure 2.17 Max Pooling

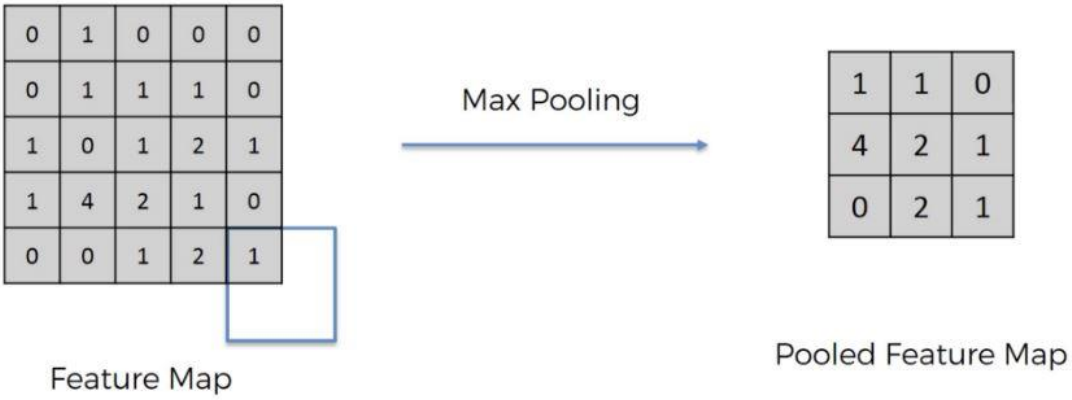


Figure 2.18 Max Pooling full filter

Basically, the purpose of the fully connected layer is to flatten the previous results followed by receiving the most basic neural network, as in Figure 2.19.

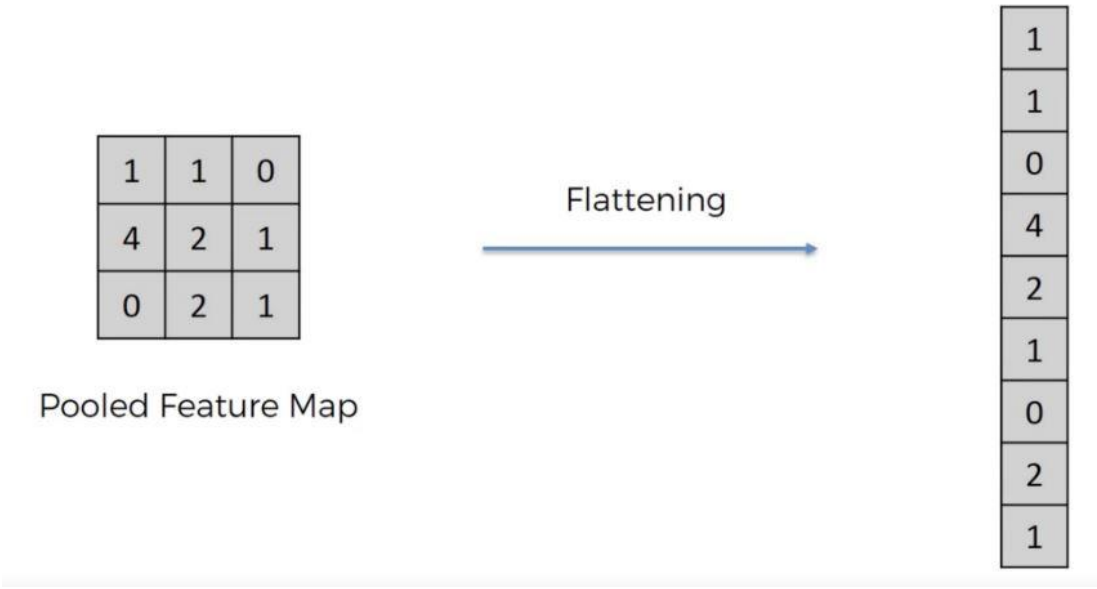


Figure 2.19 Fully Connected Layer

Completely associated layers interface each neuron in one layer to each neuron in another layer. It is in principle the same as the traditional multi-layer perceptron neural network (MLP). The straightened lattice experiences a completely associated layer to group the pictures.

In receptive field neural systems, every neuron receives a contribution from a number of areas in the previous layer. In a completely associated layer, every neuron receives a contribution from each component of the previous layer. In a convolutional layer, neurons receive contributions from only a confined sub-area of the previous layer. The sub-area is a square (e.g., a 5 × 5 matrix). The information field of a neuron is called its responsive field. Along these lines, in a completely associated layer, the responsive field is the entire previous layer. In a convolutional layer, the open region is smaller than the entire previous layer.

Each neuron in a neural network computes an output value by applying some function to the input values coming from the receptive field in the previous layer. Learning in a neural system advances by making gradual acclimations to the predispositions and loads. The vector of weights and the bias are called a filter and represent some features of the input (e.g., a particular shape). A distinctive element of CNNs is that numerous neurons share a similar channel. This decreases memory loading on the grounds a

single vector of loads are utilized over every open field sharing that channel rather than each responsive field having its own predisposition and vector of loads [88].

2.11 Validation

When a customary picture examination strategy has been connected, it is essential to evaluate the execution to realize how exact it is or contrast it with different techniques as k-Fold Cross-Validation and Random Split Validation, however, we need to calculate and know the sensitivity and specificity of each run of our system, so with Confusion Matrix Validation, we could separate the outcomes of each run from the sensitivity and specificity. The Confusion Matrix is a forbidden description which expresses the connection between the anticipated occurrences and the correct outcomes [89]. There are four outcomes as below:

• True Positive. The example has a place with the class as anticipated.

• False Positive. The event does not have a place with the class but rather is anticipated as being positive. This is the purported Type 1 error.

• True Negative. The event does not have a place with the class as anticipated.

• False Negative. The example belongs to the class; however, it is anticipated as being negative.

The results of the detection according to color are given in Table 2, and we calculated sensitivity as the ability of a test to label an object accurately.

Sensitivity = X / X + Y

= X (True Positive) / X + Y (False Positive + True Positive)

= Likelihood of existing test positive when object is detected

For the ability of a search to tag an image as spirulina accurately is described by specificity,

Specificity = M / N + M

= M (True Negative) / M + N (True Negative + False Negative)

= Likelihood of existing test being incorrect

A few measurements can be determined to outline the execution in specific perspectives.

And accuracy calculated as: P=TP+TN and N=FN+FP

$$ACC = \frac{TP+TN}{TP+TN+FN+FP} = \frac{TP+TN}{P+N} \quad (4)$$

CHAPTER 3

METHODOLOGY

This chapter clarifies how the methods and image adjustment were used and the manner in which the task results were obtained. Additionally, Figure 3.1 shows all the steps that are taken to detect spirulina from water as the sections below cover all the materials and steps which are taken to detect spirulina in water and identified as very explicit points. The consequences of this procedure are presented in Chapter 4. Starting from dataset sort planning to improvement on detection algorithms. Once our dataset was prepared, it was important to apply image preparation, such as resizing the image, to ensure that every input image had the same resolution. Then every detection method would be covered.

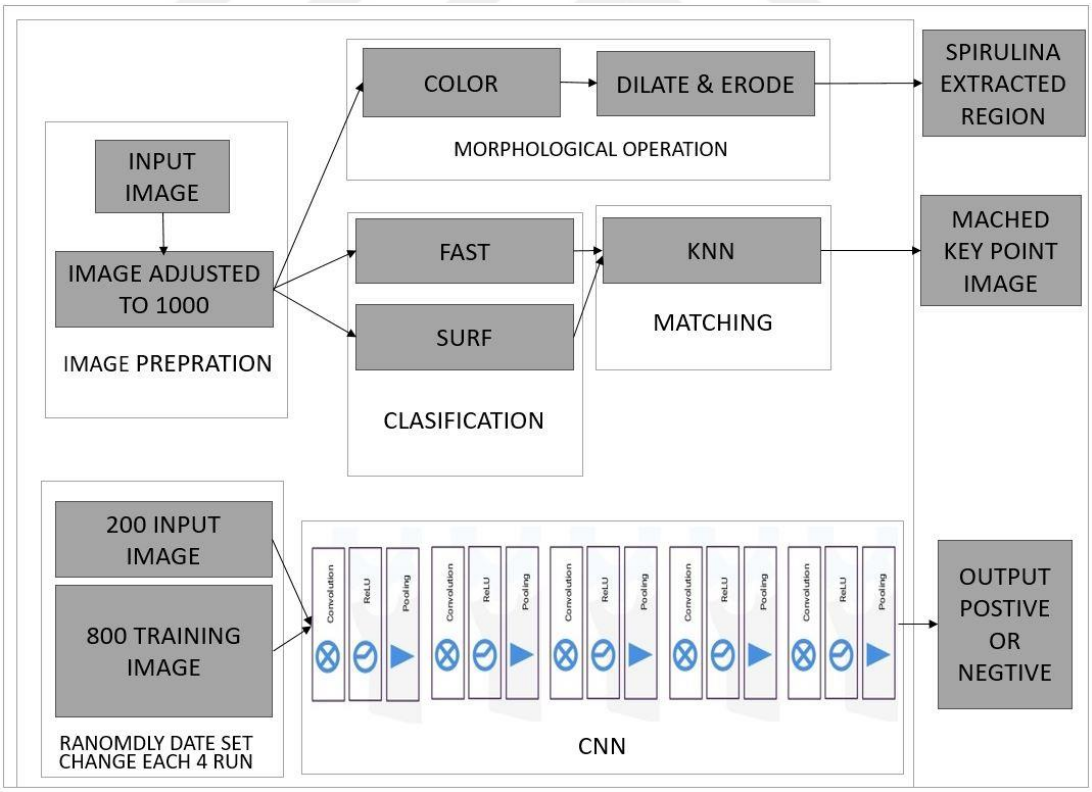


Figure 3.1 Methodology Diagram

3.1 Image Adjustment

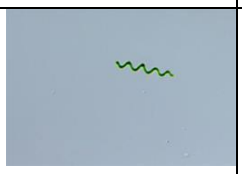
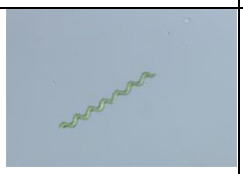
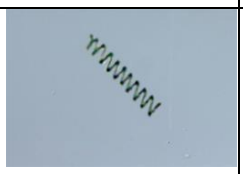
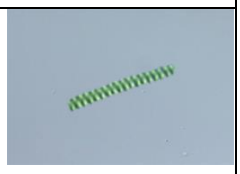
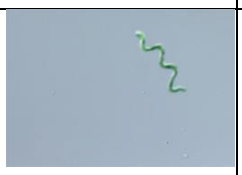
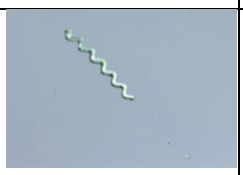
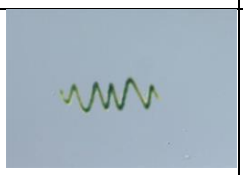
Spirulina culture sample images are captured from microscope views. Those images are the input images for further processing. All genuine laboratory images are obtained from Selçuk University. The images were collected by students of Assoc. Prof. Dr. Cengiz Akköz in a recent thesis study on the diatomaceous life cycle in Turkey. All images of the water samples were collected from rivers and lakes in Turkey. However, the number of genuine spirulina images was not satisfactory for this presented study. This thesis presented an artificial image generation theory to produce alternative images with all size probability and rotation probability in our image dataset preparation. We acquired a total of 60 original single-cell images for the four types of spirulina. In order to obtain an experimental image dataset, we modified the original images with random rotations of several degrees. No filter was applied to the images. Computational delays were avoided by resizing every image to 227×227-pixel sized images, which helped us to obtain stable and fixed output results using the Nearest-Neighbor interpolation method by replacing multiple pixels with a pixel of the same color. Our artificial image generation process yielded 1000 supplementary images of which 60 were of spirulina and 40 of non-spirulina. We used a single letter code to refer the types of spirulina, namely "A" for Laxa, "B" for Major, "C" for Nordst, "D" for Princeps and "X" for non-spirulina cells. In line with this type-coding used, the naming notation is shown in Table 3.1. The table contains six columns. The first column shows the names of the original images without any modification. The second and third columns show the generated combinations of two cells out of the original images. As an example, the code "AA" means 2 Laxa cells in one image and the code "AB" means 1 Laxa and 1 Major cell in an image. The fourth column contains one type combined with two different types. For example, the code "ABC" means 1 Laxa (A) cell with 1 Major and 1 Nordst cell 1 Laxa in one image. Column five contains all species in one image. For example, the code "ABCD" means 1 Laxa, 1 Major, 1 Nordst and 1 Princeps cell in one image. Column six shows the naming convention used to combine all the above cell variations with type X, which is not a spirulina cell. The letter "M" used in the table notation means multiple cells in the image. For example, "(AB) M" means multiple type A and multiple type B in one image.

Table 3.1 Image Datasets with 1000 images

Original image	Two cell same type	Original type with different types	Original type with multiple different types	All in one image	X
A	AA,(AA)M	AB,AC,AD (AB,AC,AD)M	ABC,ABD,ACD (ABC,ABD,ACD)M		
B	BB,(BB)M	BC,BD,BA (BC,BD,BA)M	BAC,BAD,BDA (BAC,BAD,BDA)M	ABCD	X(A,AA,AX,B,C,D,AB, AC,AD,ABC,ABD,B,BB, BC,BD,BCD,C,CC,CD, D,DD,ABCD)
C	CC,(CC)M	CD,CA,CB (CD,CA,CB)M	CBA,CBD,CDA (CBA,CBD,CDA)M	(ABCD)M	X(A,AA,AX,B,C,D,AB, AC,AD,ABC,ABD,B,BB, BC,BD,BCD,C,CC,CD, D,DD,ABCD)M
D	DD,(DD)M	DC,DA,DB (DC,DA,DB)M	DCA,DBA,DBC (DCA,DBA,DBC)M		

Finally, 500 spirulina images and another 500 non-spirulina images became available in our dataset for the 4 species. The total number for each are shown in Table 3.2

Table 3.2 Image Datasets with 1000 image

	(A)	(B)	(C)	(D)
Original Image	 Total 15 Images	 Total 22 Images	 Total 13 Images	 Total 10 Images
Modified Image (Rotation - Resizing)	 Total 125 Images	 Total 125 Images	 Total 125 Images	 Total 125 Images
Non-spirulina Images	 Total 500 Images			

3.2 Detection Based on Color

The goal of color pre-processing is to identify the green intensity threshold for spirulina cells in microscopic images as green is the dominant color in spirulina images. The LAB image format was used to experiment on the green color intensity threshold and with manual tests, the thresholds for the LAB format was determined, as in Figure 3.2.

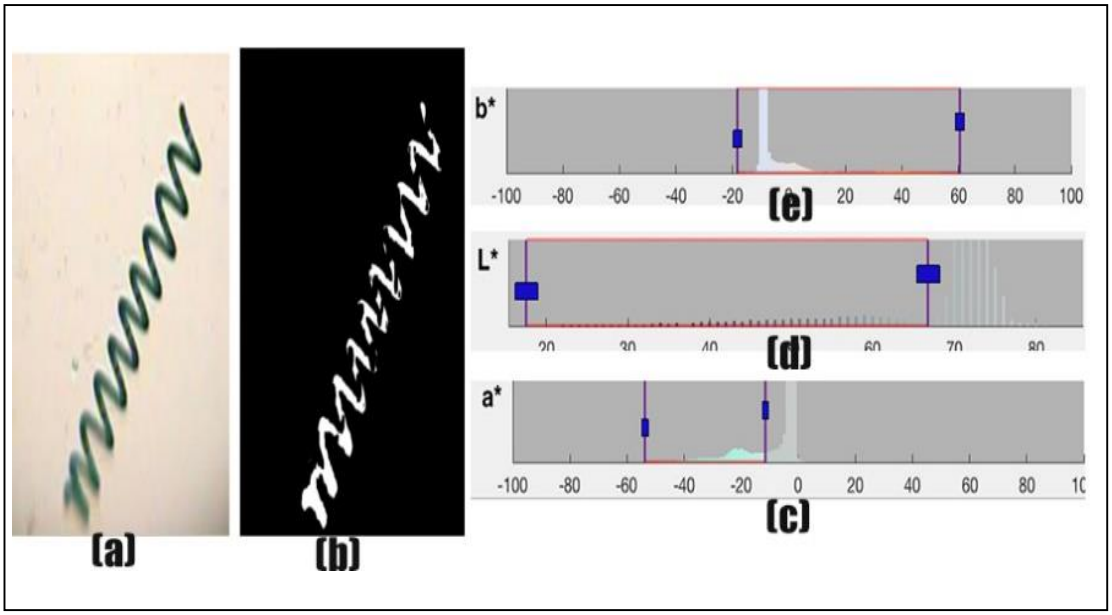


Figure 3.2 (a) Original Image, (b) Color Threshold, (c) A*Green–Red Value Threshold, (d) Dimension L for the Lightness, (e) B* Blue-Yellow Color Components.

The texture of spirulina is very irregular and uneven and there are not many colors. The intensity of the green in this image changes in addition to its brightness. Therefore, the best option here is to study the color level of the image in this manner, and when we apply thresholds to it, it will deal with the spirulina as one compete object.

We threshold the image so that only the color that we want to contour appears white and the others convert to black. This step does not change much here; however, it must be performed as contouring works best with black and white (threshold) images.

The threshold limit was produced as B with intensity ranging from −20 to 60, as shown in Figure 3.1, as L with intensity ranging from 20 to 68 and as A with intensity ranging from −63 to −25. With these values, we were able to detect the green regions. All the results are presented in Chapter 4.

Since irregularities also appeared in the background, we were able to obtain the largest contour using this method, the largest contour being, of course, the spirulina. We were also able to find the index of the spirulina contour in the contours array, from which we found the area and center of the spirulina.

3.3 Spirulina Detection Using FAST Features

Is a corner detection method, which could be used to extract feature points and later used to track and map objects in many computer vision tasks, The most promising advantage of the FAST corner detector is its computational efficiency. The FAST features method uses a circle of 16 pixels to classify whether a candidate point *p* is actually a corner. Each pixel in the circle is labelled with an integer from 1 to 16 clockwise [8]. In the presented study, initially the FAST features in the original image are identified, after which the algorithm seeks the images of those that these features match. The results of the FAST feature experiment are shown in Table 4.3 in chapter 4. The first step image is converted to a gray scale image, as in Figure 3.3 The lightness method is used to convert RGB to grayscale. This method averages the most prominent and least prominent colors: (max(R, G, B) + min(R, G, B)) / 2.

The average method simply averages the values: (R + G + B) / 3.

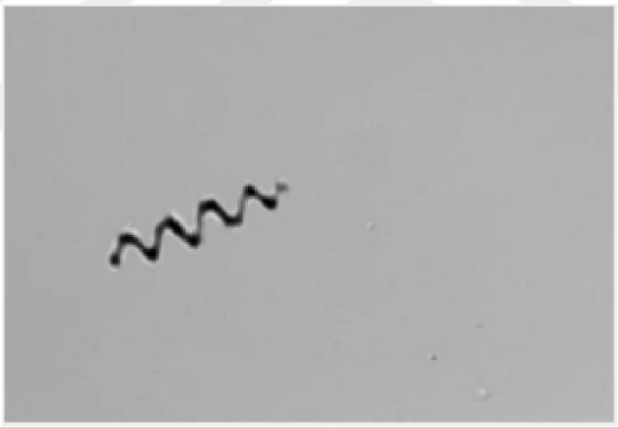


Figure 3.3 Gray Spirulina image

This model explains the best way to perform corner location by utilizing the highlights from the quickened portion test (FAST) calculation. The FAST calculation determines whether a corner is available by testing the circular territory around the potential focal point of the corner. The test recognizes a corner whether the pixels in an adjacent region are either brighter than those in the middle in addition to a limit, or whether they, short of an edge, are darker than the inside. After converting the image to

grayscale, the next step is to find the key points that are basically values of the corner about the key points, as in Figure 3.4.

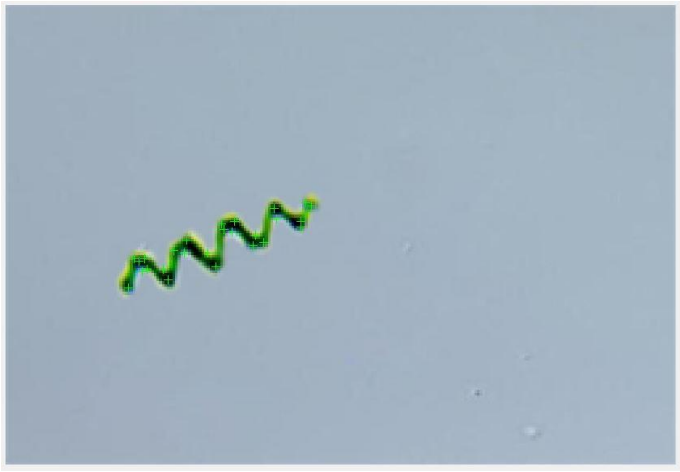


Figure 3.4 FAST key points

After applying FAST feature detection over both input image then the matching between two images followed by using the KNN-based matcher to define the KNN (K nearest neighbor) method to match each vectors of each key point we have in the image after applying FAST feature on the both input original images, the result of matching key point of FAST features are shown in Figure 3.5

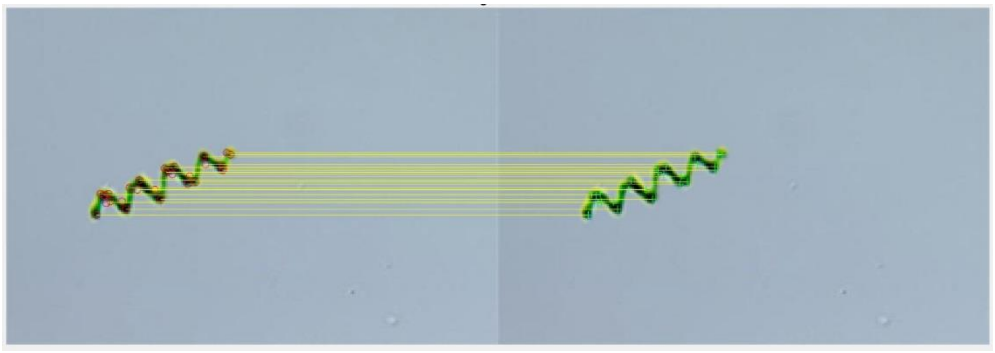


Figure 3.5 Matching key points

3.4 Spirulina Detection Using SURF Features

In Computer Vision, Speeded up Robust Features (SURF) is a patented local feature detector and descriptor. It can be used for tasks such as object recognition, image registration, classification or 3D reconstruction, SURF was first presented by Herbert

Bay, et al., at the 2006 European Conference on Computer Vision [90]. SURF achieves a special blurring effect on an original image, called Scale-Space and ensures that the points of interest are scale invariant. SURF descriptors are used to locate and recognize objects, people or faces. In this study, SURF was used to extract features of interest from spirulina images [10]. The SURF method uses a blob detector based on the Hessian matrix to find points of interest. The determinant of the Hessian matrix is used as a measure of local change around the point and points are selected where this determinant is maximal [13]. In contrast to the Hessian-Laplacian detector, SURF also uses the determinant of the Hessian to select the scale. In the first step in this study, SURF features in the original spirulina images are identified and later, these features are used in the detection process.

In the first step the image is converted into a grayscale image. The lightness method is used to convert RGB to grayscale. This method averages the most prominent and least prominent colors: (max(R, G, B) + min(R, G, B)) / 2.

The average method simply averages the values: (R + G + B) / 3, as shown in Figure 3.6.

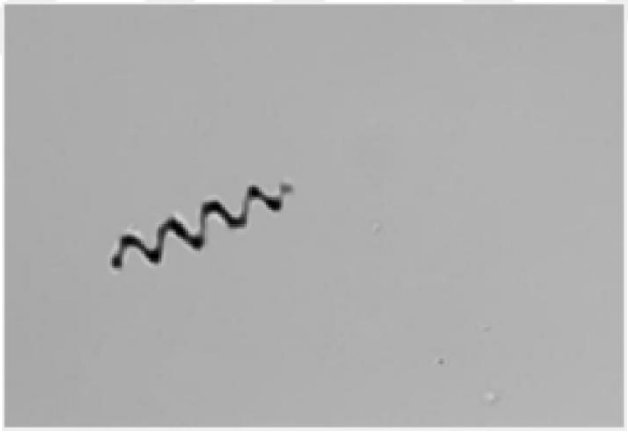


Figure 3.6 Grayscale image of spirulina

The SURF corner finder works by taking flat and vertical coordinates of the picture and searches for territories where both are high. This is measured by the SURF corner descriptor which is characterized for our situation as the framework.

Figure 3.6 shows the scope of the identifier windows that were utilized in this work. Valid highlights are found as neighborhood maxima over a 3 × 3 × 3 territory where

the third measurement is the locator window estimate. Therefore, an element must be locally exceptional over a spatial range and a scope of scales. After converting the image to grayscale, the next step is to find the key points. We calculate the descriptors (which are basically stores information about the key points), as shown in Figure 3.7.

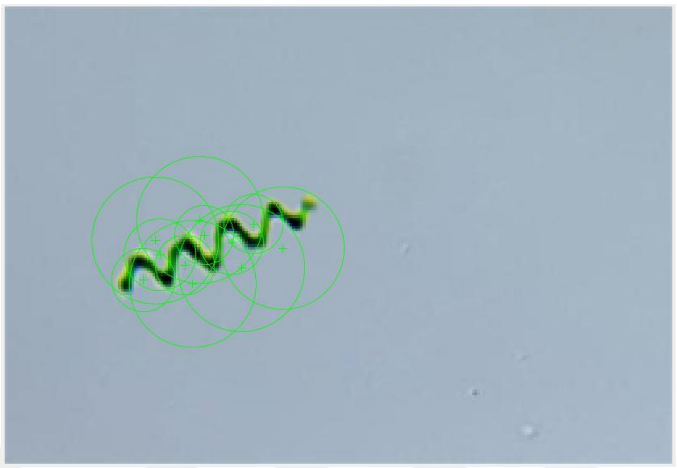


Figure 3.7 SURF key points

After applying SURF feature detection over both input image then the matching between two images followed by using the KNN-based matcher to define the KNN (K nearest neighbor) method to match each vectors of each key point we have in the image after applying FAST feature on the both input original images, the result of matching key point of FAST features are shown in Figure 3.8.

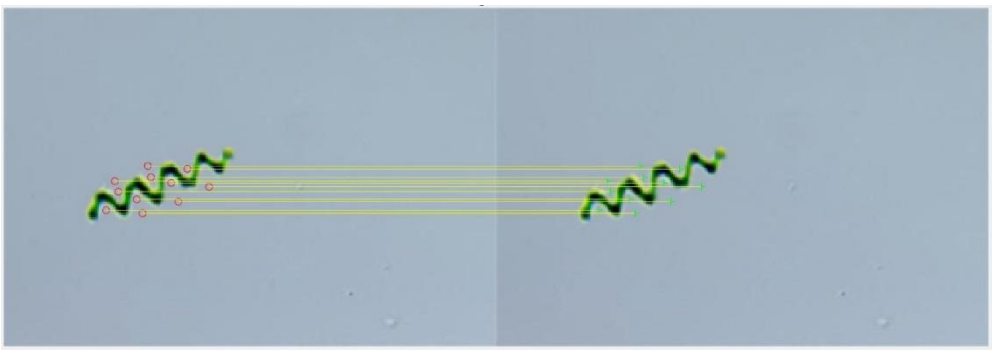


Figure 3.8 SURF key points

3.5 Image Dataset Preparation for Deep Learning

Here, the specific method followed in this study was to build a suitable dataset of spirulina classes. For the detection methodology, we used the Convolutional Neural Networks. In this study, the first steps were to build a dataset of spirulina classes with

a series of image pre-processing, such as data augmentation, data labelling, and image recovery. Details of the image generation methodology are similar to those defined in the paper on diatom detection in water [9]. Furthermore, we will explain the other steps.

3.6 Data Labeling Preparation for Deep Learning

This section provides further understanding of our modified images as raw data considered for classes which have been selected for digitized images. Because there is a need for knowledge and learning of what variety of types can be located and seen，as described and proposed by Gonzalez, R [10]. These labelled images are used in the training phase, and afterwards they are labelled. All class-labelling of data was performed manually.

3.7 Image Preparation for Deep Learning

After image labelling, the image processing steps were applied to the dataset images. These images were small, ranging between 120 × 120 pixels and 500 × 500 pixels.

Le Hou proposed Convolutional Neural Network for Whole Slide Tissue Image Classification one and he train a CNN image patches with selecting a random 300 x 300 sub-patch from each 500×500 image [91]. Andrearczyk and Paul proposed Convolutional Neural Networks for Texture Classification, the results are also robust to small variations of the batch size but they also adapt the latter to the training sizes. Finally, they keep the cropping of the input images to 227x227 for the sake of comparison for input images larger than 227x227 [92]. Because of the different shapes and sizes of spirulina, in this thesis the resizing method would be applied to every image to become 227 × 227 pixels. Some images in the modified dataset were not of high resolution or size, and if unlabeled images had been entered into our training dataset, it would have caused unstable results. Therefore, a standard manual and individual check by a specialist above the chosen areas of interest would be required. The original number of cell images decreased significantly and after unselecting the

mention images, we had a total fall rate of 5% of the total number of images in the prepared dataset.

3.8 Deep Learning

The Convolutional Neural Network model was applied in this study. In the following sections, a comprehensive summary of CNN technology is given. Afterwards, special aspects of validation, testing and preparation are described.

3.9 Convolutional Neural Networks

Neural networks with deep classes are called convolutional neural networks (CNN). They contain more layers for detection features and in creating implementing the neurons, there are essentially new behaviors in machine learning theories. CNNs contain a number of layers of importance to control tuning for statistical parameters needed for testing and the manner of the training model to achieve good results. Every single image sample is fed as an input to train the model by calculating its features. With this repetition, the net is enhanced by pushing it closer to a solution which is required in the reducing loss function. Additionally, there are specific scientific tools to resolve this problem, including the Nester Accelerated Gradient (NAG) presented in [11]. The Stochastic Gradient Descent is a common tool, and another similar tool, known as Adaptive Gradient, is shown in [12]. Tuning the developed method previously is important and advantageous when performed in the technique of transfer learning, and taking an earlier reduced representation of model with learned layers for common features and using this model with our modified dataset images. In this thesis, Alex Net is the most common model with multiple layers with every layer being able to be adjusted [13]. Figure 3.9 shows how the network is designed. As shown in detail below, the Alex Net layers from one to five are the convolution layers and the last three layers are fully connected layers:

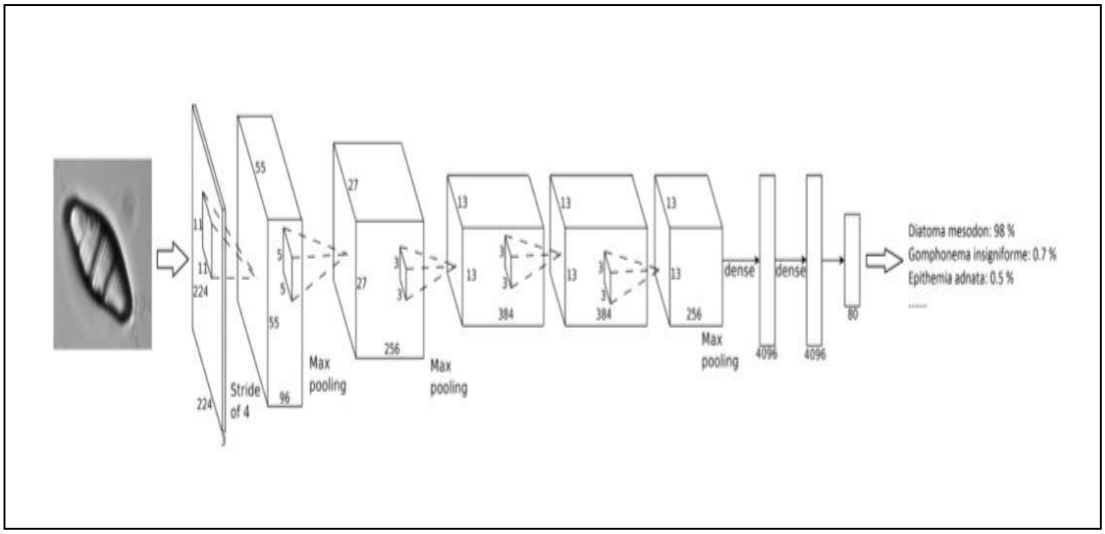


Figure 3.9 Alex Net Network architecture

In the training part, the rate of initial learning and the period of regularization technique for neural networks for back propagation will be out of the main tuning part. The SGD (Stochastic Gradient Descent) is used to compute the next update to the parameters and to achieve the best results. The computer prepared for training this model is a CPU Intel Core i7 7700K with 8 GB VRAM. The deep neural network has two main problems, as explained and reported in the structure paper [6]. The first problem is that of data development because the model may fit with several images from the same direction. Rotations of all features will be predicted only in the direction in which is trained. It is of great importance to find duplicate images in the dataset and to ensure that different versions of the same samples be provided. The second main problem is in the architectures of the dropout layers such that some of the layers for the detection features are turned off. The idea is to "turn off" randomly some of the neurons where the network has avoided local dependencies.

3.10 Testing

For image classification using deep learning, a rule of thumb is 1,000 images, where this number can go down significantly if one uses pre-trained models as AlexNet [93], Pre-trained models is usually applied when there is a small dataset used to train the model [94]. Aydogdu and Vakkas proposed eye detection using CNN with 500 image Images are used in training of the networks, the rest of the 20% is used for testing, of

the images with less than 5% error the date [95]. Emersic and Dejan proposed Ear Recognition using pre trained model AlexNet with limited dataset, CNN-based model with a 1300 dataset images. The result of paper work is 95.51 accuracy [96].

In this thesis one-thousand images were modified and 20% were identified for testing. These images were used in testing the efficiency of the classification. Tests were run on the same hardware mentioned in section 3.10.

3.11 Validation

Once a modified system is used over an image to compare the model with other models, performance quantification becomes necessary. The confusion matrix is an important technique that maps the classified behavior as mentioned in the study. The most relevant ideas are reviewed in [14].

3.12 Code behind the Steps

The net contains eight layers with loads; the initial five are convolutional and the other three are completely associated and directly fully connected. The yield of the last completely associated layer is bolstered in two ways, the first being spirulina and the second being non-spirulina as output.

In the first and second convolution layers, the layer of response-normalization max-pooling layers follows both of the reaction standardization layers just as the last (fifth) convolutional layer. The ReLU non-linearity is connected to the yield of each convolutional and completely associated layer. The input to the net is a $227 \times 227 \times 3$-pixel image.

CNNs make use of filters (also known as kernels) to detect which features, such as edges, are present throughout an image. A filter is a matrix of values, called weights that are trained to detect specific features. The filter moves over each part of the image to check whether the feature it is meant to detect is present. To provide a value representing how confident it is that a specific feature is present, the filter performs a convolution operation, which is an element-wise product and sum between two matrices, as shown in Figure 3.10.

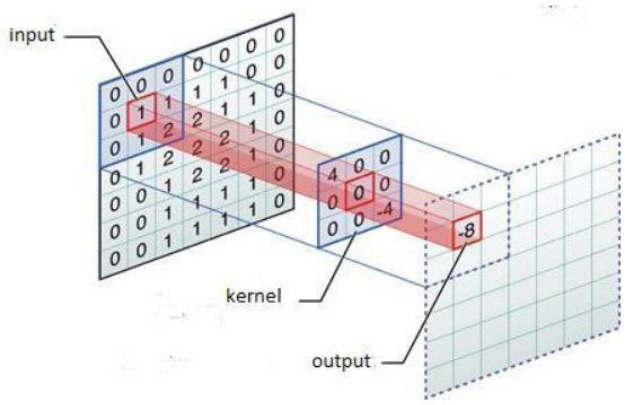


Figure 3.10 Image source

The convolution activity occurred between the created matrix and a part of the picture. If the value of the result is large, it means that there are available features in that part of the picture. Alternatively, if the result did not present the value, the result is depressed.

A created matrix that is responsible for checking for right-hand runs over a slice of the picture is seen in Figure 3.11.

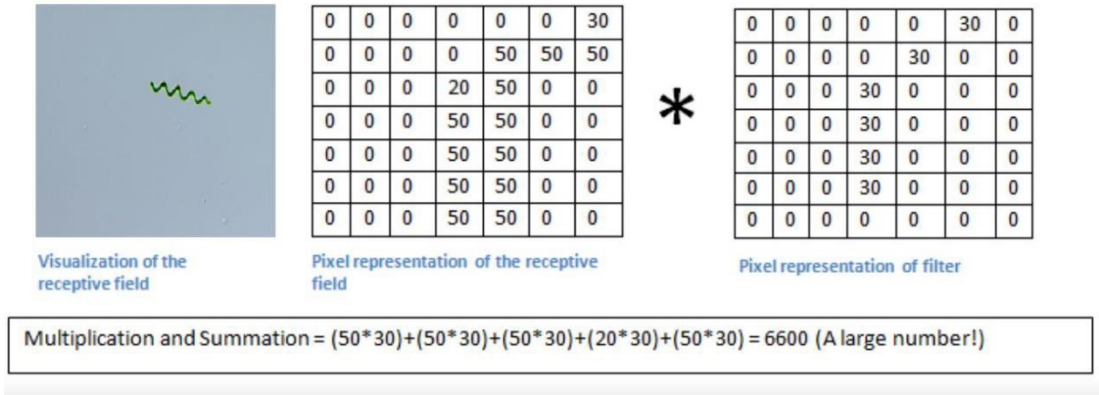


Figure 3.11 Image source

When the equivalent matrix runs over a piece of the picture with a significantly unique arrangement of edges, the convolution yields small outputs, implying that there was no solid nearness of a right hand curve.

The aftereffect of running this created matrix over the entire picture is a matrix output framework that stores the convolutions of this channel over different pieces of the

picture, as seen in Figure 3.12.The created matrix must have an indistinguishable number of channels from the information picture with the goal of multiplication.

Element-wise carry on, At that point, the channel must also contain three channels (RGB in our input image).

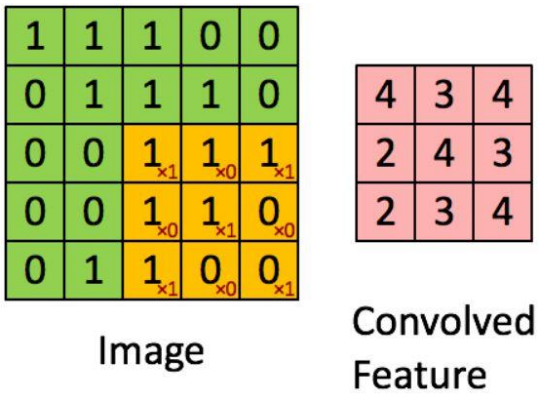


Figure 3.12 Matrix filter running over an image

Max pooling essentially decreases the description view size as well as the measure of memory and the number of tasks performed later in the system. The size output of the maximum pooling task can be determined by utilizing the following condition:

$$n_{out} = floor\left(\frac{n_{in}-f}{s}\right) + 1 \quad (5)$$

Where *n* indicates the element of the information input picture, *s* means the number of runs and *f* means the size of the window.

An additional advantage of max pooling is that it powers the system to concentrate on a couple of neurons rather than every one of them, which has a regularizing impact on the system, making it less inclined to over fit the preparation of information.

The maximum pooling task comes down to the for-loop function and the other three for the while loop functions.

The for-loop is utilized to run through each layer of the information picture, and the while-circles slide the window over all aspects of the picture.

After various convolutional layers and down inspecting tasks, the 3D picture description is changed over into an element vector that is passed into a Multi-Layer

Perceptron, which simply is a neural system within an event of three layers. This is applied to as a Fully-Connected Layer.

In the completely associated activity of a neural system, the information description is smoothed into a component vector and passed through a system of neurons to output prediction probabilities. The next picture represents the levelling task.

The lines are linked to shape with a long element vector. In the event that numerous information layers are available, its lines are likewise linked to shape a much longer component vector.

The yield layer of a CNN is responsible for creating the likelihood of each class (every digit) given the information picture. To acquire these probabilities, we introduce our last dense layer to contain an indistinguishable number of neurons as classes being available. The yield of this thick layer at that point passes through the Softmax enactment work, which maps all the last thick layer yields to a vector whose components total to one, thus:

$$\sigma(x_j) = \frac{e^{xj}}{\sum_i e^{xi}} \tag{6}$$

Here, *x* indicates every part in the last panel's outputs.

We determine the loss to gauge how precise our system was in determining the spirulina from the picture by utilizing a loss function. The loss function capacity assigns a genuine esteemed number to characterize the model's precision when anticipating the spirulina yield. A typical loss function capacity to utilize when predicting various yield classes is the Categorical Cross-Entropy, characterized as follows:

$$H(y, y) = \sum_\iota y_\iota \log \acute{\frac{1}{\acute{y}_\iota}} = -\sum_\iota y_\iota log \hat{y_\iota} \tag{7}$$

Here, $\hat{y}$ is convolution neural network prediction. To load our data to our system and obtain results, the following steps are taken.

First Input Data, the spirulina preparation and test image sets are acquired from our spirulina generated images file. The documents store the pictures and mark and label the information as tensors, so the records must be followed.

Second Starting the Methods, We initially characterize the techniques to introduce both the channels for the convolutional layers and the loads for the dense layers. To make for a smoother preparation process, we instate each channel with a standard deviation of 1 and a mean of 0.

Third Activities of Backpropagation, to register the directions that will drive the system to refresh its weights, we have to characterize the strategies that back spread the angles through the max-pooling layers and convolutional layers.

Finally Network Building, to characterize a strategy that joins the forward and reverse activities of a convolutional neural system, the system's parameters and hyper parameters are taken as sources of information and it releases the angles.

CHAPTER 4

RESULTS

4.1 Color Based Study Results

The results of the detection by color are given in Table 4.1. We calculated the sensitivity and the ability of a test to label an object accurately. All measurements are shown in Table 4.2. The results indicate that the color based detection method can detect spirulina in water with an (TP+TN)/ Total Data set, (392+453) /1000 = 84-percent success rate. Sensitivity calculated as (TP/ (TP+FN)) (392/ (392+49)) = 88. Specificity calculated as (TN/ (TN+FP)) (453/ (453+106)) = 81.

Table 4.1 Results of Spirulina Detection using color extraction

TP	FP	Total Test Positive
392	106	498
FN	TN	Total Test Negative
49	453	502
Total	Total	Total
441	559	1000

Table 4.2 Sensitivity and Specificity of Spirulina Detection using color extraction results.

Sensitivity	Specificity
0.88	0.81

4.2 FAST Features Detection Results

The results of the experiment are shown in Table 4.3. An example of our table content here for the first column is the original image example of image A. The second column shows the modified image example modified A and its number in column 3 which are

125 modified images, fourth column is number of FAST features is extracted from original image A which is 29, column five is average total number of features extracted from modified images and her for modified A total average of all images features is 98 Column six is matching which is the total number of matching images is found in her A original gets matches with A modified images 67 times The last column.

Table 4.3 Results of the FAST method.

Image Detector			FAST		
images			Features		Matches
Original image	Modified image	#Modified image	Original image	Modified image	
A 	A	125	29	98	67
	BCD	375	29	354	6
	X	500	29	220	0
B 	B	125	21	58	48
	ACD	375	21	403	2
	X	500	21	220	0
C 	C	125	37	62	23
	ABD	375	37	324	0
	X	500	37	220	0
D 	D	125	10	38	38
	ABC	375	10	103	1
	X	500	10	220	0

The results of the FAST Feature method are presented in Table 4.3. Here, we used Type A for Laxa, Type B for Major, Type C for Nordest and Type D for Princeps as the original image input. Moreover, we mention each type as having 6 original images and we took one from the six original images as matching headers for every type randomly. Then we separated the modified images into folders. As examples inside the modified image column, we see A, which contains 125 modified images of mixed the stander her is the all images contains different shapes but must contains A type of spirulina in every image. This will also apply for B, C and D, and all matches of the modified images in our dataset are shown.

Table 4.4 and Table 4.5 show summaries of the Validation, Specificity, and Sensitivity experiments for color detection. Accuracy calculated (TP+TN)/Total Data set, (176+500)/1000 = 67 percent success rate. Sensitivity calculated as (TP/(TP+FN)) (176/(176+8))=95. Specificity calculated as (TN/(TN+FP)) (500/(500+315))=61.

Table 4.4 Results of Spirulina Detection using FAST.

TP 176	FP 316	Total Test Positive 491
FN 8	TN 500	Total Test Negative 509
Total 186	Total 816	Total 1000

Table 4.5 Sensitivity and Specificity of Spirulina Detection using FAST.

Sensitivity	Specificity
0.95	0.61

As presented in Table 4.5, the Specificity is 0.95 with only 8 negative false results, while all the spirulina objects were 100% detected correctly and successfully.

4.3 SURF Features Detection Results

The results of the experiment are shown in Table 4.6 example of our table content here for the first column is the original image example of one image of A. The second column has modified image examples of modified type (A) and numbering 125 modified images in Column 3. The fourth column is the number of SURF features that are extracted from original image A, which is 13. Column 5 is the average total number of features extracted from the modified images, and here for the modified A, the total average of all image features is 125. Column 6 matches, which is the total number of matching images found in her type (A) original gets matches with A modified images 57 match.

Table 4.6 Results of the SURF method.

Image Detector			SURF		
images			Features		Matches
Original image	Modified image	#Modified image	Original image	Modified image	
A 	A	125	13	87	57
	BCD	375	13	180	2
	X	500	13	126	0
B 	B	125	9	132	26
	ACD	375	9	179	0
	X	500	9	126	0
C 	C	125	15	87	31
	ABD	375	15	165	0
	X	500	15	126	0
D 	D	125	6	168	24
	ABC	375	6	176	1
	X	500	6	126	0

The results of SURF Feature method are shown in Table 4.6. Here, we used Type A for Laxa, Type B for Major, Type C for Nordest and Type D for Princeps as original image inputs. Moreover, we observe that each type has six original images and we took one from the six original images as a matching header for every type randomly. Then we separated the modified images into folders. As examples inside the modified image column, we see A, which contains 125 modified images of mixed the stander her is the all images contains different shapes but must contains type A of spirulina in every image. This will also apply for B, C and D, and all matches of the modified images in our dataset are shown.

Table 4.7 and Table 4.8 show summaries of the Validation, Specificity, and Sensitivity experiments for color detection. Accuracy calculated (TP+TN)/ Total Data set, (138+500)/1000 = 63 percent success rate. Sensitivity calculated as (TP/ (TP+FN)) (138/ (138+3)) =97. Specificity calculated as (TN/ (TN+FP)) (500/ (500+359)) =58.

Table 4.7 Results of Spirulina Detection using Deep Learning Third Run.

TP 138	FP 359	Total Test Positive 497
FN 3	TN 500	Total Test Negative 503
Total 141	Total 859	Total 1000

Table 4.8 Sensitivity and Specificity of Spirulina Detection results in the third run.

Sensitivity	Specificity
0.97	0.58

As presented in Table 4.8, the Specificity is 0.97 with only 3 negative false results, while all the Spirulina objects were 100% detected correctly and successfully.

4.4 Deep Learning Results

Using the datasets and the method introduced in the previous section, some experiments were performed by varying the number of samples for training and by cross-validating to check the performance of the convolutional neural networks in this problem. Hence, the training was based on using the same number of samples per species throughout the 4 classes, resulting in the experiments being balanced. The results re illustrated in validation tables presenting the calculated sensitivities and specificities.

4.5 Deep Learning First Run

Table 4.9 and Table 4.10 show summaries of the Validation, Specificity, and Sensitivity experiments for the first run on the CNN theory with the prepared dataset. In this case, 200 of images were randomly selected as the input of the system. However, we ensured that the input images for each run differed from one another.

Table 4.9 Results of Spirulina Detection using Deep Learning First Run.

TP 100	FP 0	Total Test Positive 100
FN 9	TN 91	Total Test Negative 100
Total 109	Total 91	Total 200

Table 4.10 Sensitivity and Specificity of Spirulina Detection results in the first run.

Sensitivity	Specificity
1.00	0.91

As presented in Table 4.10, the Specificity is 0.91 with only 9 negative false results, while all the spirulina objects were 100% detected correctly and successfully.

4.6 Deep Learning Second Run

Table 4.11 and Table 4.12 show summaries of the Validation, Specificity, and Sensitivity experiments for the second run on the CNN theory with the prepared dataset. In this case, another 200 of images were randomly selected as the input of the system. However, we ensured that the input images for each run differed from one another.

Table 4.11 Results of Spirulina Detection using Deep Learning Second Run.

TP 99	FP 1	Total Test Positive 100
FN 0	TN 100	Total Test Negative 100
Total 99	Total 101	Total 200

Table 4.12 Sensitivity and Specificity of Spirulina Detection results in second run.

Sensitivity	Specificity
0.99	1.00

As presented in Table 4.12, the Sensitivity is 0.99 with only one false positive result, while all the spirulina objects detected 99% successfully and correctly.

4.7 Deep Learning Third Run

Table 4.13 and Table 4.14 show summaries of the Validation, Specificity, Sensitivity experiments for the third run on the CNN theory with the prepared dataset. In this case, another 200 of images were randomly selected as the input of the system. However, we ensured that the input images for each run differed from one another.

Table 4.13 Results of Spirulina Detection using Deep Learning Third Run.

TP 100	FP 0	Total Test Positive 100
FN 0	TN 100	Total Test Negative 100
Total 100	Total 100	Total 200

Table 4.14 Sensitivity and Specificity of Spirulina Detection results in the third run.

Sensitivity	Specificity
1.00	1.00

As presented in Table 4.14, Sensitivity is 100% with no False Positives, while all the spirulina objects were detected 100% correctly and successfully.

4.8 Deep Learning Fourth Run

Table 4.15 and Table 4.16 show summaries of the Validation, Specificity, Sensitivity experiments for the fourth run on the CNN theory with the prepared dataset. In this case, another 200 of images were randomly selected as the input of the system. However, we ensured that the input images for each run differed from one another.

Table 4.15 Result of Spirulina Detection using Deep Learning Fourth Run.

TP 100	FP 0	Total Test Positive 100
FN 0	TN 100	Total Test Negative 100
Total 100	Total 100	Total 200

Table 4.16 Sensitivity and Specificity of Spirulina Detection results in the fourth run.

Sensitivity	Specificity
1.00	1.00

As presented in Table 4.16, the Sensitivity is 100% with no False Positive results, while all the spirulina objects detected correctly and successfully.

4.9 Discussion

As spirulina detection is a unique topic, there were not any related works in this area. Every method in thesis was unique and none had ever been previously applied.

In this section, we discuss the method has been applied to spirulina and the results.

The first method was the color based detection wherein every measurement is shown in Table 4.1 the results indicate that the color based detection method can detect spirulina in water at an 84-percent accuracy rate of success. The error rate occurred due to there being all low-level green in the images and the presence of environmental noise with high levels of green in the images.

This thesis presented the specific feature detection algorithm for detection spirulina from water, the common feature detection algorithms are SIFT, HOG, SURF and FAST as SURF more advance from SIFT and better for corner detection and its upgrade from SIFT and because SURF is one of the most and popular interest point detector and descriptor and spirulina contains texture features SURF used in this thesis, and the other algorithms as HOG is not used because of disadvantage of HOG as it is very sensitive to image rotation and because Spirulina in water has rotation we will use FAST and SURF. The second method to detect spirulina is using the FAST feature based detection, we were unable to detect all the spirulina from our modified images. In fact, there were issues detecting the D type spirulina from which we were able to detect only eight features from the original image. We were only able to achieve and it was very low for detecting spirulina in water, a result lower than 70% accuracy.

In the third method, the SURF feature based detection, we were unable to detect all the spirulina from our modified images. There were problem detecting the D type spirulina, we were able to detect only 9 features from the original image and detection rate only 24 image from 125 which was very low for detecting spirulina in water, also a result lower than 70% accuracy. Finally, for the results of detections based on Deep Learning, we presented the system and results of four runs and we are randomly selected a number of images as an input of the system; however, we ensured that the input images for each run differed from one another. As presented in Table 4.10, Specificity is 0.91 with only 9 negative false results, while all the spirulina objects were detected correctly and successfully. In Table 4.12, Sensitivity is 0.99 with only

one False Positive result, while all the spirulina objects detected correctly and 99% successfully. Table 4.10 shows that Sensitivity is 100% with no False Positive results, while all the spirulina objects detected correctly and successfully. In Table 4.12, it can be observed that Sensitivity is 100% with no False Positive results, while all the spirulina objects detected correctly and successfully. The results from the first run of our system are shown in Table 4.9, from which it can be observed that nine error detections occurred as true negatives but where no spirulina in the image could be detected by the system. For the second run, Table 4.11 shows that the system received only one false positive wherein there was no spirulina in the images and the system resulted in an output as spirulina.

N fault cross validation used to calculate the accuracy in last method witch is CNN, in general the method run for 10 time but in our case in the 4th run there were no change in result we could detect all Spirulina images with 100% accuracy and the system output same result in last 3 runs, the table 4.17 shows all related work with each method and accuracy result.

Table 4.17 Related work and results summary.

Year and reference	samples	method	accuracy
2007[5]	2000	Morphological	87%
2011[6]	1000	Morphological (circular diatoms)	86.5%
2002[7]	600	Color Detection	83.4%
2017[9]	2400	Feature Detection	98%
2012[10]	1093	Morphological	96.17%
2002[11]	781	Bagging Tree classifier	96.9%
2003[12]	500	Multiple Discriminant Analysis	80.3%
2016[13]	10000	Feature Detection	94.7%
Proposed	1000	Color	80%
Proposed	1000	SURF	63%
Proposed	1000	FAST	67%
Proposed	1000	CNN	99%

The purpose of these results compared with related work on diatom detection in water were presented by Du Buf and Bayer [8], who proposed Diatom Identification using RGB level formatted images in extracting diatoms from images. They reported an accuracy rate of 83.4%, the rate lower than the deep learning approach results being

due to the environment of the image such that if the images were noisy with objects in the environment, the false rate of the color detection method would be high.

Another comparison with related work on diatom detection is Du Buf H (2002), who used the Bagging Tree classifier with over 781 samples at an accuracy of 96.9% [11]. However, this paper only covered circular diatoms such that texture features were not considered as our spirulina species was zigzag shaped. The spirulina detection based on color with a ratio being 84% resulted in successful detection.

In another comparison between the proposed results in this thesis, the color-based detection success rate was 84%, which was best result it could have had from other feature extraction detection. However, the disadvantage of using color detection in spirulina extraction can be that results contain noise images that possibly lead to false detection. The results of feature extraction detection and the rate successful detection where only 15 matches are detected out of 39 images for both methods (SIFT and FAST) means successful detection rates occur at less than 70% accuracy.

The results obtained in this study indicate that Deep Learning based detection is a successful method to detect spirulina in water since Deep Learning calculations, as discussed previously, attempt to take in abnormal features highlights from information in a steady manner. This circumvents the need for specialty and hard-core feature descent.

CHAPTER 5

CONCLUSIONS

In this thesis, four alternative methods for automatic detection of spirulina in water were experimented with, and the results have been presented. Our first three methods used the old approach to match spirulina images with spirulina used as the original images in addition to every modified image being tested for a match. Our results in the color-based method detection exceeded an 84-percent successful detection rate, which is evaluated to be satisfactory for automatic spirulina detection in water. The disadvantage of color detection is its sensitivity to the color green. In cases where there are green non-spirulina objects, the level of success of the method would be considerably affected. The results of the second spirulina detection method using FAST features were almost satisfactory in A1 and A2, A3, A4. However, we obtained a very low number of features such that there were only 5 matches in A5 out of 33 modified images. The results of the SURF experiment are quite unsatisfactory. A match even between identical types of spirulina was hardly observed. In brief, the SURF method does not seem to be a good prospect for spirulina detection in water. The results obtained in this study indicate color-based detection as the most successful method to detect spirulina in water. In this thesis, we covered spirulina image detection in water using CNN and image processes in addition to methodology and dataset preparation being covered and artificial image generation being built. In this thesis, the main problems were discussed as generally the input image being provided with the features of images and identical rotations, and secondly in the architectures of the dropout layers. The CNN technique will be applied for the first time for the detection of spirulina in water by this planned study. The CNN methodology shows that spirulina detection can always be taken with a large number of datasets. In this thesis, the dataset created 1000 spirulina images for training and testing. CNN was used for the issue of spirulina order and this has revealed some intriguing results, such as its invariance to picture from various preparation strategies. For the results, we presented a system and

the results of four runs, we randomly selected a number of images as the input of the system; however; we ensured that the input images for each run differed from one another.

The results in the first run of our system are validated in Table 4.9 and their sensitivity and specificity are mentioned in Table 4.10. There were 9 errors detected as true negatives as spirulina in the images which the system could not detect. For the second run in Table 11, the system yielded only one false positive for which there was no spirulina in the image but the system result output was spirulina. The results in the third and fourth runs were identical to the no false output detection with 100% sensitivity and specificity.

Concerning future work, a further step would be to construct an automated spirulina detection system by tuning the CNN. Therefore, neither a biology specialist nor manual labelling would be required to monitor spirulina in water.

This examination can be considered a starter towards the improvement of PC programming that can distinguish and perceive diatomaceous pollution in green growth. Another procedure yet to be created is to check the number of algae to comprehend the dimension of sullying.

Concerning the work above and beyond is accomplish arrangement as well as location as well. Along these lines, the technique would be accessible to determine spirulina position and species over a Whole Slide Image (WSI). This can be accomplished, for instance, by utilizing Region-based Convolutional Neural Networks (CNNs).

REFERENCES

[1] H. Buf, M. Bayer, “Automatic Diatom Identification,” New Jersey, London, World Scientific. 2002.

[2] G. Drebes, “The Biology of Diatoms,” Volume 13 of Botanical Monographs. University of California Press. pp. 250–283. ISBN 978-0-520-03400-6. Retrieved 14 November 2013.

[3] Stoermer, E. Smol, J. “The Diatoms: Applications for the Environmental and Earth Sciences”. Cambridge University Press. 2004.

[4] A. Rola, J. Pulhin, R. Hall, “Water Resources in the Philippines: Overview and Framework of Analysis,” Water Policy in the Philippines. Global Issues in Water Policy, vol 8. Springer, Cham, ISBN 978-3-319-70968-0, 30 January 2018.

[5] R. Edward, D. Tom, “Machine learning for high speed corner detection,” In 9th European Conference on Computer Vision, vol. 1, pp. 430–443. 2006.

[6] G. Yahui, L. Jinfei, C. Changping, J. Chen, L. Junrong, Y. Chenhui, “Automatic Identification of Diatoms with Circular Shape using Texture Analysis,”. 2011.

[7] H. Du Buf and M. Bayar, "ADIAC achievements and future work," in Automatic Diatom Identification. vol. 51, J. M. H. D. Buf and B. M. M., Eds. Singapore: World Scientific Publishing Company, pp. 289-298.

[8] E. Rosten, T. Drummond, “Machine learning for high speed corner detection,” in 9th European Conference on Computer Vision, vol. 1, pp. 430–443. 2006.

[9] G. Bueno, O. Deniz, A. Pedraza, J. Salido, G. Cristobal, B. Saul, “Automated Diatom Classification (Part A): Handcrafted feature approaches,” Appl. Sci, in press. 2017.

[10] I. Dimitrovski, D. Kocev, S. Loskovska, S. Dzeroski, “Hierarchical classification of diatom images using ensembles of predictive clustering trees,”. Ecol. Inform. 2012, 7, 19–29.

[11] H. DuBuf, M. Bayer, “Automatic Diatom Identification; Seriesin Machine Perception and Artificial Intelligence,” World Scientific Publishing Co, Munich, Germany, 2002.

[12] J. Pappas, E. Stoermer, “Legendre shape descriptors and shape group determination of specimens in the Cymbella cistula species complex,” Phycologia 2003, 42, 90–97.

[13] Q. Lai, K. Lee, A. Tang, K. Wong, H. So, K. Tsia, “High-throughput time-stretch imaging flow cytometry for multi-class classification of phytoplankton,” Opt. Express 2016, 24, 28170–28184.

[14] P. Chakravorty, "What Is a Signal" IEEE Signal Processing Magazine, vol. 35, no. 5, pp. 175-177, Sept. https://doi.org/10.1109/MSP.2018.2832195. 2018.

[15] G. Rafael, “Digital Image Processing,” 3rd. Pearson Hall. ISBN 9780131687288. 2008.

[16] A. Logvinenko, “The geometric structure of color,” Journal of Vision, 15(1), 15.1.16. http://doi.org/10.1167/15.1.16. 2015.

[17] D. Anderson, “Color Spaces in Frame Grabbers: RGB vs. YUV,” Retrieved 2008-04-08.

[18] G, Hans. “Industrial Color Testing: Fundamentals and Techniques,” Wiley-VCH. ISBN 3-527-30436-3. 2001.

[19] G. Buxbaum, G. Pfaff, “Industrial Inorganic Pigments,” Wiley-VCH. ISBN 3-527-30363-4. 2001.

[20] Agoston, K. Max, “Computer Graphics and Geometric Modeling: Implementation and Algorithms,” London: Springer. pp. 300–306. ISBN 978-1-85233-818-3. 2003.

[21] C. Heng, J. Xihua, S. Angela, W. Jingli, “Color image segmentation: Advances and prospects,” Pattern Recognition. 34 (12): 2259. CiteSeerX 10.1.1.119.2886. doi:10.1016/S0031-3203(00)00149-7.

[22] S. Alvy, “Color gamut transform pairs,” Computer Graphics. 12 (3): 12–19. doi:10.1145/965139.807361. 1978.

[23] W. Michael, B. William, C. Beatty, “An experimental comparison of RGB, YIQ, LAB, HSV, and opponent color models,” ACM Transactions on Graphics. 6 (2): 123–158. doi:10.1145/31336.31338.

[24] J. Stephen, “Stephen Johnson on Digital Photography,” O'Reilly. ISBN 0-596-52370-X. 2006.

[25] P. Charles, “Rehabilitation of gamma,” Photonics West'98 Electronic Imaging. International Society for Optics and Photonics, 1998.

[26] I. Stokes, M. Anderson, S. Chandrasekar, and R. Motta, “A Standard Default Color Space for the Internet – sRGB,” 2014.

[27] W. Burger, J. Mark, “Principles of Digital Image Processing Core Algorithms,” Springer Science & Business Media. pp. 110–111. ISBN 978-1-84800-195-4. 2010.

[28] S. Sahu, M. Mittal, “Conversion of a Color Image to a Binary Image,” CoderSource.net. 2005-04-18. Archived from the original on 2008-06-10. Retrieved 2008-06-11.

[29] D. Evanson, “Photoshop Fundamentals: Working in Different Color Modes,” Graphics.com. Retrieved 2017-10-28.

[30] S. Vavilis, E. Paredes, S. Kostas, “A Hybrid Binarization Technique for Document Images,” Learning Structure and Schemas from Documents, Springer Berlin Heidelberg, pp. 165–179, ISBN 9783642229121, retrieved 2019-04-28. 2011.

[31] D. Lowe, “International jornal of computer vision,” vol. 60, no. 2, 2004, pp. 91.

[32] E. Rosten, T. Drummond, “Machine learning for high speed corner detection,” in 9th European Conference on Computer Vision, vol. 1, 2006, pp. 430–443.

[33] R Edward, D. Tom, “Machine Learning for High-speed Corner Detection,” Published in ECCV 2006, pp 430-443, DOI:10.1007/11744023_34.

[34] D. Rosten, T. Edward, “Machine Learning for High-speed Corner Detection,” 2006.

[35] S. Brahmbhatt, “Practical OpenCV”, Berkeley, CA, Apress, 2013.

[36] E. Rosten, G. Reitmayr, T. Drummond, Real-time Video Annotations for Augmented Reality. 2000.

[37] B. Herbert, E. Andreas, T. Tinne, V. Luc, V, "Speeded Up Robust Features Computer Vision and Image Understanding (CVIU)," Vol. 110, No. 3, pp. 346–359. 2008.

[38] H. Bay, A. Ess, T. Tuytelaars, and L.V. Gool, "Speeded Up Robust Features," ETH Zurich, Katholieke Universiteit Leuven. 2016.

[39] B.J. David, Y. Brenier, "A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem," Numerische Mathematik. 84 (3): 375–393. doi:10.1007/s002110050002. 2002.

[40] T. Lindeberg, "Image matching using generalized scale-space interest points," Journal of Mathematical Imaging and Vision, volume 52, number 1, pages 3-36. 2015.

[41] H. Bay, T. Tuytelaars, L.V. Gool, "SURF: Speeded Up Robust Features," Proceedings of the 9th European Conference on Computer Vision, Springer LNCS volume 3951, part 1. pp. 404–417. 2017.

[42] A. Mordvintsev, K. Abid, "OpenCV-Python TutorialsDocumentation," November 18, 2013.

[43] S. Balnco, E. Becares, "Diagnosis in the duero basin (Spain),". Archiv fur Hydrobiologie. 17(3-4):267-286, 2017.

[44] S. Blanco, L. Ector, and E. Becares, "Epiphytic diatoms as water quality indicators in spanish shallow lakes," Vie et Milieu, 54(2-3):71-80, 2014.

[45] S. Blanco and E. Becares. Are biotic indices sensitive to river toxicants? a comparison of metrics based on diatoms and macro-invertebrates. Chemosphere, 79(1):18-25, 2010.

[46] S. Blanco, C. Cejudo-Figueiras, L. Tudesque, E. Becares, L. Hoffmann, and L. Ector, "Are diatom diversity indices reliable monitoring metrics," Hydmbiologia, 695(1):199-206, 2012.

[47] S. Blanco, C. Cejudo-Figueiras, I. Alvarez-Blanco, E. Van Donk, E.M. Gross, L.A. Hansson, K. Irvine, E. Jeppesen, T. Kairesalo, B. Moss, et al. Epiphytic diatoms

along environmental gradients in western european shallow lakes. CLEAN—Soil, Air; Water, 42(3):229-235, 2014.

[48] C. Cejudo-Figueiras, S. Blanco, I. Alvarez-Blanco, L. Ector, and E. Becares, "Nutrient monitoring in spanish wetlands using epiphytic diatoms," Vie a milieu, 60(2):89-94, 2011.

[49] EU WFD. Directive 2000/60/EC. Establishing a framework for Community action in the field of water policy. Technical report, The European Parliament and the Council of the European Union, 2000.

[50] H. du Buf, M.M. Bayer, "Automatic Diatom Identification," Series in Machine Perception and Artificial Intelligence. 2002.

[51] R. Henrikson, "Microalga Spirulina, superalimento del future," Ronore Enterprises. 2t ed. Editions Urano, Barcelona. Esparia.1994, pp. 222.

[52] K.A. Shabana, M.S. Arabi, "Spirulina An Overview," International Journal of Pharmacy and Pharmaceutical Sciences, School of Bio Sciences and Technology, VIT University, VOL 4, ISSUE 3, 2012, ISSN-0975-1491. 2012.

[53] S.J.M. Droop, P.A. Sims, D.C. Mann, and R.J. Pankhurst, "A taxonomic database and linked iconograph for diatoms," pages 503-508. Springer Netherlands, Dordrecht, 1993.

[54] J.P. Smol and E.F. Stoermer. The diatoms: applications for the environmental and earth sciences. Cambridge University Press, 2010.

[55]] H. Du Buf, M. Bayer, S. Droop, R. Head, S. Juggins, S. Fischer, H. Bunke, J. Roerdink, "Diatom identification," a double challenge called adiac. In Image Analysis and Processing, 1999. Proceedings. International Conference on, pages 734-739. IEEE, 1999.

[56] H. du Buf and M.M. Bayer. Automatic Diatom Identification. Series in Machine Perception and Artificial Intelligence. 2002.

[57] J.L. Pappas, E.F. Stoermer, "Legendre shape descriptors and shape group determination of specimens in the cymbella cistula species complex," Phycologia, 42(1):90-97, 2003.

[58] D. Mou, E.F. Stoenner, S. tabellaria, "Shape groups based on fourier descriptors," Journal of Phycology, 28(3):386-395, 1992.

[59] E. Falasco, S. Blanco, F. Bona, J. Goma, D. Hlubikova, M.H. Novais, L. Hoffmann, and L. Ector. Taxonomy, morphology and distribution of the sellaphora stroemii complex (bacillariophyceae). Fottea,9(2):243-256, 2009

[60] C. Cejudo-Figueiras, E.A. Morales, C.E. Wetzel, S. Blanco, L. Hoffmann, and L. Ector. Analysis of the type of fragilaria construens var. subsalina (bacillariophyceae) and description of two morphologically related taxa from europe and the united states. Phycologia, 50(1):67-77, 2011.

[61] I. Dimitrovski, D. Kocev, S. Loskovska, and S. Dzeroski. Hierarchical classification of diatom images using ensembles of predictive clustering trees. Ecological Informatics, 7(I):19-29,2012.

[62] M. Coste, S. Boutry, J. Tison-Rosebery, and F. Delmas. Improvements of the biological diatom index (bdi): Description and efficiency of the new version (bdi-2006). Ecological Indicators, 9(4):621-650, 2009.

[63] T. Fawcett, "An introduction to roc analysis," Pattern recognition letters, 27(8):861-874, 2006.

[64] Y. Bengio, A. Courville, P. Vincent, "Representation Learning: A Review and New Perspectives," IEEE Transactions on Pattern Analysis and Machine Intelligence. 35 (8): 1798–1828. arXiv:1206.5538. doi:10.1109/tpami.2013.50. PMID 23787338.

[65] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview," Neural Networks. 61: 85–117. arXiv:1404.7828. doi:10.1016/j.neunet.2014.09.003. PMID 25462637. 2015.

[66] B. Yoshua, L. Yann, H. Geoffrey, "Deep Learning," Nature. 521 (7553): 436–444. Bibcode:2015Natur.521..436L. doi:10.1038/nature14539. PMID 26017442. 2015.

[67] C. Dan, U. Meier, J. Schmidhuber, "Multi-column deep neural networks for image classification," 2012 IEEE Conference on Computer Vision and Pattern Recognition: 3642–3649. arXiv:1202.2745. doi:10.1109/cvpr.2012.6248110. ISBN 978-1-4673-1228-8. June 2012.

[68] K. Alex, S. Ilya, H. Geoffry, “ImageNet Classification with Deep Convolutional Neural Networks,” Neural Information Processing Systems, NIPS 2012.

[69] M. Adam, W. Greg, K. Konrad, “Toward an Integration of Deep Learning and Neuroscience,” Frontiers in Computational Neuroscience. 10: 94. doi:10.3389/fncom.2016.00094. PMC 5021692. PMID 27683554. 2016.

[70] B.A. Olshausen, “Emergence of simple-cell receptive field properties by learning a sparse code for natural images,” Nature. 381 (6583): 607–609. Bibcode:1996Natur.381..607O. doi:10.1038/381607a0. PMID 8637596. 1996.

[71] B. Yoshua, L. Dong-Hyun, B. Jorg, M. Thomas, L. Zhouhan (2015-02-13). "Towards Biologically Plausible Deep Learning". arXiv:1502.04156 [cs.LG].

[72] L. Deng, D. Yu, “Deep Learning: Methods and Applications,” Foundations and Trends in Signal Processing. 7 (3–4): 1–199. doi:10.1561/2000000039. 2014.

[73] B. Yoshua, “Learning Deep Architectures for AI,” Foundations and Trends in Machine Learning. 2 (1): 1–127. CiteSeerX 10.1.1.701.9550. doi:10.1561/2200000006.

[74] L. Yann, B. Yoshua, H. Geoffrey, “Deep learning,” Nature. 521 (7553): 436–444. Bibcode:2015Natur.521..436L. doi:10.1038/nature14539. PMID 26017442. 28 May 2015.

[75] G. Alex, S, Jürgen, B. Yoshua, S. Lafferty, W. Chris, C. Aron, “Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks,” Neural Information Processing Systems (NIPS) Foundation. Curran Associates, Inc: 545–552. 2009.

[76] D. Hebb, “The Organization of Behavior,” A Neuropsychological Theory.Wiley, New York, NY. 1949.

[77] M. Marvin, P. Seymour, “Perceptrons: An Introduction to Computational Geometry,” MIT Press. ISBN 978-0-262-63022-1. 1969.

[78] D.E Rumelhart, M. James, “Parallel Distributed Processing: Explorations in the Microstructure of Cognition,” Cambridge: MIT Press. ISBN 978-0-262-63110-5. 1986.

[79] M. Keith, L. James, G. Georges, H. Philip, W. Lele, K. Theodore, G. Ludwig, S. Richard, H. James, "Photovoltaic retinal prosthesis with high pixel density," Nature Photonics. 6 (12): 391–397. Bibcode: 2012NaPho.6391M. doi:10.1038/nphoton.2012.104. ISSN 1749-4885. PMC 3462820. PMID 23049619.

[80] H. Jun, M. Claudio, "The influence of the sigmoid function parameters on the speed of backpropagation learning," In Mira, José; Sandoval, Francisco (eds.). From Natural to Artificial Neural Computation. Lecture Notes in Computer Science. 930. pp. 195–201. doi:10.1007/3-540-59497-3_175. ISBN 978-3-540-59497-0. 1995.

[81] S. Walker, D. Duncan, "Estimation of the probability of an event as a function of several independent variables," Biometrika. 54 (1/2): 167–178. doi:10.2307/2333860. JSTOR 2333860. 1967.

[82] W. Zhang, "Shift-invariant pattern recognition neural network and its optical architecture," Proceedings of Annual Conference of the Japan Society of Applied Physics. 1988.

[83] B. Moez, M. Franck, W. Christian, G. Christophe, B. Atilla, "Sequential Deep Learning for Human Action Recognition," In Salah, Albert Ali; Lepri, Bruno (eds.). Human Behavior Unterstanding. Lecture Notes in Computer Science. 7065. Springer Berlin Heidelberg. pp. 29–39. CiteSeerX 10.1.1.385.4740. doi:10.1007/978-3-642-25446-8_4. ISBN 978-3-642-25445-1. 2011-11-16.

[84] C. Ronan, J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," Proceedings of the 25th international conference on Machine learning. ACM, 2008.

[85] L. Yann, "LeNet-5, convolutional neural networks," Retrieved 16 November 2013.

[86] R. Vadim, "Appropriate number and allocation of ReLUs in convolutional neural networks," Research Bulletin of NTUU "Kyiv Polytechnic Institute". 1: 69–78. doi:10.20535/1810-0546.2017.1.88156. Retrieved 17 February 2019.

[87] C. Dan, U. Meier, J. Masci, L.M. Gambardella, J. Schmidhuber, "Flexible, High Performance Convolutional Neural Networks for Image Classification," Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence-Volume Volume Two. 2: 1237–1242. Retrieved 17 November 2013.

[88] D.H. Hubel, T.N. Wiesel, "Receptive fields of single neurons in the cat's striate cortex," J. Physiol. 148 (3): 574–91. October 1959.3

[89] Geisser, S. "Predictive Inference". New York, NY: Chapman and Hall. ISBN 978-0-412-03471-8. 1993.

[90] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Speeded Up Robust Features", ETH Zurich, Katholieke Universiteit Leuven, 10 September. 2008.

[91] L. Hou, D. Samaras, T. M. Kurc, Y. Gao, J. E. Davis, and J. H. Saltz, "Patch-based Convolutional Neural Network for Whole Slide Tissue Image Classification". Proc IEEE Comput Soc Conf Comput Vis Pattern Recognit. 2016 Jun-Jul; 2016: 2424–2433. PMC 2016 Oct 28.

[92] V. Andrearczyk, P. F. Whelan, "Using Filter Banks in Convolutional Neural Networks for Texture Classification" Vision Systems Group, School of Electronic Engineering, Dublin City University, Glasnevin, Dublin 9, Ireland. 2016.

[93] P. W. Blog, "How Many Images Do You Need to Train A Neural Network", https://petewarden.com/2017/12/14/how-many-images-do-you-need-to-train-a-neural-network/

[94] Larsen-Freeman, D. (2013). Transfer of Learning Transformed. Language Learning, 63, pp.107-129.

[95] M. Fatih Aydogdu and Vakkas Celik and M. Fatih Demirci "Comparison of Three Different CNN Architectures for Age Classification", Department of Computer Engineering, TOBB University of Economics and Technology. 2017.

[96] E. Ziga, D. Stepec, V. Struc, "Training Convolutional Neural Networks with Limited Training Data for Ear Recognition in the Wild". Faculty of Computer and Information Science, University of Ljubljana, Slovenia. Conference on Automatic Face and Gesture Recognition, – International Workshop on Biometrics in the Wild IEEE, 2017.