KTH ROYAL INSTITUTE
OF TECHNOLOGY

Degree Project in Information and Network Engineering

Second cycle, 30 credits

Reliable Detection of Water Areas in Multispectral Drone Imagery

A faster region-based CNN model for accurately identifying the location of small-scale standing water bodies

SHENGYAO SHANGGUAN

Reliable Detection of Water Areas in Multispectral Drone Imagery

A faster region-based CNN model for accurately identifying the location of small-scale standing water bodies

SHENGYAO SHANGGUAN

Master's Programme, Information and Network Engineering, 120 credits
Date: December 26, 2022

Supervisor: Hongbo Zhao
Examiner: Markus Flierl
School of Electrical Engineering and Computer Science
Swedish title: Tillförlitlig detektering av vattenområden i multispektrala drönarbilder
Swedish subtitle: En snabbare regionbaserad CNN-modell för noggrann identifiering av var småskaliga stående vattenförekomster finns

Abstract

Dengue and Zika are two arboviral viruses that affect a significant portion of the world population. The principal vector species of both viruses are Aedes aegypti and Aedes albopictus mosquitoes. They breed in very slow flowing or standing pools of water. It is important to reduce and control such potential breeding grounds to contain the spread of these diseases. This thesis investigates a model for the detection of water bodies using high-resolution images collected by Unmanned Aerial Vehicles (UAVs) in tropical countries, exemplified by Sri Lanka, and their multispectral information to help detect water bodies where larvae are most likely to breed quickly and accurately.

Although machine learning has been studied in previous work to process multispectral image information to obtain the location of water bodies, different machine learning methods have not been compared, only random forest algorithms have been used. Because Convolutional Neural Networks (CNNs) are known to provide advanced classification performance for visual recognition tasks, in this thesis, faster region-based CNNs are introduced to perform fast and accurate identification of water body locations.

In order to better evaluate the experimental results, this thesis introduces Intersection over Union (IoU) as a criterion for evaluating the results. On the one hand, IoU can judge the success rate of the model for water region recognition, and on the other hand, analysis of the model recall rate under different IoU values can also evaluate the model's ability to detect the range of water regions. Meanwhile, the basic CNN network and random forest algorithm in the previous work are also implemented to compare the results of faster region-based CNNs.

In conclusion, the faster region-based CNN model achieves the best results with a 98.33% recognition success rate for water bodies in multispectral images, compared to 95.80% for the CNN model and 95.74% for the random forest model. In addition, the faster region-based CNN model significantly outperformed the CNN model and the random forest model for training speed.

Keywords

Water Detection, Faster region-based convolutional neural networks, Multiple images, Convolutional neural networks, Random Forest

Sammanfattning

Dengue och zika är två arbovirala virus som drabbar en stor del av världens befolkning. De viktigaste vektorerna för båda virusen är myggorna Aedes aegypti och Aedes albopictus. De förökar sig i mycket långsamt rinnande eller stående vattensamlingar. Det är viktigt att minska och kontrollera sådana potentiella grogrunder för att begränsa spridningen av dessa sjukdomar. I denna avhandling undersöks en modell för att upptäcka vattenområden med hjälp av högupplösta bilder som samlas in av Unmanned Aerial Vehicles (UAV) i tropiska länder, exemplifierat av Sri Lanka, och deras multispektrala information för att hjälpa till att upptäcka vattenområden där larverna sannolikt förökar sig snabbt och noggrant.

Även om maskininlärning har studerats i tidigare arbeten för att bearbeta multispektral information från bilder för att få fram platsen för vattenförekomster, har olika metoder för maskininlärning inte jämförts, utan endast random forest-algoritmer har använts. Eftersom Convolutional Neural Networks (CNN) är kända för att erbjuda avancerade klassificeringsprestanda för visuella igenkänningsuppgifter i denna avhandling introduceras snabbare regionbaserade CNN för att utföra snabb och exakt identifiering av vattenkroppparnas läge.

För att bättre kunna utvärdera de experimentella resultaten införs i denna avhandling Intersection over Union (IoU) som ett kriterium för utvärdering av resultaten. Å ena sidan kan IoU bedöma modellens framgång för igenkänning av vattenområden, och å andra sidan kan analysen av modellens återkallningsfrekvens under olika IoU-värden också utvärdera modellens förmåga att upptäcka olika vattenområden. Samtidigt genomförs även det grundläggande CNN-nätverket och algoritmen för slumpmässig skog i det tidigare arbetet för att jämföra resultaten av Faster regionbaserad CNN.

Sammanfattningsvis ger den snabbare regionbaserade CNN-modellen de bästa resultaten med 98,33% av alla igenkänningsresultat för vattenkroppar i multispektrala bilder, jämfört med 95,80% för CNN-modellen och 95,74% för modellen med slumpmässig skog. Dessutom överträffade den snabbare regionbaserade CNN-modellen CNN-modellen och random forest-modellen avsevärt när det gäller träningshastighet.

Nyckelord

Vattendetektering, Snabbare regionbaserade konvolutionella neurala nätverk, Flera bilder, Konvolutionella neurala nätverk, Random Forest

Acknowledgments

First of all, I would like to thank Prof. Markus Flierl. As both my Examiner and my supervisor, he not only gave me the opportunity to do this interesting master's degree project but also gave me plenty of support and useful advice. In addition, his course "Image and Video Processing" also provided necessary and advanced knowledge for this project. And I also would like to thank Yasas Mahima from the University of Colombo, Sri Lanka for collecting data and providing advice.

Secondly, I would like to thank all the teachers who taught me in the past two years and any other employees in KTH. With your help and support, I can work hard to learn, expand my horizons, be exposed to diverse cultures, and improve myself in a comfortable and relaxed environment.

Additionally, I would like to thank all my friends and classmates in Sweden, Yicheng, Tianyu, Xiaoting, Hongting, Jia, Ziyi, Ruiqi, Xingyu, Yue, Yuchen, Rongfei, Zheng, Yuehao, Ruihan, Xiaoye, and so on. I cannot finish this master thesis without your accompanying and encouragement. Specifically, I would like to thank Yixiao for giving me suggestions and guidance in machine learning and having discussions with me when I met unexpected difficulties and Qianyao for supporting each other since we have something in common with the project.

Finally, I would like to give my most sincere thanks to my parents for always supporting me with the most selfless and unconditional love. I would dedicate the thesis to them and I believe I could be better with their love.

Stockholm, January 2023
Shengyao Shangguan

Contents

1 Introduction **1**
1.1 Background . . . 1
1.2 Purpose . . . 2
1.3 Goals . . . 2
1.4 Delimitations . . . 2
1.5 Structure of the thesis . . . 2

2 Background **5**
2.1 Multispectral image . . . 5
2.2 Resolution . . . 6
2.2.1 Spatial resolution . . . 6
2.2.2 Spectral resolution . . . 7
2.3 Index . . . 8
2.3.1 Normalized difference vegetation index (NDVI) . . . 8
2.3.2 Normalized difference water index (NDWI) . . . 9
2.4 Image classification . . . 11
2.4.1 Color feature-based classification . . . 11
2.4.2 Texture-based image classification . . . 11
2.4.3 Shape-based image classification . . . 12
2.4.4 Spatial relationship-based image classification . . . 13
2.5 Classification algorithm . . . 13
2.5.1 Random forest . . . 15
2.5.2 Convolutional neural network . . . 17

3 Methods **25**
3.1 Research process . . . 25
3.2 Data collection . . . 26
3.2.1 Sampling . . . 26
3.2.2 Sample size . . . 26

3.2.3 Social and ethical concerns . . . 28
3.3 Experimental design . . . 28
3.3.1 Test environment . . . 28
3.3.2 Hardware to be used . . . 28
3.4 Evaluation framework . . . 30

4 Experiment **33**
4.1 Faster region-based CNN . . . 33
4.2 CNN . . . 37
4.3 Random forest . . . 38

5 Results **39**
5.1 Faster region-based CNN . . . 39
5.2 CNN . . . 42
5.3 Random forest . . . 44

6 Conclusions and future work **45**
6.1 Conclusions . . . 45
6.2 Limitations . . . 45
6.3 Future work . . . 46
6.4 Reflections . . . 47

References **49**

List of Figures

2.1 Spectral range diagram [2] 6
2.2 Same picture with different spatial resolution[3] 7
2.3 NDVI of United States. Acquired on 2021-10-29[5] 9
2.4 NDWI of Italy. Acquired on 2020-08-01 [7] 10
2.5 Histogram of Figure 3.1 12
2.6 The basic process of integration learning 15
2.7 Example of Bagging algorithm using Bootstrapping to generate multiple sub-data 16
2.8 The specific process of the random forest algorithm 17
2.9 Object detection system overview using R-CNN[36] 21
2.10 Fast R-CNN architecture[38] 22
2.11 Basic structure of Faster R-CNN.[39] 24

3.1 Environment picture in Sri Lanka 26
3.2 Result of initial image 27
3.3 RedEdge-P high-resolution multispectral and RGB sensor . . . 28
3.4 Main multispectral bands of REDEDGE-P.[3] 29
3.5 DJI Phantom 4.[40] 29
3.6 DJI Phantom 4 with REDEDGE-P 30

4.1 Architecture of faster region-based CNN[42] 33
4.2 Total loss of epoch 27 of initial image 34
4.3 Result of sub-image 35
4.4 Total loss of epoch 27 of sub-image 36
4.5 Total loss of epoch 200 of sub-image 36

5.1 Results of faster region-based CNN 40
5.2 Bias of IoU 1 41
5.3 Bias of IoU 2 42
5.4 CNN Result of sub-image 43

5.5 CNN loss of tiles 43
5.6 Random forest result 44

6.1 Intersection of bounding boxes 47

List of acronyms and abbreviations

CART	Classification and Regression Tree
CNN	Convolutional Neural Network
FNN	Feedforward Neural Network
IFOV	Instantaneous Field of View
KNN	K-Nearest Neighbors
LDA	Linear Discriminant Analysis
LSTM	Long Short-Term Memory
MNDWI	Modified Normalized Difference Water Index
MSS	Multispectral Scanner
NDMI	Normalized Difference Moisture Index
NDVI	Normalized Difference Vegetation Index
NDWI	Normalized Difference Water Index
NIR	Near-Infrared
RCNN	Region Based-Convolutional Neural Network
ReLU	Rectified Linear Unit
SIANN	Shift-Invariant Artificial Neural Network
SVM	Support Vector Machine
SWIR	Short-wave Infrared
TM	Thematic Mapper
WTA	Winner Take All

Chapter 1

Introduction

1.1 Background

Dengue and Zika are two arboviral viruses that infect a large proportion of the world's population. Aedes aegypti and Aedes albopictus mosquitoes are the primary vector species for both viruses. They spawn in slow-flowing or stagnant pools of water. Restrictions and regulation of such potential breeding habitats are critical to keep these diseases from spreading.

Water bodies have different spectral characteristics from other substances. With the continuous maturity of remote sensing technology, the use of multispectral images for the identification of water bodies has achieved very good results. Reflection of water bodies is mainly in the blue-green light band, and absorption is strong in all other bands, especially in the near-infrared band. However, when water contains other substances, the reflection spectrum curve will change. When the water contains sediment, the reflectance of the visible band will increase, and the peak appears in the yellow-red region. When the water contains chlorophyll, the near infrared band is significantly increased.

With the continuous development of machine learning algorithms, their advantages in target recognition have led to a wide range of applications, and water body recognition is no exception. In this thesis, a faster region-based Convolutional Neural Network (CNN) algorithm is used to identify water bodies from multispectral images collected by drones and compared with CNN and random forest algorithms.

1.2 Purpose

This thesis is part of a larger project dedicated to using machine learning, drones, multispectral images, etc. to quickly and accurately identify water bodies in humans' daily environment. The research of this project helps to study and reduce the harm caused by related infectious diseases by discovering mosquito larval habitats in water bodies.

1.3 Goals

The goal of this project is to perform water body recognition on multispectral images collected by drones via faster region-based CNN and compare it with other possible methods. This has been divided into the following three sub-goals:

1. Construct a faster region-based CNN network to identify water areas from multi-spectral images taken by drones.
2. Construct a CNN network and a random forest algorithm to detect whether a smaller image is a water area.
3. Compare and analyze the results of different algorithms, draw conclusions, and put forward suggestions for future work.

1.4 Delimitations

This article only uses Faster region-based CNN to identify water bodies in multispectral images, and is not dedicated to improving the performance of Faster region-based CNN networks, nor does it compare the advantages and disadvantages of water body recognition in multispectral and RGB images. In addition, for the CNN and random forest algorithms used for comparison, this paper does not have a complicated design. It only discusses the feasibility of these two methods and compares and analyzes them with the results of Faster region-based CNNs.

1.5 Structure of the thesis

This thesis contains a total of 6 chapters. Chapter 2 presents relevant background information about multispectral images and image classification.

Chapter 3 presents methods used to solve the problem, the details of data processing and how to evaluate the result. Chapter 4 presents how the different algorithms are implemented. Chapter 5 presents results with analysis. Chapter 6 presents conclusions of the experiment and a discussion of future work.

Chapter 2

Background

2.1 Multispectral image

A multispectral image is an image that contains many bands, mostly between 3 and 15, sometimes only 3 bands (color images are an example) but sometimes much more, even a hundred, which is called hyper-spectral imaging [1]. Each band is a gray scale image that represents the brightness of the scene obtained according to the sensitivity of the sensor used to generate the band. In such an image, each pixel is associated with a set of values consisting of pixels in different bands, i.e., a vector. This string is then called the pixel's spectral marker. The human eye can see electromagnetic waves with wavelengths between 380 and 780 nanometers. Electromagnetic waves with wavelengths outside this range, such as infrared, are invisible to humans. Multispectral imaging allows us to obtain additional information that the human eye cannot see.

Multispectral cameras are used in the examination of printed circuit boards, the detection of counterfeit banknotes, and the characterization of skin in dermatology. The food business is one of the primary uses for multispectral cameras. Multispectral photographs help farmers determine the health of their crops. A drone with a multispectral camera flies over the field, collecting visual data. The data is then wirelessly sent to a ground-based computer. On a computer, the farmer sees the data in several shades, such as red and green. If the red area indicates sick plants, the farmer will know exactly which areas require pesticide application. This has huge benefits for farmers as well as the environment. Farmers save time and money, and healthy plants are not over-treated. Automated techniques such as food sorting are used in the food manufacturing process. Multispectral cameras can identify foreign objects

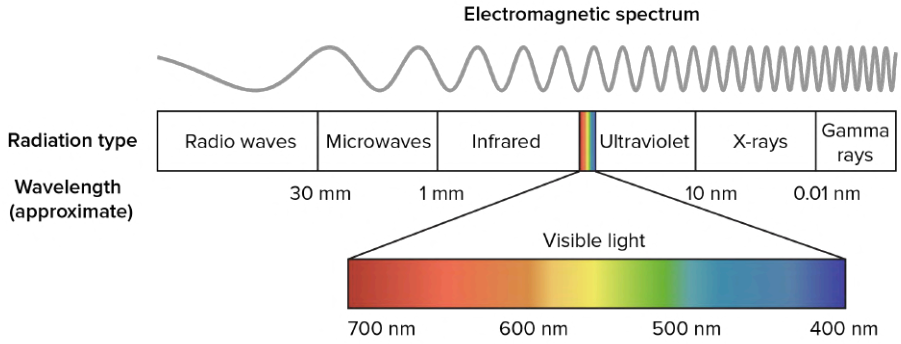


Figure 2.1: Spectral range diagram [2]

such as trash, stones, or dirt that are difficult to distinguish from conventional cameras. For example, traditional RGB cameras can detect damaged apples or those with red blemishes. However, multispectral cameras are capable of detecting even the tiniest ding in an apple. They can determine whether a portion of the apple is softer than others. In summary, multispectral cameras can be used when more information is needed than regular RGB cameras can provide. They have several advantages in a variety of ways.

2.2 Resolution

2.2.1 Spatial resolution

Spatial resolution is defined as the shortest distance between two adjacent features seen in a remotely sensed image. It is typically expressed in terms of the number of black and white ”line pairs” that contain resolution per unit length (line pairs/mm) for photographic images; for scanned images, it is typically expressed in terms of the size of Instantaneous Field of View (IFOV), or image element, which is the smallest area that can be resolved in a scanned image. Ground resolution is the actual size of the spatial resolution value on the ground. It is given in terms of the breadth of the line pair's coverage of the ground (meter) for photographic pictures; it is represented in terms of the actual size of the ground (meter) to which the image element corresponds for scanned images. For example, the spatial resolution or ground resolution of a Landsat multi-band scan is 79 meters (image element size 56*79 m^2). However, even when the line pair width and image element size are the same, the ground resolution is different. To express the same

information inside a row pair on a photographic image using machine-scanned optical images, approximately 2.8 image elements are required. For example, on a 1:100,000 image, a TM sensor on a Landsat with a ground resolution of 30m has an impact resolution of 0.3mm. As a result, the impact resolution changes according to the impact scale. Space resolution is a critical metric for evaluating the effectiveness of sensors and remote sensing data, as well as for determining the shape and size of objects.

The spatial resolution specifies distinct dimensions and dimensions independent of the picture, which indicates the spatial detail of the image. The higher the spatial resolution, the more capable it is of object recognition. However, the magnitude of spatial resolution simply reflects the degree of visibility of image features; the degree of distinguishability of each target on the image is proportional to the shape and size of the target, as well as its relative difference in brightness and structure from surrounding objects.

The different imaging results of the experimental sensor at a resolution of 2cm and 4cm at an altitude of 60m are shown in Figure 2.2.

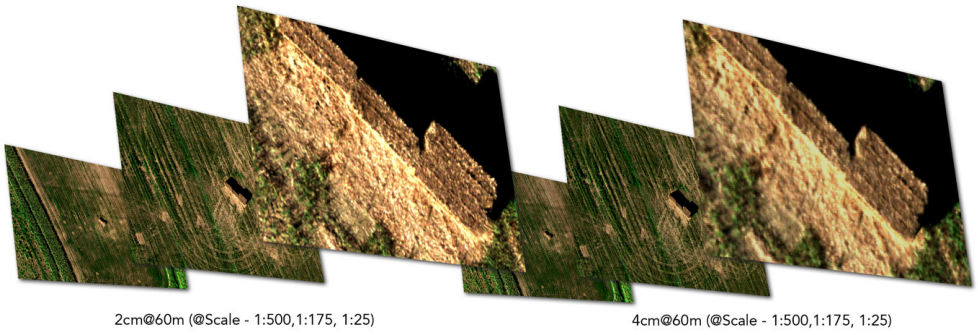


Figure 2.2: Same picture with different spatial resolution[3]

2.2.2 Spectral resolution

The spectral resolution is the smallest wavelength range that can be distinguished by the electromagnetic wave radiation received by the sensor detector device. The smaller the wavelength range of the band, the higher the spectral resolution. It also refers to the measurement of the wavelength range in which the sensor can be divided within its operating wavelength range. The more wavebands, the higher the spectral resolution. For example, Multispectral Scanner (MSS) and Thematic Mapper (TM), in the visible range,

the spectral range of all three bands of MSS is 0.1 microns; the spectral range of TM bands 1 to 3 is 0.07 microns, 0.08 microns, and 0.06 microns, respectively. The latter spectral resolution is higher than the former; MSS has 4 to 5 bands; TM has 7 bands, which also indicates that the latter spectral resolution is higher than the former. Since the difference in reflected or radiated electromagnetic wave energy of the ground feature spectrum is ultimately reflected in the gray scale difference of remote sensing images, spectral resolution also reflects the ability to distinguish different gray scale levels. For example, the multi-band scanner can distinguish 128 levels in the three visible bands, while the fourth band (wavelength range 0.3 microns) can distinguish only 64 levels, and the spectral resolution in the visible band is higher than that in the near-infrared band. Spectral resolution is one of the important indicators to evaluate the detection capability of remote sensing sensors and the ability of remote sensing information. Improving the resolution of the wave spectrum is beneficial for selecting the best combination of wavebands or bands to obtain effective remote sensing information and improve interpretation. However, for scanning sensors, the improvement of spectral resolution not only depends on the improvement of detector performance, but is also constrained by spatial resolution.

2.3 Index

2.3.1 Normalized difference vegetation index (NDVI)

Normalized Difference Vegetation Index (NDVI)[4] is a widely used remote sensing technique to identify vegetation and calculate its health and vitality. NDVI is positively correlated with measures such as leaf area index and expected leaf cover. In general, healthy and/or thick vegetation reflects a large amount of Near-Infrared (NIR) light but absorbs relatively little red light, while sparse or unhealthy vegetation exhibits reduced NIR reflectance. On the other hand, when vegetation is sparse or unhealthy, we note a decrease in NIR reflectance but an increase in red reflectance due to reduced absorption of red light by chlorophyll. NDVI is an indicator that combines data from the red and NIR bands to produce a single representative value. Its expression is as follows

$$\text{NDVI} = \frac{(Xnir - Xred)}{(Xnir + Xred)} \tag{2.1}$$

The negative sign in the numerator ensures that the numerator is always

smaller than the denominator value, regardless of our red and reddish values, meaning NDVI values are always between minus 1 and plus 1. Healthy vegetation has high NIR and low red reflectance values, with NIR values dominating. Since NIR values dominate the NDVI equation, NDVI tends to be positive. However, less healthy vegetation with red reflectance plays a greater role, lowering the overall NDVI value, but still positive. Since almost all NIR light is absorbed by water, the red reflectance value will be greater than the NIR light. Therefore, the molecule will become negative and the NDVI value will be negative or less than zero.

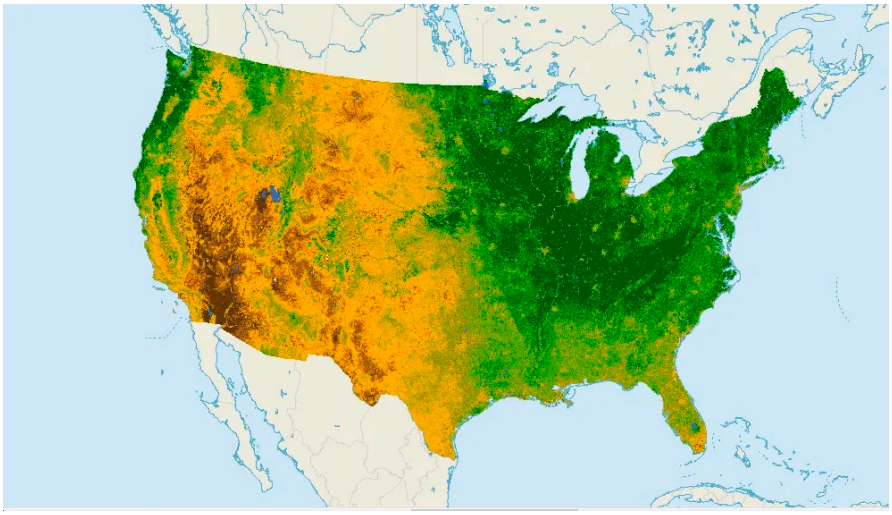


Figure 2.3: NDVI of United States. Acquired on 2021-10-29[5]

2.3.2 Normalized difference water index (NDWI)

Normalized Difference Water Index (NDWI) usually has two meanings. The first is used to detect changes in plant leaf water content and is calculated using both NIR and Short-wave Infrared (SWIR) wavelengths and was proposed by Gao in 1996 [6].

$$\text{NDWI} = \frac{(Xnir - Xswir)}{(Xnir + Xswir)} \tag{2.2}$$

Compared with NDVI, it can effectively extract the water content of the vegetation canopy; when the vegetation canopy is under water stress, the NDWI index can respond in a timely manner, which is important for drought monitoring.

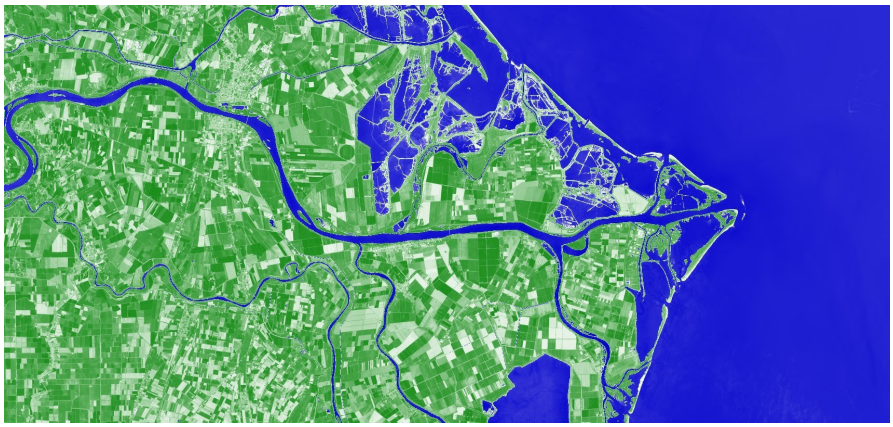


Figure 2.4: NDWI of Italy. Acquired on 2020-08-01 [7]

By contrast, Wilson used the Normalized Difference Moisture Index (NDMI) [8] in their study of forests in the U.S. state of Burmaine, and its expression is identical to Gao's. Because the meaning and use of Gao's NDWI and Wilson's NDMI are the same and different from Mcfeeters' NDWI used to study water bodies, the index used to study vegetation water content is generally renamed NDMI.

Another is used to distinguish water bodies from non-water bodies on remotely sensed images, using green and NIR wavelengths. It was proposed by McFeeters in 1996 [9] and is defined as:

$$\text{NDWI} = \frac{(Xgreen - Xnir)}{(Xgreen + Xnir)} \tag{2.3}$$

According to the analysis of the characteristic spectral data, there is a clear downward trend of the water body at Green-NIR, with positive values for increasing values and negative values for decreasing values, and the calculated results reflect the rate of change, which is also the slope. Between Green and NIR, the reflectivity of the water body has a clear decreasing trend, reaching a minimum value at NIR, so use can separate the water body from the non-water body.

Based on McFeeters' proposed NDWI analysis in 2005, Xv modified NDWI's wavelength combination and proposed a modified NDWI (Modified NDWI) [10], and conducted experiments on remote sensing images containing different water body types. NDWI index images are often mixed with urban building land information, which increases the range and area of extracted water bodies. It is also found that Modified Normalized Difference Water Index (MNDWI) outperforms NDWI in revealing microscopic characteristics

of water bodies, such as suspended sediment distribution and water quality changes. In addition, MNDWI can easily distinguish between shadows and water bodies, which solves the problem of eliminating shadows in water body extraction. The expression is:

$$\text{MNDWI} = \frac{(Xgreen - Xmir)}{(Xgreen + Xmir)} \tag{2.4}$$

2.4 Image classification

Image classification is an image processing method that distinguishes different categories of targets based on their respective characteristics reflected in the image information. It uses a computer to quantitatively analyze an image and assign each image element or region in the image to one of several categories instead of human visual interpretation.

2.4.1 Color feature-based classification

Color is a visual feature of the surface of an object, and each object has its own unique color characteristics. For example, when people talk about green color, it is often associated with trees or grassland, and when they talk about blue color, it is often associated with the sea or blue sky. The use of color features for image classification can be traced back to the color histogram method proposed by Swain and Ballard. Because of its simplicity and insensitivity to changes in image size and rotation, color histogram has received a lot of attention from researchers, and almost all content-based image database systems currently use color classification methods as an important means of classification, and many improvements have been proposed.

2.4.2 Texture-based image classification

The texture feature is also one of the important features of images, and its essence is to depict the neighborhood gray space distribution pattern of pixels.

In the early 1970s, Haralick et al. proposed a gray scale symbiotic matrix representation of texture features, which extracts the spatial gray scale correlation of textures, and first constructs a gray scale symbiotic matrix based on the distance and direction between pixels, and then extracts meaningful statistics from this matrix as texture feature vectors. Based on a psychological study of the visual perception of texture by human eyes, Tamuar et al. proposed

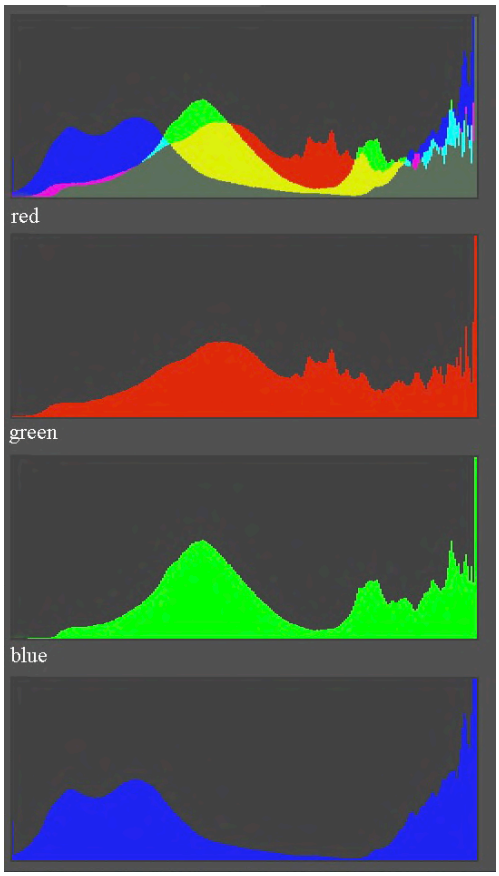


Figure 2.5: Histogram of Figure 3.1

six texture attributes that can simulate the visual model of texture, namely granularity, contrast, orientation, line shape, uniformity, and roughness.

2.4.3 Shape-based image classification

In two-dimensional image space, the shape is generally considered to be the area surrounded by a closed contour curve, so description of the shape involves description of the contour boundary and description of the area surrounded by this boundary. Most of the current shape-based classification methods revolve around building image indexes from contour features of shapes and regional features of shapes. The main descriptions of shape contour features are linear segment descriptions, spline fitting curves, Fourier descriptors, and Gaussian parametric curves, among others.

In fact, a more common approach is to use a combination of area and boundary features for similarity classification of shapes. For example, Eakins

et al. proposed a set of redrawing rules and simplified shape contours with line segments and arcs, and then defined two types of shape adjacency family and shape family subfamily functions to classify shapes. The adjacency subfamily mainly uses shape boundary information, while the shape family mainly uses shape region information. When matching shapes, in addition to the shape differences in each family, the differences in center of mass and perimeter in each family are also compared, as well as the differences in the position feature vector of the whole shape, and the discriminant distance query is the weighted sum of these differences.

2.4.4 Spatial relationship-based image classification

In image information systems, it is very important to distinguish different images in the image library based on the spatial positional relationships between objects in the image. Therefore, how to store image objects and their relationships to facilitate image classification is an important issue in the design of image database systems. Moreover, using the spatial relationship between objects in images to distinguish images is in line with people's habit of recognizing images, so many researchers have begun to study the classification method based on the spatial location relationship of objects from the spatial location relationship in images. As early as 1976, Tanimoto proposed the image element method to represent entities in images and proposed the use of image elements as an index of image objects [11]. Subsequently, it was adopted by the University of Pittsburghchang and proposed the representation of 2D-String for the classification of image spatial relations [12], which was adopted and improved by many people because of its simplicity and the possibility of reconstructing their symbolic maps from 2D-String for some images.

2.5 Classification algorithm

The following section will focus on supervised learning, which is the largest and best-developed classification of machine learning algorithms.

Linear Discriminant Analysis (LDA) [13] was invented by Fisher and dates back to 1936, when the concept of machine learning did not exist. It is a supervised data dimensionality reduction algorithm that projects a vector into a low-dimensional space by linear transformation, ensuring that the differences between samples of the same type after projection are small and that samples of different classes are as different as possible. The Bayesian classifier began

in the 1950s and is based on Bayesian decision theory, which assigns samples to the class with the highest posterior probability. Logistic regression [14] has a similar long history, dating back to 1958. It directly predicts the probability of a sample belonging to a positive sample and has been used for problems such as ad click rate prediction and disease diagnosis. The perceptron model [15], a linear classifier that can be seen as the predecessor of artificial neural networks, was born in 1958, but it was too simple to even solve the heterogeneous problem, so it did not have practical value and served more as an ideological enlightenment, laying the ideological foundation for the algorithms that followed. The K-Nearest Neighbors (KNN) algorithm [16] was born in 1967, an algorithm based on the idea of template matching, which is simple but effective and is still used today.

Before 1980, these machine learning algorithms were fragmented and unsystematic. However, their role in the development of machine learning as a whole cannot be ignored. It was from 1980 onwards that machine learning really became an independent direction. Since then, various machine learning algorithms have been proposed in large numbers and have been developed rapidly.

The three typical implementations of decision trees: ID3 [17], Classification and Regression Tree (CART) [18], and C4.5 [19] were important results from the 1980s to the early 1990s and are simple but interpretable, which makes decision trees still used in some problems today. A true backpropagation algorithm for training multilayer neural networks was born in 1986 [20], which is the training algorithm still used in deep learning today, laying the foundation for neural networks to move towards perfection and application. In 1989, LeCun designed the first truly convolutional neural network [21] for handwritten digit recognition, which is the progenitor of deep convolutional neural networks that are now widely used. Between 1986 and 1993, the theory of neural networks was greatly enriched and refined, but many factors at that time limited its large-scale use. The 1990s were the years of rapid development of machine learning. Two classical algorithms-Support Vector Machine (SVM) [22] and AdaBoost [23]-were born in 1995, and they dominated for a long time thereafter, while neural networks were not taken seriously. SVM represents the triumph of kernel technology, an idea that allows otherwise nonlinear problems to be handled well by implicitly mapping input vectors into a high-dimensional space. AdaBoost, on the other hand, represents the triumph of integrated learning algorithms, which can actually achieve amazing accuracy by integrating and using some simple weak classifiers. The now very popular Long Short-Term Memory (LSTM)

[24] appeared in 2000 and remained obscure for a long time until it was integrated with deep recurrent neural networks after 2013 and was successful in speech recognition. Random forests [25] appeared in 2001, in the same integrated learning as AdaBoost algorithm, which is simple but surprisingly effective in many problems, so it is still used on a large scale. A classic work of distance metric learning in 2009 [26] is considered one of the later appearances of classical machine learning algorithms, and in later years, this idea of obtaining distance functions by machine learning was widely studied and a number of papers appeared. From 1980 until the rise of deep learning in 2012, supervised learning developed rapidly, and various ideas and methods emerged. Moreover, no machine learning algorithm has achieved an overwhelming advantage over a large number of problems, which is very different from the current era of deep learning.

2.5.1 Random forest

Integration learning learns multiple estimators through training, and when a prediction is needed, the results of the multiple estimators are combined by a combiner as the final result output. The advantage of integration learning is that it improves the generality and robustness of a single estimator and has better prediction performance than a single estimator. Another feature of integrated learning is that it can be easily parallelized.

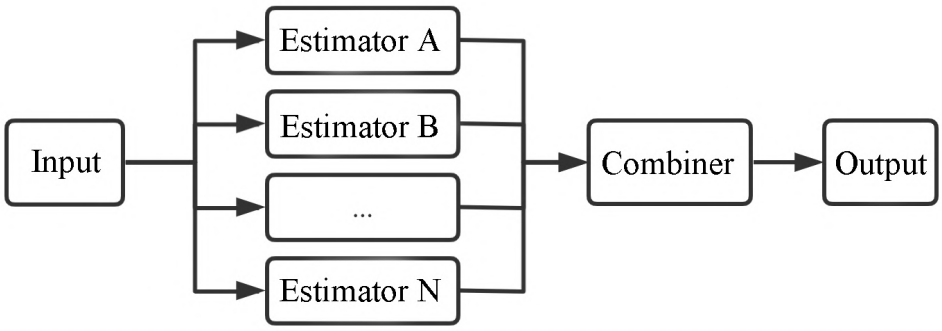


Figure 2.6: The basic process of integration learning

The Bagging algorithm is an integrated learning algorithm called Bootstrap Aggregating, which, as the name suggests, consists of two parts: Bootstrap and Aggregating.

The specific steps of the algorithm are: suppose there is a training dataset of size N, and each time there is a put-back selection of sub-datasets of size

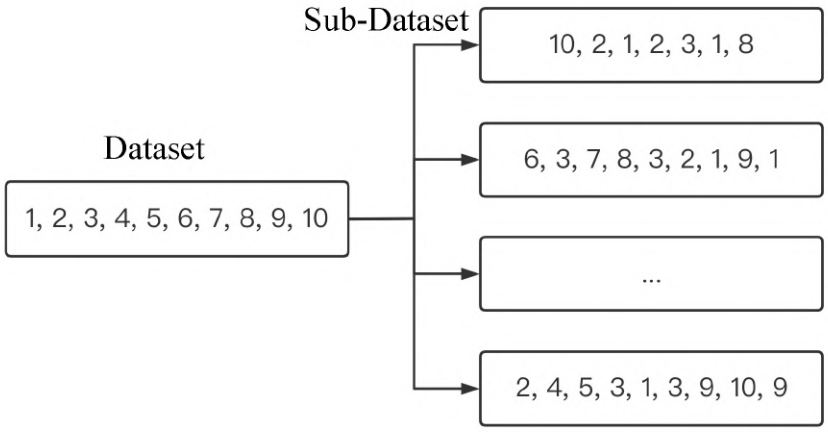


Figure 2.7: Example of Bagging algorithm using Bootstrapping to generate multiple sub-data

M from the dataset, a total of K selections, based on these K sub-datasets, are trained to learn K models. When it is time to make a prediction, these K models are used to make a prediction, and then the final prediction result is obtained by taking the average or majority classification.

Combining multiple decision trees together, each time the dataset is randomly selected with put-back, while some features are randomly selected as input, so the algorithm is called a random forest algorithm. It can be seen that the random forest algorithm is a Bagging algorithm with decision trees as estimators.

Figure 2.8 shows the specific flow of the Random Forest algorithm[25], where the combiner selects the majority of classification results as the final result in the classification problem, and takes the average of multiple regression results as the final result in the regression problem. Using the Bagging algorithm reduces overfitting, which leads to better performance. Individual decision trees are very sensitive to noise in the training set, but the above problem is effectively mitigated by the Bagging algorithm, which reduces the correlation between multiple decision trees trained.

Suppose the size of the training set T is N, the number of features is M, and the size of the random forest is K. The specific steps of the random forest algorithm are as follows: iterate through the size of the random forest K times. A new D sub-training set is formed by sampling N times from the T training set with back sampling. M features are randomly selected, where $m < M$. Using the new D and M training set features, a complete decision

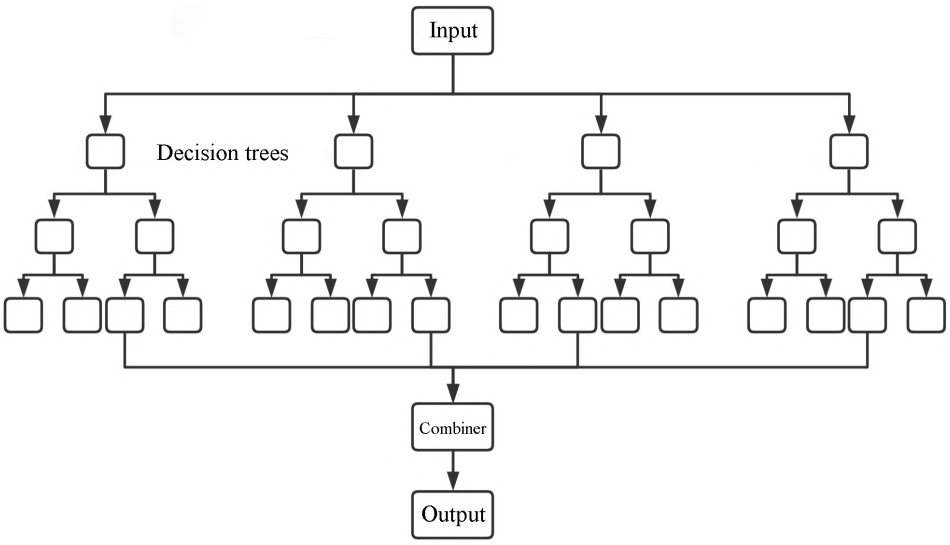


Figure 2.8: The specific process of the random forest algorithm

tree is learned, and then the random forest is obtained. The choice of M in the above algorithm: for the classification problem, $\sqrt{M}$ features can be used in each division, and for the regression problem, $\frac{M}{3}$ but not less than 5 features are selected.

2.5.2 Convolutional neural network

Convolutional Neural Networks (CNN) are a class of Feedforward Neural Network (FNN) with convolutional computation and deep structure, which is one of the representative algorithms of deep learning [27] [28]. Convolutional neural networks are capable of representation learning and shift-invariant classification of input information according to their hierarchical structure. It is also called Shift-Invariant Artificial Neural Network (SIANN). [29]

Convolutional neural networks differ from ordinary neural networks in that they contain a feature extractor consisting of a convolutional layer and a subsampling layer (pooling layer). In the convolutional layer of a convolutional neural network, a neuron is connected to only a portion of the neighboring neurons. The convolutional layer of a CNN typically contains several feature maps, each of which consists of a number of rectangularly arranged neurons, and neurons of the same feature map share weights, wherein the shared weights are the convolutional kernel. The convolutional kernel is usually initialized in the form of a random fractional matrix, and the

kernel will learn to obtain reasonable weights during the training process of the network. The direct benefit of shared weights (convolution kernels) is to reduce connectivity between network layers while reducing the risk of overfitting. Subsampling, also called pooling, is usually done in two forms: mean pooling and max pooling. Subsampling can be seen as a special kind of convolution process. Convolution and subsampling reduce model parameters and greatly simplify model complexity.

Related research that influenced the origin of CNN is related to the visual cortex, the most famous of which is the work done by Hubel and Wiesel in 1968 [30], who studied the brain wave responses of cats when viewing different pictures, and the conclusions they obtained were a great inspiration for later CNN.

The year before Hubel and Wiesel were awarded the Nobel Prize (1980) for their outstanding contributions, Japanese scientist Kunihiko Fukushima proposed the neocognitron [31], whose goal was to build a network structure that could achieve pattern recognition like the human brain and thus help us understand how the brain works. In this work, he creatively introduced many new ideas from the human visual system to artificial neural networks, which are considered by many to be the prototype of CNN. neocognitron is an improvement on his earlier work Congnitron [32], where the new network structure can satisfy translation invariance. Basically, most modern CNN structures are represented in this model. Three important ideas of convolutional operations: sparse interaction, parameter sharing, and isovariant representation also only parameter sharing is not considered. However, although this model does a very good job, its biggest limitation is that it uses unsupervised learning based on Winner Take All (WTA), so this model has been very lacking in practicality.

The next decade saw no major breakthroughs in CNNs until about a decade later, around 1989 to 1990, when LeCun applied backpropagation to a network like Neocoginitro to do supervised learning [21]. This paper introduces the important idea of weight sharing, and more importantly, it simplifies the convolution operation, makes it easy to apply backpropagation to CNNs, and uses it to solve a real-world problem.

In 1992, Juyang Weng published Cresceptron [33]. Although his model is not very impressive from a structural point of view, two processing methods in this paper are widely used today. The first is Data Augmentation, where we translate, rotate, scale, and other transform operations on the training inputs and then add them to the training set. On the one hand, it expands the training set, on the other hand, it improves the robustness of the algorithm

and reduces the risk of overfitting. The second technique is the proposed maximum pooling, which changed the situation of using average pooling for down-sampling.

In 1998, LeCun proposed LeNet-5 [34]. The structure is consistent with the CNN we often see today. Compared to previous work, the number of layers of the network is deepened to 7 layers. It contains two convolutional layers and two pooling layers.

The first breakthrough of the new century was not in algorithms but in engineering. In 2006, researchers successfully accelerated CNNs using GPUs, which were four times faster than CPU implementations. Although there is no algorithmic boost here, the implications are probably greater than the usual algorithmic boost.

In 2012, on the task of imageNet2012 image classification, the model proposed by Alex et al. [35] ranked first with an error rate of 15.3%, marking in some ways the resurgence of neural networks and the rise of deep learning.

Convolutional neural networks usually contain the following types of layers.

The convolutional layer is the core building block of convolutional neural networks. In image recognition, we refer to convolution as two-dimensional convolution, where a discrete two-dimensional filter (also called a convolution kernel) performs a convolution operation with a two-dimensional image, which simply means that the two-dimensional filter slides to all positions on the two-dimensional image and makes an inner product with that pixel point and its domain pixel points at each position. Convolution operations are widely used in image processing, and different convolution kernels can extract different features, such as edge, linear, and corner features. In deep convolutional neural networks, low-level to complex features of images can be extracted by convolutional operations.

The rectified linear layer is used for nonlinear mapping of the convolutional layer output. The excitation function used in CNN is generally Rectified Linear Unit (ReLU), which is characterized by fast convergence and simple gradient finding but is more fragile.

The pooling layer is usually added after the convolutional layer to reduce the feature map. The main purpose is to reduce computational effort by reducing network parameters and controlling overfitting to some extent. The pooling operation is independent for each depth slice, and the gradient is usually 2*2. Compared to the convolutional operation performed by the convolutional layer, the pooling layer generally performs the following operations.

- Max Pooling: Takes a maximum of 4 points. This is the most common pooling method.
- Mean Pooling: Takes the mean value of 4 points.
- Gaussian Pooling: Borrowed from Gaussian Blur. Not commonly used.
- Trainable pooling: Trains a function that accepts 4 points as input and outputs 1 point. Not commonly used.

The most common pooling layer is size 2*2, with a step size of 2, and downsamples each depth slice of the input. The pooling operation will keep the depth size constant. If the input cell size of the pooling layer is not an integer multiple of two, it is usually taken to zero padding to a multiple of two and then pooled.

The fully connected layer, which combines all local features into global features, is used to calculate the final score for each category. Fully connected layers and convolutional layers can be converted to each other: for any convolutional layer, all that is needed to turn it into a fully connected layer is to turn the weights into a huge matrix, where most of them are 0 except for some specific blocks (because of local perception), and many blocks have the same weights (because of weight sharing). Conversely, for any fully connected layer, it can also be turned into a convolutional layer.

Ross Girshick et al. proposed the Region Based-Convolutional Neural Network (RCNN) algorithm in 2014 [36]. R-CNN models first select several proposed regions from the images and then label their classes and bounding boxes (e.g., offsets). Then, they perform forward computation using CNNs to extract features from each proposed region. The characteristics of each region are used to predict their categories and bounding boxes.

Specifically, the R-CNN consists of four main components.

- A selective search of the input image is performed to select multiple high-quality proposed regions [37]. These proposed regions are typically selected at multiple scales and have different shapes and sizes. The category and ground-truth bounding boxes of each region are labeled.
- A pre-trained CNN in truncated form is placed before the output layer. It converts each proposed region into the required input dimension of the network and outputs the features extracted from the proposed regions using forward computation.

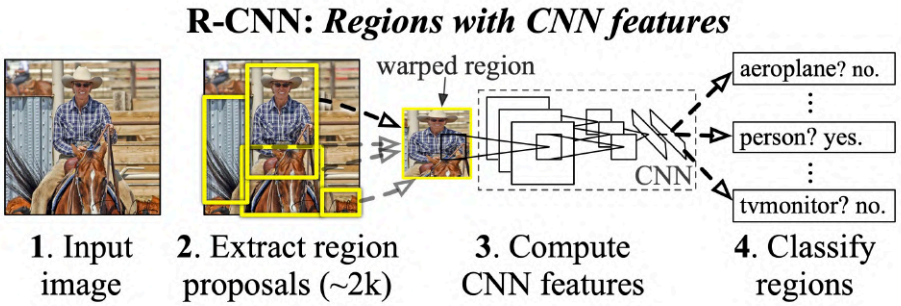


Figure 2.9: Object detection system overview using R-CNN[36]

- Multiple support vector machines are trained for target classification using the features and labeled categories of each region as samples. Here each support vector machine is used to determine whether a sample belongs to a certain category or not.

- A linear regression model is trained for ground-truth bounding box prediction by combining the features of each region and the labeled bounding boxes as an example.

Although the R-CNN model uses pre-trained CNNs to efficiently extract image features, the main drawback is its slow speed. As you can imagine, we can select thousands of proposed regions from an image and require thousands of forward computations by CNN to perform target detection. This huge computational load means that R-CNNs are not widely used in practical applications.

The main performance bottleneck of the R-CNN model is the need to extract features for each proposed region independently. Because these regions have a high degree of overlap, independent feature extraction leads to a large number of repeated computations. Fast R-CNN[38] improves R-CNN by performing CNN forward computation only on the image as a whole. Figure 2.10 illustrates a fast R-CNN model.

Its main computational steps are as follows.

- In contrast to the R-CNN model, the Fast R-CNN model uses the entire image as input to the CNN for feature extraction, rather than each proposed region. In addition, the network is usually trained to update model parameters. Since the input is a complete image, the shape of the CNN output is $1 * c * h_1 * w_1$

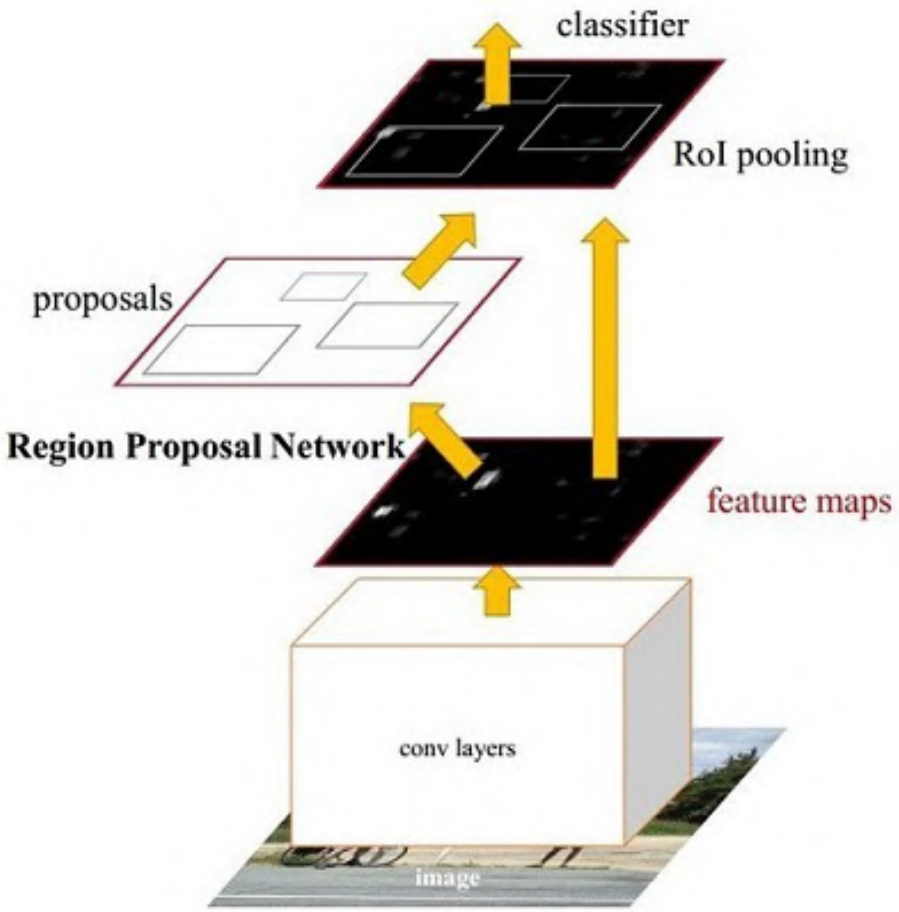


Figure 2.10: Fast R-CNN architecture[38]

- Suppose selective search generates n proposed regions, and their different shapes represent the different shapes of the CNN output regions of interest (RoIs). Features of the same shape must be extracted from these RoIs (here we assume a height of h_2 and a width of w_2). Fast R-CNN introduces RoI pooling, which uses CNN output and RoIs as input, and the output is a concatenation of features extracted from each proposed region (shape of $n * c * h_2 * w_2$).
- A fully-connected layer is used to convert the output into the shape of $n * d$, where d is determined by the model design.
- During category prediction, the shape output of the fully connected layer is converted again to $n*q$ and softmax regression is used (q is the number of categories). During the prediction of the bounding box, the output

shape of the fully connected layer is again converted to $n * 4$. This means that we predict species and bounding boxes for each proposed region.

The RoI pooling layer in Fast R-CNN is a bit different from the pooling layers we discussed earlier. In a normal pooling layer, we set the pooling window, padding, and step size to control the output shape. In a RoI pooling layer, we can directly specify the output shape of each region, such as specifying the height and width of each region's output as h_2, w_2. Suppose the height and width of the RoI window are h and w, and this window is divided into a grid of sub-windows with the shape $h_2 * w_2$. The size of each sub-window is approximate $(h/h_2) * (w/w_2)$ The height and width of the sub-window must always be integers, and the largest element is used as the output of the given sub-window. This allows the RoI pooling layer to extract features of the same shape from RoIs of different shapes.

After the accumulation of R-CNN and fast region-based CNN, Ross B. Girshick proposed the new faster region-based CNN in 2016 [39], which structurally integrates feature extraction, proposal extraction, bounding box regression (rectification), and classification into a network, which improves overall performance, especially in detection speed.

Faster region-based CNN can actually be divided into 4 main components

- Convolutional Layer: as a CNN network target detection method, faster region-based CNN first uses a set of basic convolutional+ReLU+pooling layers to extract feature maps of images. the feature maps are shared for subsequent RPN and fully connected layers.

- Region Proposal Networks: The RPN network is used to generate region proposals. This layer judges whether the anchors belong positive or negative through softmax, and then uses bounding box regression to correct the anchors to obtain accurate proposals.

- RoI Pooling: This layer collects the input feature maps and proposals, extracts the proposal feature maps after combining this information, and sends them to the subsequent fully connected layer to determine the target category.

- Classification: Using the proposal feature maps to calculate the category of proposals, and again bounding box regression to obtain the final exact position of the detection box.

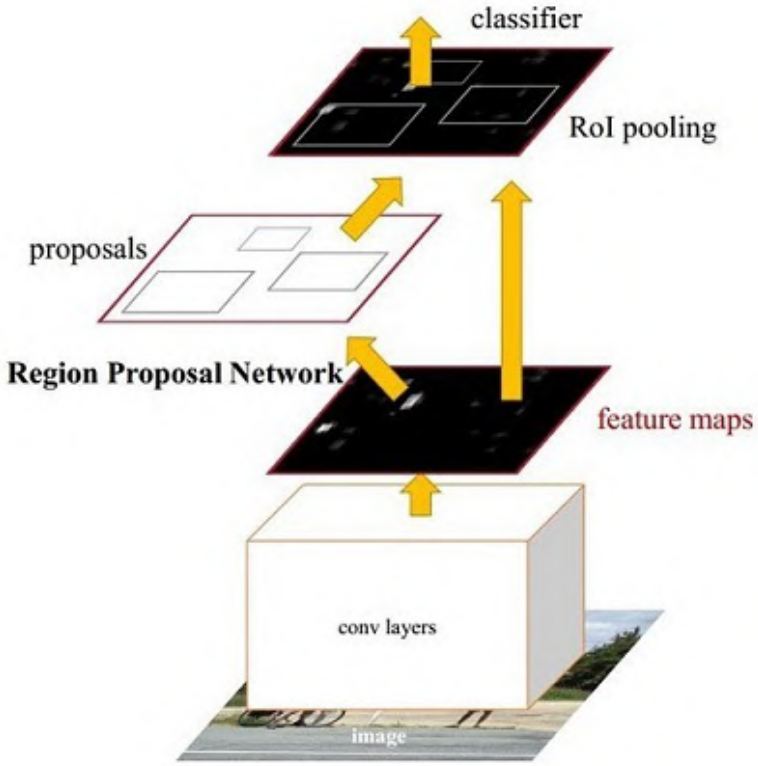


Figure 2.11: Basic structure of Faster R-CNN.[39]

Chapter 3

Methods

3.1 Research process

For image classification, CNN will be the first choice for this project. This project will also use the random forest algorithm for comparison. In Gabriel et al.'s study, NDVI was used as an input for high-accuracy detection of malaria vector larval habitats using drone-based multispectral imagery. And their studies have yielded good results. If experimental conditions permit, this project will also be further explored using NDVI or NDWI.

Table 3.1 shows RGB and multispectral bands used in the classification.

	Spectral Band	label	Wavelength center	Bandwidth
RGB Bands	Blue	blue	475nm	32nm
	Green	green	560nm	27nm
	Red	red	668nm	14nm
Multispectral Bands	Red Edge	edge red	717nm	12nm
	Near Infrared	nir	842nm	57nm
Other	NDWI	ndwi	0	2

Table 3.1: RGB and multispectral bands used in the classification.

3.2 Data collection

3.2.1 Sampling

The drones equipped with the above sensors will collect data in Sri Lanka, take multispectral photos of farmland and forests in parts of the region, and conduct experiments. Figure 3.1 shows where the experimental pictures were taken.

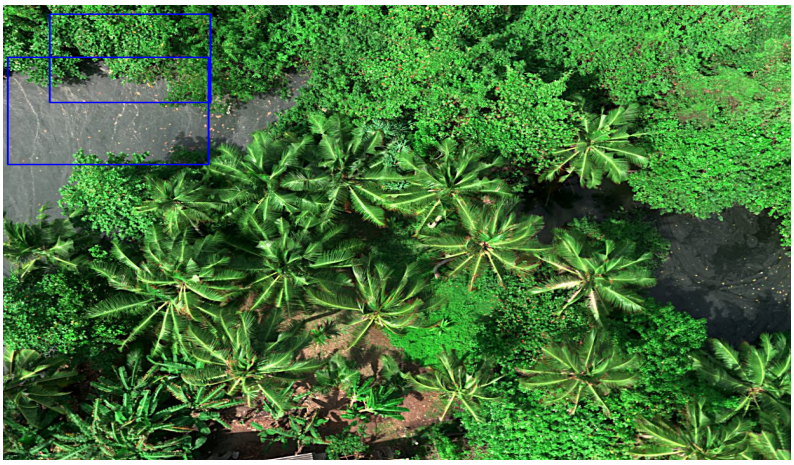
Figure 3.1: Environment picture in Sri Lanka

3.2.2 Sample size

The total number of images planned for training is approximately 300. Each shot will produce two sets of images with the suffixes .tif and .jpg. The tif file contains all the information of the five spectral bands in the multispectral image and is used for subsequent machine learning. The jpg file, on the other hand, contains RGB information visible to the human eye for better visualization to perform label classification and result evaluation. The image size is 1240 * 920 pixels.

During the data collection, 112 plots were obtained due to uncontrollable factors (energy shortages and power shortages due to the war). Experiments with the available data showed that the results were not satisfactory, as shown in 3.2.

One of the reasons for the above results is that the training set contains too few images to allow the model to obtain effective recognition. But another



(a) Result 1

(b) Result 2

Figure 3.2: Result of initial image

reason is that the single image is too large and contains too many redundant factors such as trees, houses, roads, etc. To solve the above problems, the image segmentation method is used. The original images were divided into 8 equal parts in both horizontal and vertical directions, making the original set of images 64 times larger, with a total of 7168 images. The bounding boxes corresponding to the original images are also segmented so that the newly generated labels can correspond to the sub-images one by one according to the coordinate positional relationship, and the size of each sub-image is 155*115.

For CNN and Random Forest, the image is divided into smaller sub-images to better capture the water body. The 10 original size images are divided into 40 in both horizontal and vertical directions, and the size is changed from

1240*920 to 31*23, with a total of 16,000 images.

3.2.3 Social and ethical concerns

The altitude and range of drone flights are in compliance with local laws and regulations. The areas where the flights and pictures are taken are public areas, and the pictures do not contain any information about people and do not violate the privacy of others.

3.3 Experimental design

3.3.1 Test environment

The code work for this project is written in python, version 3.7 is required. the machine learning part is done by pytorch, version pytorch 1.5, torchvision 0.6 is required.

3.3.2 Hardware to be used

The REDEDGE-P shown in 3.3 from MicaSense[3] was used for this project. This sensor features specialized optics and an industry-leading industrial imaging sensor and a narrow-band scientific grade filter. In addition, it underwent a rigorous factory calibration process, creating a high-quality, calibrated, rugged tool for high-quality output. 3.4 shows the main multispectral bands of REDEDGE-P.

Figure 3.3: RedEdge-P high-resolution multispectral and RGB sensor

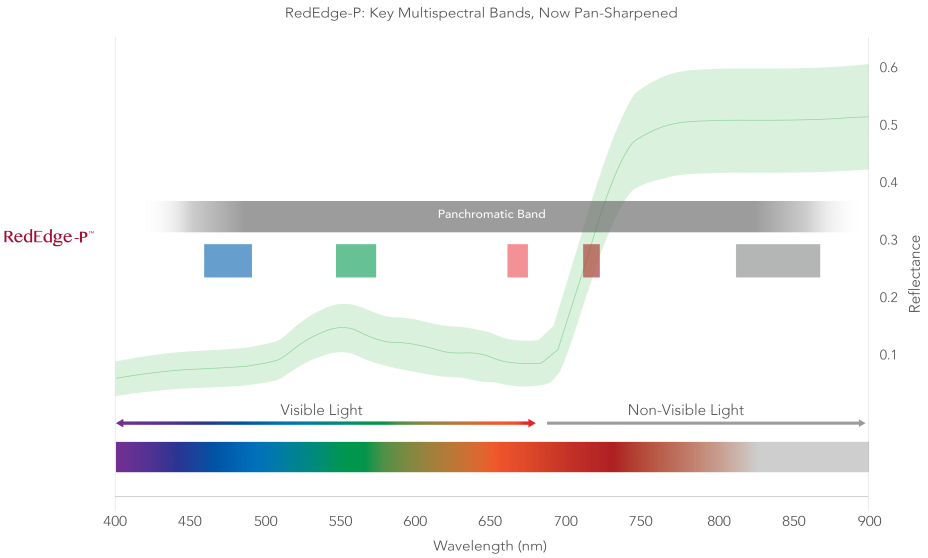


Figure 3.4: Main multispectral bands of REDEDGE-P.[3]

The drone used in this research is DJI Phantom 4[40], it is an extremely smart flying camera able to intelligently track objects without a separate device, avoid obstacles and fly with the tap of your finger. All while shooting 4K video or 12-megapixel stills.

Figure 3.5: DJI Phantom 4.[40]

The REDEDGE-P-equipped DJI Phantom 4 drone used for data collection is shown in Figure 3.6.

Figure 3.6: DJI Phantom 4 with REDEDGE-P

For the software part, programming was done on Google Colab, and the software was upgraded to Google Colab Pro because some Python packages required terminal calls to be completed.

3.4 Evaluation framework

For the evaluation of classification results, this project uses Intersection over Union (IOU)[41], which is a term used to describe the degree of overlap between two framed areas on an image, and is calculated as

$$IOU = \frac{\text{Area of Intersection of two boxes}}{\text{Area of Union of two boxes}} \tag{3.1}$$

We can see that the larger the overlap area of the two parts, the higher the IOU is, and when the two parts completely overlap, the intersection is equal to concatenation, and the IOU is equal to 1. Similarly, the smaller the overlap area of the two parts, the lower the IOU, and when the two parts do not completely overlap, the intersection is equal to 0, and the IOU is also equal to 0. Therefore, the IOU value range is 0 to 1.

Sometimes in the actual experiment, the shape of the water bodies is irregular and scattered in different locations of the image, so the location and

number of pixels identified as water bodies need to be counted. Because there is no sufficiently precise method to identify water bodies, the correct location of water bodies on the image needs to be manually marked for IOU calculation. IOU is calculated as

$$IOU = \frac{\text{Pixels of Intersection of two areas}}{\text{Pixels of Union of two areas}} \tag{3.2}$$

However, IOU does not fully evaluate the classification results. For example, it may turn out that the result has a high IOU value, but the non-intersecting part is outside the real body of water in the image. This can lead to meaningless parts in subsequent water depth calculations. In addition to calculating overlapping pixel points, the edge of the water body, which is the boundary line between the water body and the land, is equally important. So we also need to consider the shape and size of the water body.

Chapter 4

Experiment

4.1 Faster region-based CNN

The faster region-based CNN network with the same structure as in [39] is used in this experiment. The structure of the network is shown in Figure 4.1.

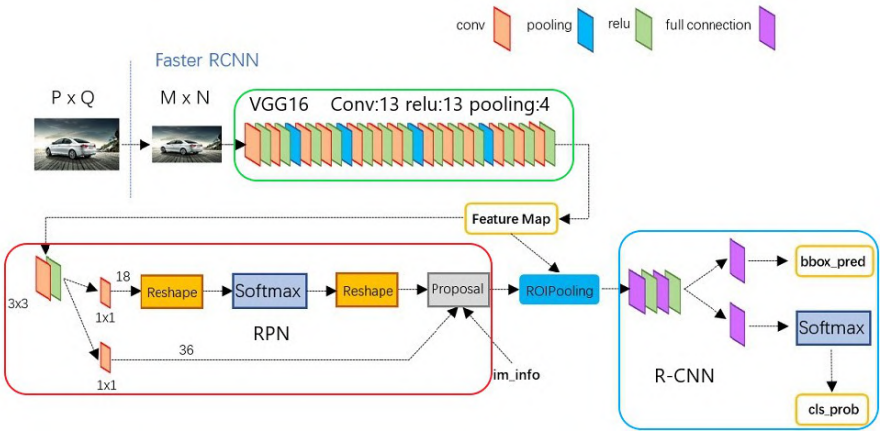


Figure 4.1: Architecture of faster region-based CNN[42]

Convolutional layers contain three types of layers: Convolutional, Pooling, and ReLU layers. In the VGG16 network used in this experiment, there are 13 Convolutional layers, 13 ReLU layers, and 4 Pooling layers in the Convolutional Layers section. In Convolutional layers, all Convolutional layers are: kernel_size=3, pad=1, stride=1, and all Pooling layers are: kernel_size=2, pad=0, stride=2. A matrix of size $M * N$ is fixed to $(M/16) * (N/16)$ by convolutional layers. In this way, the feature maps generated by Convolutional layers can all correspond to the original map.

Faster region-based CNN discards traditional sliding window and selective search methods and uses RPN to generate detection boxes directly, which is a great advantage of Faster R-CNN and can greatly improve the detection box generation speed. The RPN network is actually divided into two parts, one part is used to classify anchors by softmax to obtain positive and negative classifications, and the other part is used to calculate the bounding box regression offset for anchors to obtain the exact proposal. In fact, the entire network has completed the target localization function when it reaches the Proposal Layer.

The RoI Pooling layer is responsible for collecting proposals and calculating proposal feature maps that are fed to the subsequent network. The RoI pooling layer has two inputs: original feature maps and proposal boxes of different sizes output by RPN. But the fixed length output is achieved by RoI Pooling.

The classification part uses the already obtained proposal feature maps to calculate which category each proposal belongs to by fully connected layer and softmax, and outputs the cls_prob probability vector; at the same time, the offset position bbox_pred of each proposal is obtained again by using bounding box regression, which is used to regress to a more accurate target detection box.

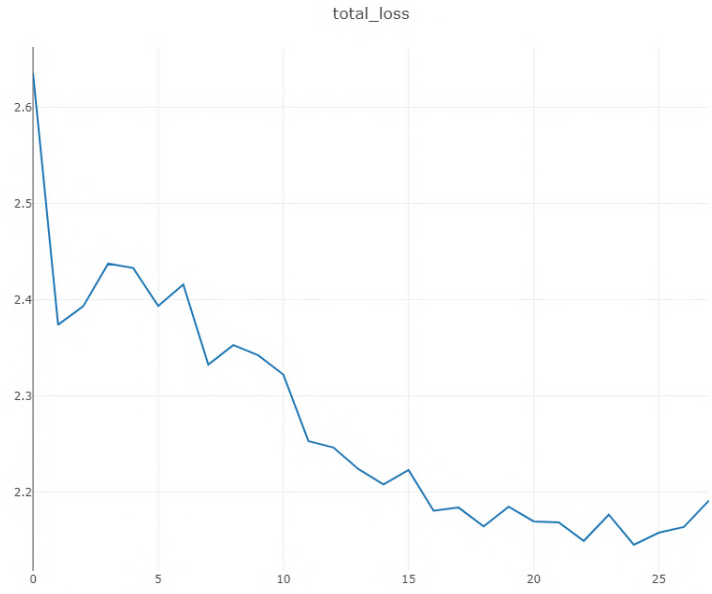
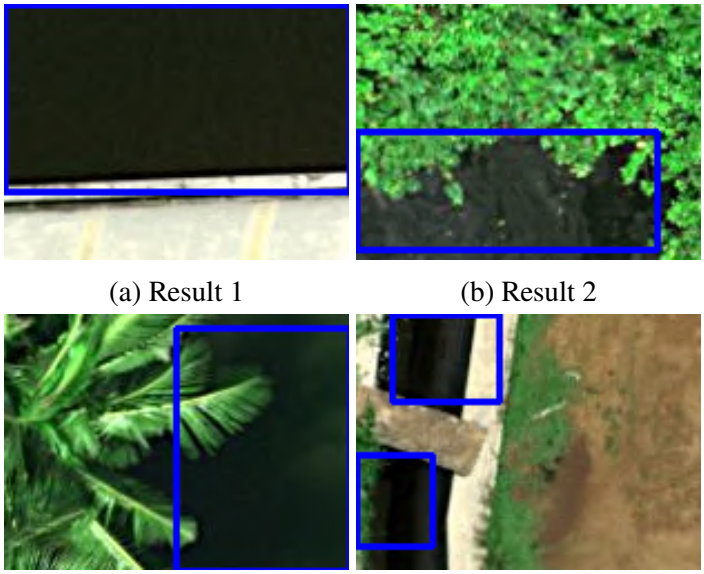


Figure 4.2: Total loss of epoch 27 of initial image

At the beginning of the experiment, multispectral raw images directly collected by drones were used. The results are shown in Figure 3.2. The number of epochs is set to 27. The loss is shown in Figure 4.2

Due to the large size of the picture and too many influencing factors, the original picture was divided into 64 parts and the experiment was carried out again. The results are shown in Figure 4.3. The loss is shown in Figure 4.4



(a) Result 1 (b) Result 2

(c) Result 3 (d) Result 4

Figure 4.3: Result of sub-image

The image of the loss may reflect that when the number of epochs reaches 27, the total loss is between 0.4 and 0.5, which is a very high value, and there is still a clear downward trend. Therefore, the number of Epochs is set at 200 to conduct experiments again. And the loss is shown in Figure 4.5

From the above image, it can be seen that when the number of epochs exceeds 100, the loss value begins to stabilize, and there is no longer a significant decline. At the same time, the loss value is stable at less than 0.1, which is acceptable. The final experimental results are also derived from this model.

In the final experiment, 80% of the 7168 images were randomly selected and used as the training set; the remaining 20%, or 1433 images, were used as the test set. The next experiment was divided into two parts, and 316 of the 1433 images contained water bodies. According to the definition of IoU, we need the value of ground truth to be the denominator of IoU calculation.

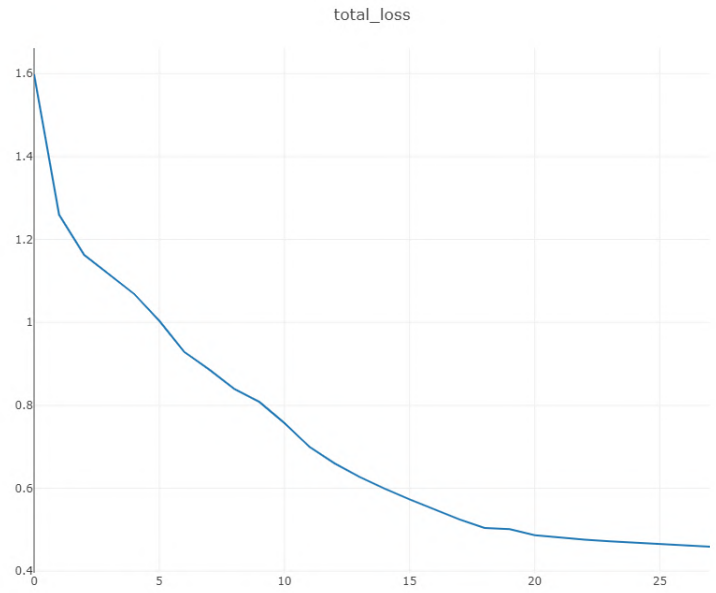


Figure 4.4: Total loss of epoch 27 of sub-image

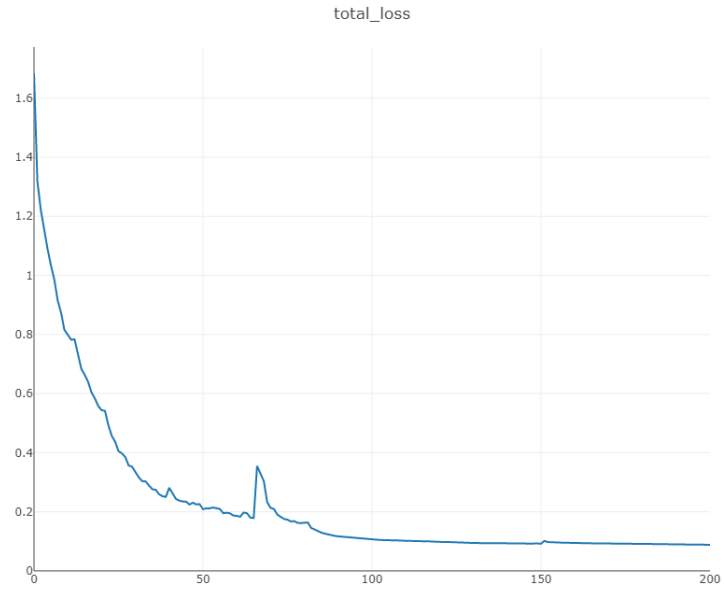


Figure 4.5: Total loss of epoch 200 of sub-image

Therefore, 316 images containing water bodies were selected as the test set for the experiments.

To compare with the results of CNN and Random Forest, all 1433 images of the test set were tested using a faster region-based CNN, in which 316 contain water. Images that do not contain water are also taken into account, i.e. if no water is detected in these images, the detection is successful, and vice versa.

4.2 CNN

In this project, a basic CNN network is implemented, using three convolutional layers and three pooling layers to extract image features. The convolutional layers are followed by batch normalization to normalize the data, which allows the data not to be oversized and cause instability in the network performance before ReLU, ensuring that the inputs to each layer of the network have the same distribution as possible. The ReLU function is used as the activation function after convolution. As a common regularization method, adding the dropout layer can attenuate the overfitting effect of deep neural networks. This method randomly inactivates a certain percentage of neural units in each training according to the set probability parameter. The dropout value in this experiment is 0.3. Finally, the sigmoid function is used to implement the judgment of water bodies. Experiments use cross-entropy as a loss function to measure the degree of model merit. And it is calculated as:

$$\text{loss} = -\frac{1}{m}\sum_{\text{i=1}}^{\text{m}}\sum_{\text{j=1}}^{\text{n}} \text{p}\left(\text{x}_{\text{ij}}\right)\log\left(\text{q}\left(\text{x}_{\text{ij}}\right)\right) \tag{4.1}$$

In the above equation p is the true probability distribution and q is the predicted probability distribution. Cross entropy measures the degree of difference between two different probability distributions in the same random variable, which is expressed in machine learning as the difference between the true probability distribution and the predicted probability distribution. The smaller the cross-entropy value, the better the model prediction. Cross entropy is often standard in classification problems with softmax, which processes the output so that the predicted values of its multiple classifications sum to one, and then calculates the loss by cross-entropy.

The size of the dataset used for CNN experiments was 16,000, of which 80%, or 12,800 images, were randomly selected as the training set and the remaining 20%, 3200 images, were used as the test set. Images containing

water and images without water were randomly distributed in the dataset.

4.3 Random forest

In this project, Scikit-learn, a free machine learning library for Python, is used to implement random forest classification, where random_state needs to have a fixed value. For processes that are essentially random, it is necessary to control the random state so that the same results are shown repeatedly. If there is no control over the random state, the results of the experiment cannot be fixed, but are revealed randomly. The random_state value in the experiment is set to 0. n_estimators is the number of base evaluators. The larger the value, the better the model tends to be. However, as n_estimators increase, a larger amount of computation is required, and also when it increases to a certain value, a decision boundary is reached. The variation in the effect of the model with n_estimator in this experiment will be discussed in Section 5.3.

The experiment of random forest used the same dataset as CNN.

Chapter 5

Results

5.1 Faster region-based CNN

Of the 7168 images after segmentation, 20% were randomly selected, i.e. 1433 images, as the test set. Of these, 316 contain water bodies. That is, there are corresponding bounding boxes that can be used to verify the test results.

Table 5.1: Results of faster region-based CNN

IoU Value (>=)	Number	Percentage
0	309	0.9778
0.01	300	0.9494
0.05	286	0.9051
0.1	277	0.8765
0.2	254	0.8038
0.3	244	0.7722
0.4	229	0.7247
0.5	209	0.6614
0.6	172	0.5443
0.7	134	0.4241
0.8	99	0.3133
0.9	62	0.1962

The purpose of this paper is to detect the water body in the picture, so for the evaluation of the result, as long as there is an intersection between the bounding box and the ground truth generated by the recognition, that is, the IoU is greater than 0, it meets our expectations for the result. The results of the test set are presented in Table 5.1. Of a total of 316 test images, 309 detected

water bodies, representing 97.78%. From this point of view, the model used in the experiment has achieved very good results.

The size of the IoU value is also a very important reference factor. The larger the IoU value, the greater the intersection of the output single bounding box and ground truth, which means that the selection of the water body range in the image is more accurate. Therefore, also in Table 1, the experimental results also include the proportion of water recognition with different IoU values.

Figure 5.1 plots the data in Table 5.1 as a line chart. The vertical axis is the number of images greater than the current IoU value. As can be seen from Figure 5.1, as the IoU value increases, the number of images that meet the predicted results decreases approximately linearly. There is no comparable benchmark result for this change. But this should be in line with the law of change predicted by the model. It is also worth noting that when the IoU value is set to 0.9, 62 (19.62%) images still meet the requirements, which is a good result.

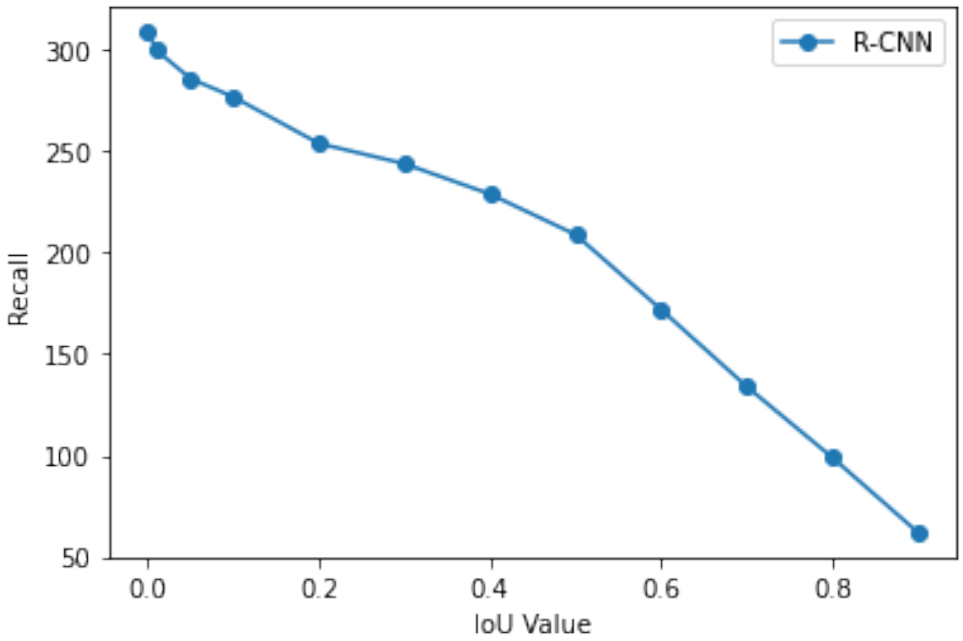


Figure 5.1: Results of faster region-based CNN

There are two other situations that need to be explained. One is when the ground truth is much larger than the output bounding box and the output bounding box is included in the ground truth. Figure 5.2 illustrates this situation. In this case, the denominator for calculating IoU's value is the union

of the two boxes, which is the size of the ground truth. The numerator used to calculate the IoU value is the intersection of the two boxes, i.e. the size of the predicted bounding box. At this time, the model output has accurately positioned the water body, but it may not be obvious by measuring the IoU value.

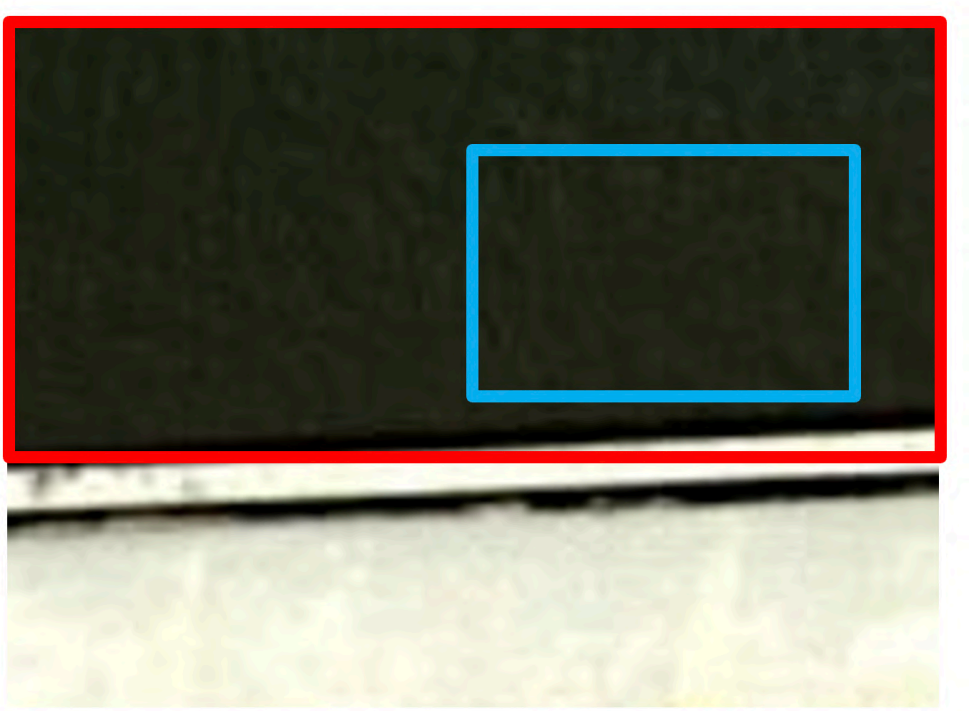


Figure 5.2: Bias of IoU 1

The second case is shown in Figure 5.3. In this case, a continuous water body may be divided into many small bounding boxes because one or a certain pixel is judged not to be water. Because IoU calculation uses an output bounding box, the value of IoU will become smaller.

Through the analysis of the above two situations, it can also be seen that it is unreasonable to consider only the size of the IoU value for the evaluation of results. Therefore, this project uses IoU greater than 0 as the standard for evaluating results.

Because the CNN and RF datasets randomly distributed images with and without water, all 1433 randomly selected images were used as the test set to compare CNN and random forest results. This randomly divided 316 water-containing and 1117 water-free images. In this case, 1409 of the 1433 test

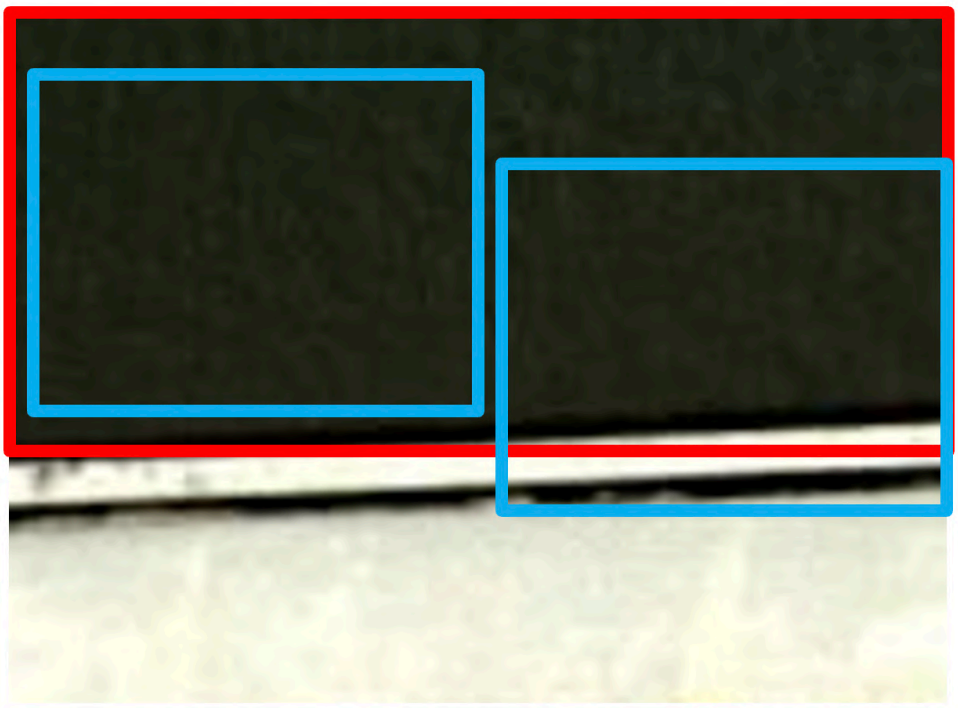


Figure 5.3: Bias of IoU 2

images achieved a correct prediction with a recall rate of 98.33%.

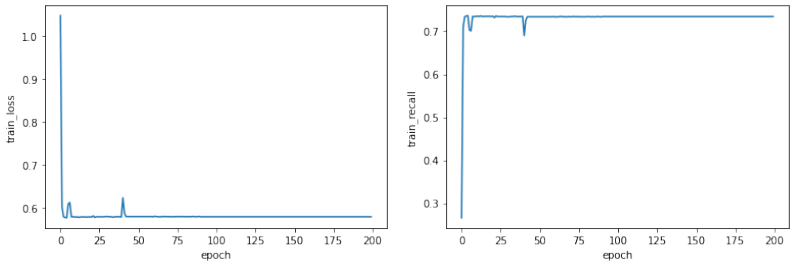
5.2 CNN

First, the same dataset used in the faster region-based CNN network was used for training and testing on the CNN network, and the results are shown in Figure 5.4.

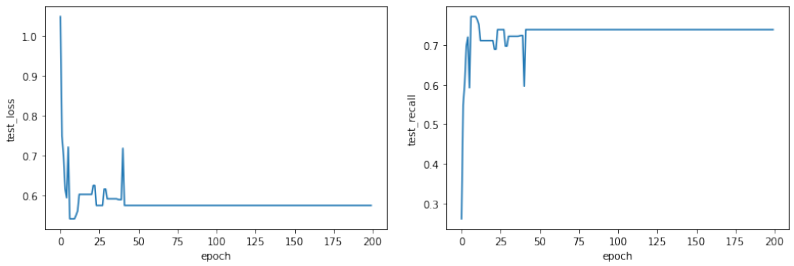
This experiment still uses 200 epochs. It can be seen from the results that when the epoch reaches 50, the recall rate of the test set no longer fluctuates and stabilizes at 0.7388.

Whereas CNN recognizes and classifies the whole picture, some of the pictures in the above dataset still have other information (houses, trees, roads, etc.), and the picture is divided into 31*23. The aim is to ensure accurate identification and determination of water bodies. The loss is shown in Figure 5.5.

Compared with previous experiments, the results of this experiment are better. The recall rate reached 95.8% after 100 epochs.



(a) Train Loss (b) Train Recall



(c) Test Loss (d) Test Recall

Figure 5.4: CNN Result of sub-image

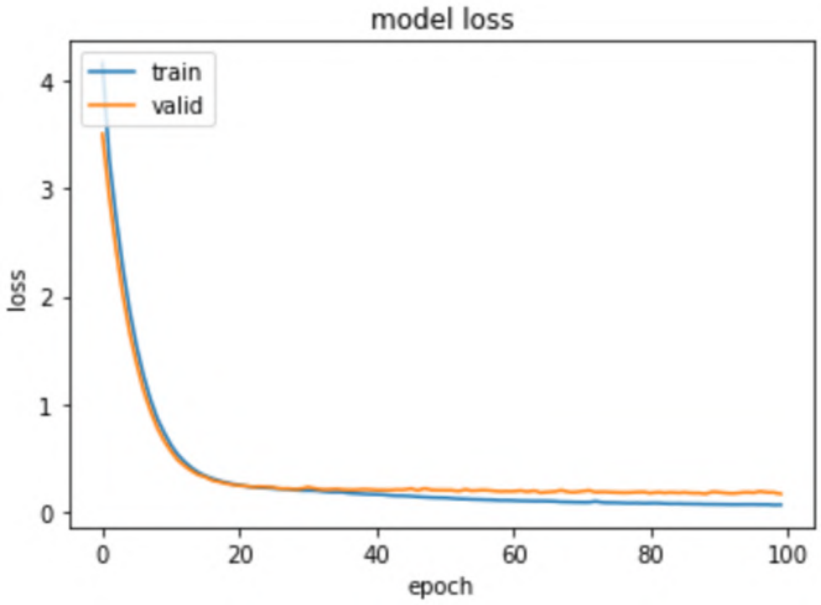
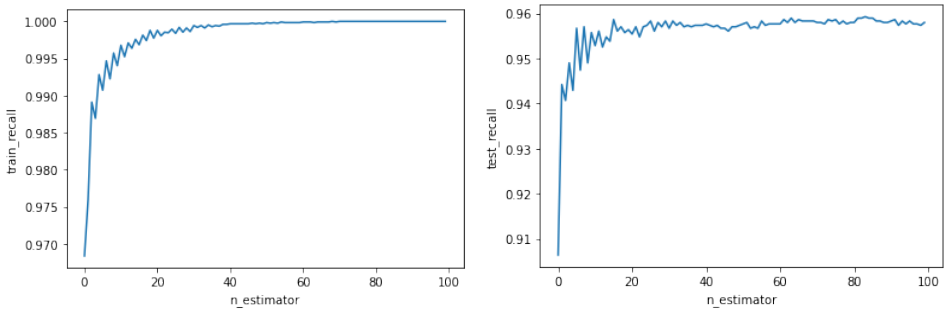


Figure 5.5: CNN loss of tiles

5.3 Random forest

The result of the random forest is shown below. Figure 5.6a shows how the recall rate of the training set changes as n_estimator grows. When the n_estimator reaches 75, the recall rate reaches 100% and remains stable until the n_estimator equals 100.



(a) RF result of training set

(b) RF result of test set

Figure 5.6: Random forest result

When n_estimator is equal to 100, the recall rate of the test set is 95.8%. Figure 5.6b shows how the recall rate of the test set changes as n_estimator grows. When n_estimator exceeds 30, the recall rate fluctuates in a small range between 95.6% and 96%. Although the recall rate does not stabilize at a fixed value, it has reached a high accuracy rate, and increasing the number of n_estimators increases the complexity of the model significantly.

Chapter 6

Conclusions and future work

6.1 Conclusions

In this master's thesis, we investigate water body recognition from high-resolution multispectral images obtained by UAVs. According to the goals set in 1.3, this thesis mainly achieves the following goals.

First, a faster region-based CNN model for high-resolution multispectral images obtained by UAVs is constructed on this paper. The structure of the model and the size of the input images are adjusted appropriately for the characteristics of the model and the desired objectives. The final decision was made to use 200 epochs and an input image size of 155*115.

Next, a CNN network with three convolutional layers and a random forest network is constructed in this thesis for comparison with the faster region-based CNN. The Sigmoid function is selected for classification of the CNN network, and the epochs are equal to 100. The number of decision trees in the random forest network is 100.

Finally, the faster region-based CNN model achieves the best results with a success rate of 98.33% recognition of water bodies in multispectral images, compared to 95.80% for the CNN model and 95.74% for the random forest model. In addition, the faster region-based CNN model significantly outperformed the CNN model and the random forest model for training speed.

6.2 Limitations

The first limiting factor in this project is data acquisition. Due to uncontrolled factors, the students we worked with were unable to obtain enough data to support our experiments. Due to the specific nature of the experimental data,

there was no publicly available dataset that could meet the data requirements for the experiments. This led to the fact that the only way to obtain as much data as possible was through image segmentation, which had some impact on the results of the experiment. At the same time, when collecting experimental data, the effects of different weather and lighting on experimental data were not taken into account, which led to some possible limitations in the transferability of experimental results, and the applicable scenarios required further research.

In addition, the experiment was conducted on Google Colab. Some cloud resource limitations and the ban on Google in mainland China affect the investigation. Although this did not directly affect the final results, the impact on the efficiency of the experiments may have limited the further development of the experiments, resulting in some work that could have been done becoming future work.

6.3 Future work

The most obvious next step is to determine the exact shape of the water body. All bodies of water in nature have irregular shapes, and identifying the location of a body of water through a rectangular bounding box does not accurately reflect the boundaries and size of the water area. Once the model trained with the faster R-CNN network can accurately identify the water body in the image, there are two possible ways to achieve the description of the water body shape.

The first approach is to take the intersection of all bounding boxes of the Faster R-CNN output, as shown in Figure 6.1, where the red rectangle represents the ground truth and the blue rectangle represents the possible bounding boxes of the output. Although this method does not miss any part of the ground truth as much as possible based on existing experiments, it still cannot accurately identify the edges of the water body, and cannot determine whether the edges of the water body are beyond or not beyond the edges of the real water body.

The second method is to identify edges by individual pixels. After determining the location of the water body, there are many ways to determine individual pixels, such as using the NDWI index with multispectral information of the image or using algorithms such as CNN and random forest.

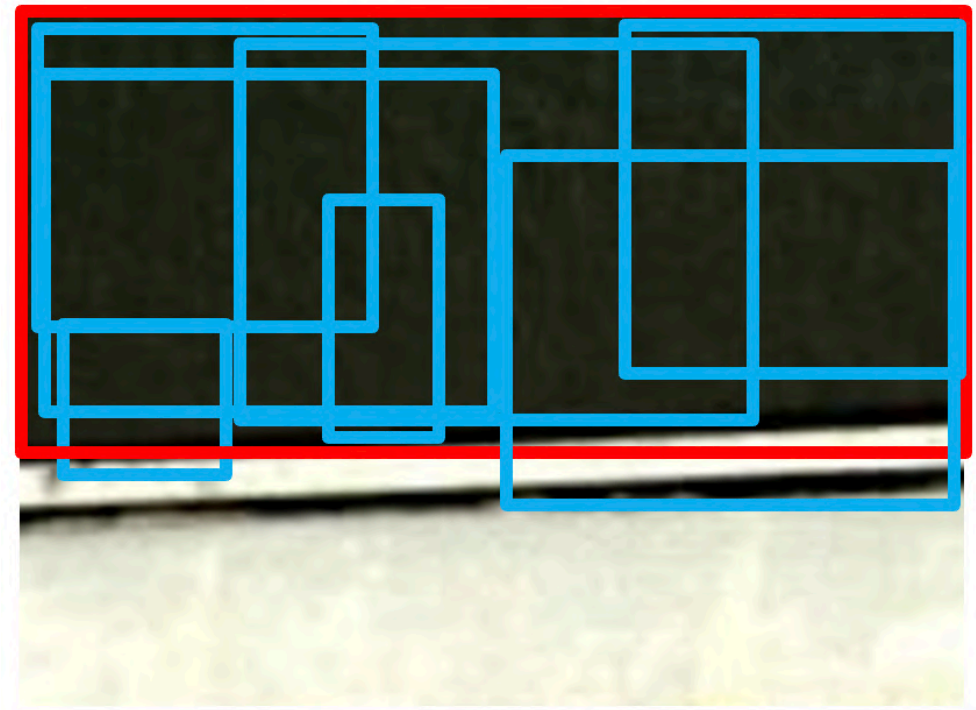


Figure 6.1: Intersection of bounding boxes

6.4 Reflections

The results of this paper will help with the use of drones to find scattered smaller bodies of water near human settlements in tropical jungles or similar environments. These bodies of water are often likely to be habitats for malaria mosquito habitats or mosquitoes that cause other infectious diseases. In further studies, data from water bodies containing malaria mosquitoes could be obtained for experiments.

References

[1] M. M. Petrou and C. Petrou, *Image processing: the fundamentals*. John Wiley & Sons, 2010. [Page 5.]

[2] Khan Academy, "Light and photosynthetic pigments," https://www.khanacademy.org/science/biology/photosynthesis-in-plants/the-light-dependent-reactions-of-photosynthesis/a/light-and-photosynthetic-pigments, 2015. [Pages ix and 6.]

[3] "The rededge-p from micasense," https://micasense.com/rededge-p/, 2021. [Pages ix, 7, 28, and 29.]

[4] C. J. Tucker, "Red and photographic infrared linear combinations for monitoring vegetation," *Remote Sensing of Environment*, vol. 8, no. 2, pp. 127–150, 1979. doi: https://doi.org/10.1016/0034-4257(79)90013-0. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0034425779900130 [Page 8.]

[5] "What is NDVI (normalized difference vegetation index)?" https://gisgeography.com/ndvi-normalized-difference-vegetation-index/, 2021. [Pages ix and 9.]

[6] Bo-Cai Gao, "NDWI—a normalized difference water index for remote sensing of vegetation liquid water from space," *Remote Sensing of Environment*, vol. 58, no. 3, pp. 257–266, 1996. doi: https://doi.org/10.1016/S0034-4257(96)00067-3. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425796000673 [Page 9.]

[7] "Ndwi normalized difference water index," https://custom-scripts.sentinel-hub.com/custom-scripts/sentinel-2/ndwi/, 2020. [Pages ix and 10.]

[8] E. H. Wilson and S. A. Sader, "Detection of forest harvest type using multiple dates of landsat tm imagery," *Remote Sensing of Environment*, vol. 80, no. 3, pp. 385–396, 2002. doi: https://doi.org/10.1016/S0034-4257(01)00318-2. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0034425701003182 [Page 10.]

[9] S. K. McFEETERS, "The use of the normalized difference water index (ndwi) in the delineation of open water features," *International Journal of Remote Sensing*, vol. 17, no. 7, pp. 1425–1432, 1996. doi: 10.1080/01431169608948714. [Online]. Available: https://doi.org/10.1080/01431169608948714 [Page 10.]

[10] H. Xu, "Modification of normalised difference water index (ndwi) to enhance open water features in remotely sensed imagery," *International Journal of Remote Sensing*, vol. 27, no. 14, pp. 3025–3033, 2006. doi: 10.1080/01431160600589179. [Online]. Available: https://doi.org/10.1080/01431160600589179 [Page 10.]

[11] S. L. Tanimoto, "An iconic/symbolic data structuring scheme," *Pattern recognition and artificial intelligence*, pp. 452–471, 1976. [Page 13.]

[12] S.-K. Chang, Q.-Y. Shi, and C.-W. Yan, "Iconic indexing by 2-d strings," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 3, pp. 413–428, 1987. doi: 10.1109/TPAMI.1987.4767923 [Page 13.]

[13] R. A. Fisher, "The use of multiple measurements in taxonomic problems," *Annals of eugenics*, vol. 7, no. 2, pp. 179–188, 1936. [Page 13.]

[14] D. R. Cox, "The regression analysis of binary sequences," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 20, no. 2, pp. 215–232, 1958. [Page 14.]

[15] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain." *Psychological review*, vol. 65, no. 6, p. 386, 1958. [Page 14.]

[16] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE transactions on information theory*, vol. 13, no. 1, pp. 21–27, 1967. [Page 14.]

[17] J. R. Quinlan, “Induction of decision trees,” *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986. [Page 14.]

[18] B. Li, J. Friedman, R. Olshen, and C. Stone, “Classification and regression trees (CART),” *Biometrics*, vol. 40, no. 3, pp. 358–361, 1984. [Page 14.]

[19] J. Quinlan, *C4.5: Programs for Machine Learning*, ser. Morgan Kaufmann series in machine learning. Elsevier Science, 1993. [Online]. Available: https://books.google.co.jp/books?id=HExncpjbYroC [Page 14.]

[20] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *nature*, vol. 323, no. 6088, pp. 533–536, 1986. [Page 14.]

[21] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation applied to handwritten zip code recognition,” *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989. [Pages 14 and 18.]

[22] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995. [Page 14.]

[23] Y. Freund, “Boosting a weak learning algorithm by majority,” *Information and computation*, vol. 121, no. 2, pp. 256–285, 1995. [Page 14.]

[24] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997. [Page 15.]

[25] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001. doi: 10.1023/A:1010933404324. [Online]. Available: https://doi.org/10.1023/A:1010933404324 [Pages 15 and 16.]

[26] K. Q. Weinberger and L. K. Saul, “Distance metric learning for large margin nearest neighbor classification.” *Journal of machine learning research*, vol. 10, no. 2, 2009. [Page 15.]

[27] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016. [Page 17.]

[28] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai *et al.*, "Recent advances in convolutional neural networks," *Pattern Recognition*, vol. 77, pp. 354–377, 2018. [Page 17.]

[29] W. Zhang, J. Tanida, K. Itoh, and Y. Ichioka, "Shift-invariant pattern recognition neural network and its optical architecture," in *Proceedings of annual conference of the Japan Society of Applied Physics*. Montreal, CA, 1988, pp. 2147–2151. [Page 17.]

[30] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology*, vol. 160, no. 1, p. 106, 1962. [Page 18.]

[31] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and cooperation in neural nets*. Springer, 1982, pp. 267–285. [Page 18.]

[32] K. Fukushima, "Cognitron: A self-organizing multilayered neural network," *Biological cybernetics*, vol. 20, no. 3, pp. 121–136, 1975. [Page 18.]

[33] J. Weng, N. Ahuja, and T. S. Huang, "Cresceptron: a self-organizing neural network which grows adaptively," in *[Proceedings 1992] IJCNN International Joint Conference on Neural Networks*, vol. 1. IEEE, 1992, pp. 576–581. [Page 18.]

[34] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. [Page 19.]

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012. [Page 19.]

[36] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014. [Pages ix, 20, and 21.]

[37] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013. [Page 20.]

[38] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, December 2015. [Pages ix, 21, and 22.]

[39] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015. [Pages ix, 23, 24, and 33.]

[40] "Dji phantom 4," https://www.dji.com/se/phantom-4, 2017. [Pages ix and 29.]

[41] B. Jiang, R. Luo, J. Mao, T. Xiao, and Y. Jiang, "Acquisition of localization confidence for accurate object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. [Page 30.]

[42] R. Girshick, "Faster R-CNN (Python implementation)," https://github.com/rbgirshick/py-faster-rcnn/blob/master/models/pascal_voc/VGG16/faster_rcnn_alt_opt/faster_rcnn_test.pt, 2018. [Pages ix and 33.]

TRITA-EECS-EX-2022:931