



โครงการการเรียนการสอนเพื่อเสริมประสบการณ์

การสกัดทางน้ำอัดโน้มน้ติจากภาพถ่ายดาวเทียมโดยใช้โครงข่ายเซลล์ประสาท
เสมือนเชิงลึก พื้นที่ศึกษาลุ่มแม่น้ำมูลของประเทศไทย

โดย

นายภากร อินทศสิงห์

เลขประจำตัวนิติต 6032730423

โครงการนี้เป็นส่วนหนึ่งของการศึกษาระดับปริญญาตรี
ภาควิชาธรณีวิทยา คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2563

การสกัดทางน้ำอัตโนมัติจากภาพถ่ายดาวเทียมโดยใช้โครงข่ายเซลล์ประสาทเสมือนเชิงลึก พื้นที่ศึกษาลุ่มแม่น้ำมูล
ของประเทศไทย

นายภากร อินทศสิงห์

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

ภาควิชาธรณีวิทยา คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2563

AUTOMATED CHANNEL-NETWORK EXTRACTION FROM SATELLITE IMAGES USING DEEP
CONVOLUTIONAL NEURAL NETWORKS: A CASE STUDY OF MUN RIVER IN THAILAND

MR.PHAKORN INTASSINGHA

A Project Submitted in Partial Fulfillment of the Requirements
for the Degree of Bachelor of Science Program in Geology
Department of Geology, Faculty of Science, Chulalongkorn University

Academic Year 2020

หัวข้อโครงการ การสกัดทางน้ำอัตโนมัติจากภาพถ่ายดาวเทียมโดยใช้โครงข่ายเซลล์ประสาทเสมือน
เชิงลึก พื้นที่ศึกษาลุ่มแม่น้ำมูลของประเทศไทย

โดย นายภากร อินทศสิงห์

สาขาวิชา ธรณีวิทยา

อาจารย์ที่ปรึกษาหลัก อาจารย์ ดร.พงศ์เทพ ทองแสง

วันที่ส่ง..... 14 พ.ค. 2564

วันที่อนุมัติ..... 7 พ.ค. 2564



.....
อาจารย์ที่ปรึกษาโครงการหลัก

(อาจารย์ ดร. พงศ์เทพ ทองแสง)

Project Title AUTOMATED CHANNEL-NETWORK EXTRACTION FROM SATELLITE IMAGES
 USING DEEP CONVOLUTIONAL NEURAL NETWORKS: A CASE STUDY OF
 MUN RIVER IN THAILAND

By Mr. Phakorn Intassingha

Field of Study Geology

Project Advisor Pongthep Thongsang, Ph.D.

Submitted date..... 14 May 2021

Approval date..... 7 May 2021



.....
Project Advisor

(Pongthep Thongsang, Ph.D.)

ภากร อินทศสิงห์ : การสกัดทางน้ำอัตโนมัติจากภาพถ่ายดาวเทียมโดยใช้โครงข่ายเซลล์ประสาทเสมือนเชิงลึก พื้นที่ศึกษาลุ่มแม่น้ำมูลของประเทศไทย (AUTOMATED CHANNEL-NETWORK EXTRACTION FROM SATELLITE IMAGES USING DEEP CONVOLUTIONAL NEURAL NETWORKS: A CASE STUDY OF MUN RIVER IN THAILAND)

อ.ที่ปรึกษาโครงการหลัก : อาจารย์ ดร. พงศ์เทพ ทองแสง, 41 หน้า.

แม่น้ำและทะเลสาบรูปแอก มีความสำคัญมากในภาคเกษตรกรรม ของประเทศไทยเนื่องจากเป็นแหล่งน้ำที่สำคัญในการทำการเกษตรซึ่งเป็นอาชีพที่มีจำนวนแรงงานถึงร้อยละ 35 ของจำนวนแรงงานทั้งหมดในประเทศ (12.37 ล้านคน จาก 38.26 ล้านคน) (สำนักงานสถิติแห่งชาติ, 2561) ซึ่งมักจะประสบภัยแล้ง หรือน้ำท่วมเนื่องจากการจัดสรรทรัพยากรน้ำในพื้นที่ซึ่งไม่มีประสิทธิภาพส่งผลให้ผู้ประสบภัยขาดแคลนรายได้

ในการศึกษาและจัดสรรทรัพยากรน้ำโดยใช้ภาพถ่ายดาวเทียมในปัจจุบันมีด้วยกัน 3 วิธีคือ 1. ใช้มนุษย์สร้างขอบเขตของวัตถุที่เป็นน้ำโดยใช้โปรแกรม ArcGIS หรือ โปรแกรม QGIS ซึ่งวิธีการนี้จะมีข้อเสียคือเกิดความคลาดเคลื่อนจากผู้สร้างขอบเขตได้ง่ายและยังสิ้นเปลืองเวลาค่อนข้างมาก 2. ใช้ดัชนีน้ำในการสกัดวัตถุที่เป็นน้ำออกมาจากภาพถ่ายดาวเทียม 3. ใช้การเรียนรู้ของเครื่องจักรในการจัดกลุ่มวัตถุที่เป็นน้ำออกจากวัตถุอื่น โดยข้อเสียของทั้งวิธีที่ 2 และ 3 คือจะมีวัตถุที่เป็นน้ำที่ไม่ต้องการศึกษา และวัตถุที่ไม่ใช่ น้ำเข้ามารบกวน ทำให้การศึกษาเฉพาะพื้นที่เป็นไปได้ยาก ดังนั้นงานวิจัยนี้จึงประยุกต์ใช้โครงข่ายเซลล์ประสาทเสมือนเชิงลึกเพื่อที่จะสกัดวัตถุที่เป็นแม่น้ำและทะเลสาบรูปแอกออกมาพร้อมทั้งคำนวณพื้นที่ของวัตถุดังกล่าวอย่างแม่นยำได้อย่างอัตโนมัติ เพื่อช่วยในการจัดสรรทรัพยากรน้ำในพื้นที่ต่าง ๆ ได้อย่างรวดเร็วและมีประสิทธิภาพ

ภาควิชา ธรณีวิทยา

ลายมือชื่อนิสิต..... *ภากร อินทศสิงห์*

สาขาวิชา ธรณีวิทยา

ลายมือชื่อ อ.ที่ปรึกษาหลัก..... *พงศ์เทพ ทองแสง*

ปีการศึกษา 2563

6032730423: MAJOR GEOLOGY

KEYWORDS: DEEP CONVOLUTIONAL NEURAL NETWORKS / CHANNEL EXTRACTION / SATELLITE IMAGES

PHAKORN INTASSINGHA: AUTOMATED CHANNEL-NETWORK EXTRACTION FROM SATELLITE IMAGES USING DEEP CONVOLUTIONAL NEURAL NETWORKS: A CASE STUDY OF MUN RIVER IN THAILAND.

ADVISOR PONGTHEP THONGSANG, Ph.D, 41 pp.


Main river and oxbow lake are significant factors in Thai agriculture for producing agricultural production. The National Statistical Office Thailand stated that in 2018, agriculturists, which has the percentage of 35 of the total labor forces or 12.37 million from 38.26 million face, lack income due to a low productivity from the drought or flood caused by the inefficient water resource management.

In the present day, the study of water resource management by using satellite images has three methods. First of all, using human digitization by ArcGIS or QGIS. The disadvantages of this method are that it has high human error and is time consuming. Secondly, using water index (NDWI) to extract water bodies from satellite images. Lastly, using machine learning to cluster the objects into each class. The weakness of the second and third methods is that there are water bodies beyond the study scope and solid bodies disrupt the specific area identification. Therefore, this study applies deep convolutional neural networks to extract Main river and Oxbow lake specifically together with calculating water surface highly accurately and automatically to improve the water resource management in local areas rapidly and efficiently.

Department: Geology

Student's Signature.....

Field of Study: Geology

Advisor's Signature.....

Academic Year: 2020

Acknowledgements

This project is accomplished with a lot of assistance. There are abundant people and environments that I would like to express thanks for.

First, I would like to express my gratitude to the kindness of MMDetection (Chen et al., 2019), the open-source of object detection toolbox and Labelling tool from Tzutalin, 2015 for providing a facility in this study.

Secondly is my advisor, Pongthep Thongsang, Ph.D. It is a whole-hearted expression that your dedicated time and advice are proved to be a major part toward the success of my project. He also support me both in academic and living problems.

Next, I would like to give special thanks to Miss Supakjarin Wijitpongsa, she is my friend who get involved in helping me accomplish this project. Regardless of whether the problem is difficult or easy, they both kindly assist me fix those problems.

Lastly, I would like to thank all of the Department of Geology staffs for providing support and places for me during this study.

Phakorn Intassingha

Author

Table of Contents

Chapter 1	1
1.1 Introduction	1
1.2 Objective2	
1.3 The scope of this study.....	2
1.4 Study area	2
Chapter2	3
2.1 Object detection.....	3
2.2 Satellite Image.....	5
2.2.1 Sentinel-2.....	5
2.3 K-means clustering	6
2.4 Convolutional Neural Networks (CNN)	7
2.5 Convolutional layer.....	8
2.5.1 Pooling layer.....	9
2.5.2 Softmax layer	9
2.6 Faster R-CNN and ResNet-50	10
2.6.1 Faster R-CNN.....	11
2.6.2 ResNet-50.....	13
2.7 RetinaNet-50	14
Chapter3	16
3.1 Overview of the workflow	16
3.2 Data collection and feature engineering	17
3.2.1 Download satellite images from Google Earth Engine	17

3.2.2	Using K-means clustering to classify water body	18
3.3	Labeling	21
3.4	Building neuron network architecture.....	22
3.5	Training models.....	23
Chapter4	26
4.1	Evaluate models.....	26
4.2	Testing models.....	30
4.3	The evaluation of water surface areas.....	31
Chapter5	36
5.1	Model performance	36
5.2	User's experience.....	36
5.3	Limitation	42
5.4	Future works.....	42

List of Figures

Figure 1.1 Study area map with 480x190 square meters.	2
Figure 2.1 Comparing model performance between SSD (a and c) and Faster R-CNN (b and d) (Huang et al., 2016).....	4
Figure 2.2 Comparing model performance between SSD, Faster R-CNN, RetinaNet by using mAP (Li et al., 2020).....	4
Figure 2.3 Illustrate of K-means clustering process (Yildirim, 2020).....	6
Figure 2.4 Schematic of overall CNN layer which compose of Input layer, Hidden layer and Output layer. (Otuyama, 2000).....	7
Figure 2.5 Illustrate of function of convolutional layer. Blue box is pixel value of input image, Green box is filter or kernel 3x3, Red box is sum of values from each pixel multiply by filter 3x3 called Feature Map (Dernart, 2017).....	8
Figure 2.6 Illustrate of max pooling `2x2. It gets the max pixel value of each 2x2 windows around the image then the result still preserves characteristic of the image (Dernart, 2017).	9
Figure 2.7 Processing of the Softmax activation function that use vector of output layer to calculate in the equation to get the probability of each class in detected object.....	9
Figure 2.8 Schematic of RCNN and Faster R-CNN model architecture (Li et al.,2018)	10
Figure 2.9 Illustrate of Rol pooling proess ordered by 1-4 (https://blog.deepsense.ai/).....	12
Figure 2.10 Illustrate of ResNet-50 architecture (He et al., 2016)	14
Figure 2.11 Schemation of RetinaNet model architecture (Lin et al, 2017)	15
Figure 3.1 Schematic of workflow in this study	16
Figure 3.2 Study area in Google Earth Engine (GEE) with downloaded polygon.....	17
Figure 3.3 Two samples of band432 of downloaded images (left 061.tiff, right 077.tiff).....	17
Figure 3.4 Reflectance curve of water (blue line), Vegetation (green line) and Soil (red line) along the wavelength in range 0-2.5 micrometer (Serco Italia SPA, 2017)	18
Figure 3.5 Comparing plot between Band 7 and Band 8 of image 066.tiff.....	19
Figure 3.6 Comparing plot between Band 7 and Band 8 of image 071.tiff.....	19

Figure 3.7 Two samples of band 8 of downloaded images	20
Figure 3.8 Two samples of classified images which is classified into two classes: 1. Water (violet color), 2. Land & vegetation (yellow color)	20
Figure 3.9 Lebellmg tool from Labellmg from Tzutalin, 2015	21
Figure 3.10 Illustrate of the amount of labeled object in each classes.....	21
Figure 3.11 Illustrate of Faster R-CNN architecture. (Modify from Mohan and Vinayakumar et al., 2019)	22
Figure 3.12 Illustrate of Retinanet architecture. (Modify from Zeng, 2019)	22
Figure 3.13 Result of features map in layer 1 (left) and layer 49 (right) when pass the classified image into Resnet-50.....	24
Figure 3.14 Functioning of Momentum parameter is helped the gradient of SGD to skip the local minima. (Dong, 2019).....	24
Figure 3.15 Illustrate of Region Proposal which is process of creating bouding boxes in object detection task. a) Create anchors whole of the image. b) Create bounding boxes from anchors. c) Filter the usable boxes by using IoU d) Filtered bounding boxes.....	25
Figure 4.1 Illustrate of learning curve in training process of Faster R-CNN ResNet-50. The model gets loss of class around 0.02 and loss of bounding box around 0.05	27
Figure 4.2 Illustrate of learning curve in training process of Faster R-CNN ResNet-50. The model gets loss of class around 0.01 and loss of bounding box around 0.06	28
Figure 4.3 Illustrate of learning curve in training process of RetinaNet50. The model gets mAP value around 0.67	29
Figure 4.4 Illustrate of learning curve in training process of Faster R-CNN ResNet-50. The model gets mAP value around 0.78.....	29
Figure 4.5 test image number test001.jpg. Result of prediction from Faster R-CNN Resnet50 (Left), and RetinaNet50 (Right).....	30
Figure 4.6 test image number test002.jpg. Result of prediction from Faster R-CNN ResNet50 (Left) is more accurate than RetinaNet50 (Right) because 4 oxbow lakes and 1 main channel in	

the image can be completely detected by Faster R-CNN ResNet-50. While Retinanet50 can detect just 2 oxbow lakes. Moreover, the precision of creating bounding boxes of Faster R-CNN ResNet-50 is higher than Retinanet50	31
Figure 4.7 Cropped image of main channel from test001.jpg (Figure 4.5 (left)). Detected object (left) and cleaned image (right) that can be measured a water surface areas equal to 488,000 m ²	32
Figure 4.8 Cropped image of oxbow lake from test001.jpg (Figure 4.5 (left)). Detected object (left) and cleaned image (right) that can be measured a water surface areas equal to 231,800 m ²	33
Figure 4.9 Cropped image of main channel from from test002.jpg (Figure 4.6 (left)). Detected object (left) and cleaned image (right) that can be measured a water surface areas equal to 2,592,400 m ²	33
Figure 4.10 Cropped image of oxbow lake from from test002.jpg (Figure 4.6 (left)). Detected object (left) and cleaned image (right) that can be measured a water surface areas equal to 877,700 m ²	34
Figure 4.11 Cropped image of oxbow lake from from test002.jpg (Figure 4.6 (left)). Detected object (left) and cleaned image (right) that can be measured a water surface areas equal to 375,400 m ²	34
Figure 4.12 Cropped image of oxbow lake from from test002.jpg (Figure 4.6 (left)) detected object (left) and cleaned image (right) that can be measured a water surface areas equal to 256,000 m ²	34
Figure 4.13 Cropped image of oxbow lake from from test002.jpg (Figure 4.6 (left)) detected object (left) and cleaned image (right) that can be measured a water surface areas equal to 491,200 m ²	35
Figure 5.1 Bar plot compare models performance by using value of mAP	36
Figure 5.2 Author's Github repository for implement this study https://github.com/phakornintt/channel_network_extraction	37

Figure 5.3 Illustrate of Tutorial notebook in Github. Author recommends new users enter the tutorial notebook in Google Colab.....	37
Figure 5.4 How to copy Google Colab Notebook to your own Google Colab	38
Figure 5.5 Drag and drop the input satellite image (tiff file) into Google Colab.....	38
Figure 5.6 Change your own file name in some line of python code before execute it.	39
Figure 5.7 Click Run all to execute all python code in the notebook.	39
Figure 5.8 test_002.tif will be transformed to classified image (test_002.jpg) which is collected in transform_kmean folder.....	40
Figure 5.9 Main channel and oxbow lakes will be detected automatically.....	40
Figure 5.10 At the end of tutorial, users will receive all cleaned images of detected objects (a), and csv file (b) in result folder automatically.....	41
Figure 5.11 Users can download the result file into own local computer by click the Download button.....	41
Figure 5.12 Future works of this study:	42

List of Tables

Table 1 Resolution and wavelength in each bands of Sentinel-2.....	5
Table 2 Each ResNet architectures is different from the amount of convolution layers. (He et al., 2016)	14
Table 3 The best SGD's parameters of each models	26

Chapter 1

Introduction

1.1 Introduction

Machine learning is becoming more widely used for classification objects; for instance, urban monitoring, fire detection, flood prediction from remotely sensed multispectrum, and radar images. They significantly impact economic and environmental issues (Camps-Valls, 2009). The remote sensing applications are classified objects in satellite images such as urban, agriculture, stream channel, etc., but some problems are found. The supervised classification needs to add the input features which are in the training process, to define the object. If the training data is insufficient or not representative, the classification results will also be inaccurate. Similarly, the unsupervised classification requires adjusting the best threshold of parameters to classify accurately. Even though each satellite image is taken in the same area, they have reflectance values differently due to light and season factors. Therefore, either the supervised and unsupervised learnings have a high cost to fine-tune the parameters (Richards, 2013).

This study aims to create an automated system to extract stream channels and oxbow lakes in the satellite images using a convolutional neural network (CNN). For machine learning, extracting the channels and oxbow lakes have to define the training area or refine the threshold in every image because of reflectance values of the channels various in each time which factors are sediments in the channels, cloud, and sunlight. Unlike the CNN method, CNN is the most popular deep learning architecture, a mathematic operation series containing a convolution layer and pooling layer. It extracts the objects' features such as shape and color along the operation to identify what the object is (Dernart, 2017), which makes CNN automatically extract the stream channels and oxbow lakes without passing through the training process or refine the parameters in every image. When channels and oxbow lakes can be extracted from satellite images automatically, besides saving a lot of time. Thus, applying CNN to extract the channels can use for further planning, improve water management and agriculture in the local area.

1.2 Objective

1. To create a tool that can be automated extract main channel and oxbow lake by using object detection
2. To compare model performance between Faster R-CNN ResNet50 and RetinaNet-50

1.3 The scope of this study

Use satellite image from Sentinel-2 along the Mun river on January-May 2020 to train model and detect main channels and oxbow lakes in the images by using K-means clustering, object detection base on faster R-CNN ResNet-50 FPN. The total number of the satellite image is 98 images which 80% of images are used as a training set, 20% are used as a validation set.

1.4 Study area

Along the Mun River in Thailand compose of Ubon Ratchathani province, Si Sa Ket province, Surin province, Buri Ram province, Nakhon Ratchasima province (Figure 1.1)



Figure 1.1 Study area map with 480x190 square meters.

Chapter2

Literature reviews

2.1 Object detection

Object detection is the process of deep learning algorithms to allow the computer or machine to specify the position of objects in form of a bounding box in images. Object detection is applied in many industries such as detected cancer in the medical industry, autonomous driving in the vehicle industry, or face recognition in security industry

Object detection has a lot of models that purposed by many researchers such as SSD, RetinaNet, YOLOv3, Fast R-CNN, Faster R-CNN, etc.

Testing the performance of models, researchers use a standard dataset: PASCAL Visual Object Classes (VOC) or COCO to compare the accuracy of their own model with other models by using mAP value.

Research of google researchers (Huang et al., 2016) compare the accuracy of SSD, Faster R-CNN model. In the research, they compare the result of two models by using several images (Figure 2.1). The result is Faster R-CNN model is more accurate than the SSD model because the SSD model can't detect some small objects and has an overall mAP smaller than Faster R-CNN.

Object detection in remote sensing images by Li et al., 2020 is another research. This study aims to compare the performance of SSD, Faster R-CNN, and RetinaNet models using the mAP value to detect objects in aerial photographs and satellite images. According to the findings, the Faster R-CNN and RetinaNet have a higher mAP value than the SSD model (Figure 2.2).

Therefore, the aim of this study will compare the performance between Faster R-CNN and RetinaNet by using mAP value.

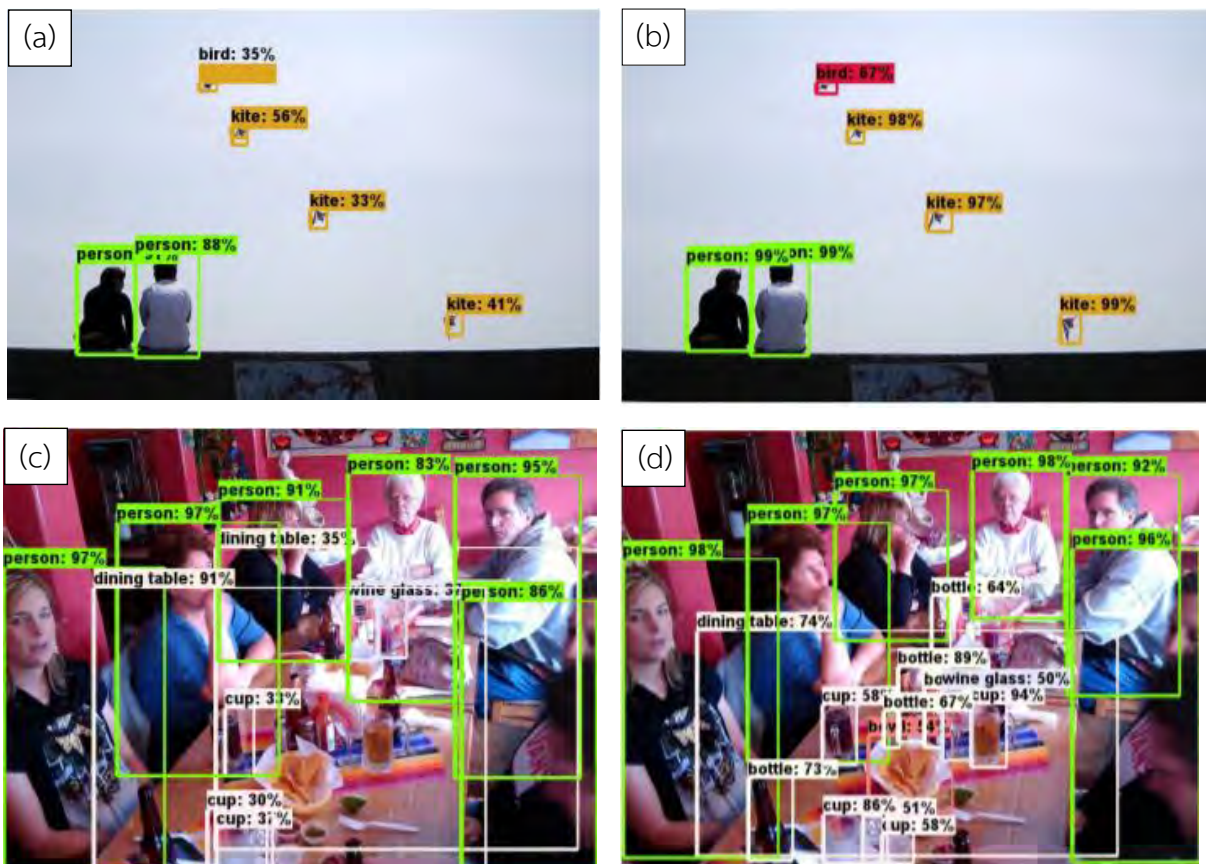


Figure 2.1 Comparing model performance between SSD (a and c) and Faster R-CNN (b and d) (Huang et al., 2016)

Model	mAP (Accuracy)
SSD	58.6
Faster R-CNN	63.1
RetinaNet	65.7

Figure 2.2 Comparing model performance between SSD, Faster R-CNN, RetinaNet by using mAP (Li et al., 2020)

2.2 Satellite Image

2.2.1 Sentinel-2

Sentinel-2 is satellite which has a wide-swath, high-resolution, multi-spectral imaging mission supporting Copernicus Land Monitoring studies, including the monitoring of vegetation, soil and water cover, as well as observation of inland waterways and coastal areas. Sentinel-2

In this study uses Sentinel-2 Level-2A which has a product is corrected by any distortion of atmosphere, terrain and cirrus clouds.

Table 1 Resolution and wavelength in each bands of Sentinel-2

Name	Description	Resolution (m.)	Wavelength (nm.)
B1	Aerosols	60	443.9
B2	Blue	10	496.6
B3	Green	10	560
B4	Red	10	664.5
B5	Red Edge 1	20	703.9
B6	Red Edge 2	20	740.2
B7	Red Edge 3	20	782.5
B8	NIR	10	835.1
B8A	Red Edge 4	20	864.8
B9	Water vapor	60	945
B10	SWIR cirrus	60	1375
B11	SWIR 1	20	1613.7
B12	SWIR 2	20	2202.4

2.3 K-means clustering

K-means clustering aims to partition data into k clusters in a way that data points in the same cluster are similar and data points in the different clusters are farther apart. A similarity of data points is determined by the distance between them. Measuring distance between data points in K-means is used Euclidean distance (Equation 1).

$$\text{Euclidean distance} = \sqrt{(x_1^2 - y_1^2) + (x_2^2 - y_2^2)} \quad (\text{Equation 1})$$

Process of K-means is iterative, it works by executing the following steps (Figure 2.3):

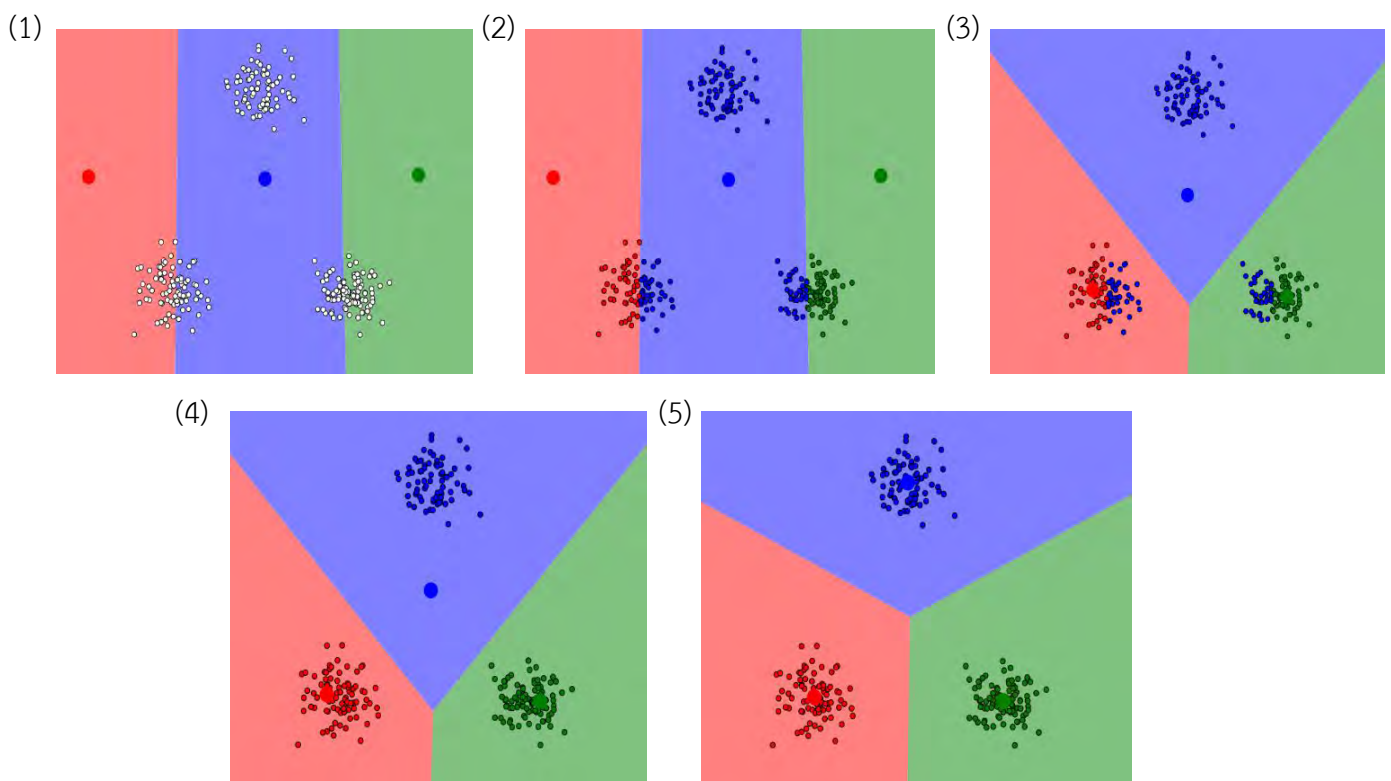


Figure 2.3 Illustrate of K-means clustering process (Yildirim, 2020)

1. Randomly select centroids (center of cluster) for each cluster.
2. Calculate the Euclidean distance of all data points to the centroids.
3. Assign data points to the closest cluster.
4. Find the new centroids of each cluster by taking the mean of all data points in the cluster.
5. Repeat steps 2,3 and 4 until all points converge and cluster centers stop moving.

2.4 Convolutional Neural Networks (CNN)

A Convolutional Neural Network (CNN) is a Deep learning algorithm which can take in an input image, learnable weights and biases to various aspects/objects in the image and be able to differentiate one from the other. The CNN compose of many layers such as Convolution layer, Pooling layer. All of the layers have connectivity pattern of neurons like human brain and have different function (Figure). The shared weights parameter in Figure 2.4 is a significant number that controls probability of predicted class. Then, each weights will be optimized in training phase with optimizer such as Stochastic Gradient Descent (SGD), Adaptive Moment Estimation (Adam).

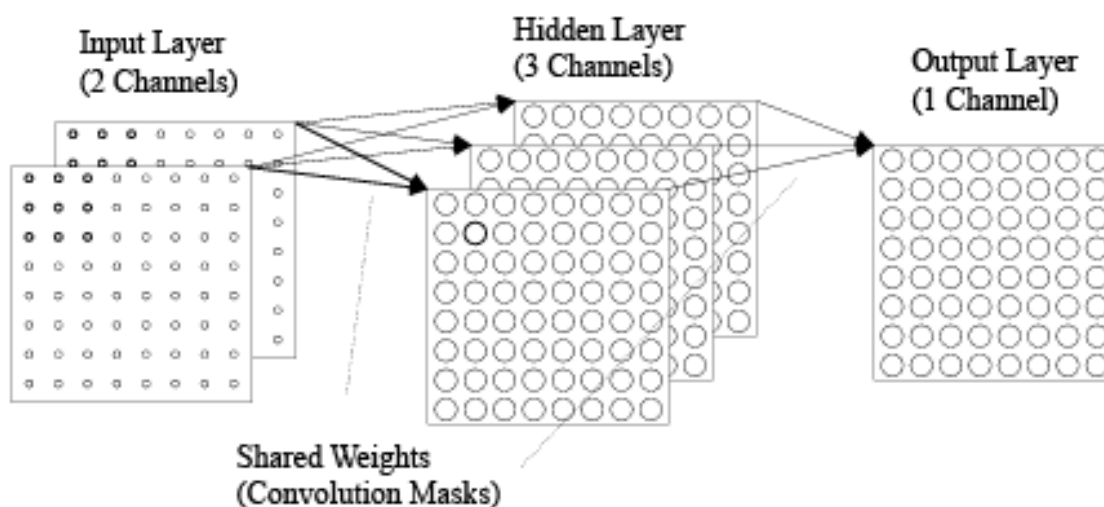


Figure 2.4 Schematic of overall CNN layer which compose of Input layer, Hidden layer and Output layer. (Otuyama, 2000)

2.5 Convolutional layer

A convolutional layer is the one of Hidden Layer in Figure 2.4 which composes of filter or kernel that has size 3x3 (green boxes) in the Figure 2.5. The filter will multiply over the pixel values of the input image (blue boxes) and sum all result in the filter box then serve the output called feature map (red boxes)

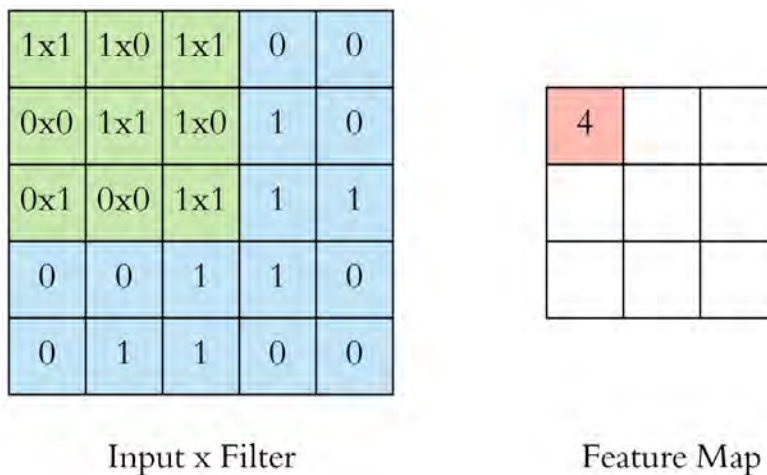


Figure 2.5 Illustrate of function of convolutional layer. Blue box is pixel value of input image, Green box is filter or kernel 3x3, Red box is sum of values from each pixel multiply by filter 3x3 called Feature Map (Dernart, 2017).

2.5.1 Pooling layer

After the convolution operation, the feature map is passed into a pooling layer to reduce the height and width of the feature map for avoid overfitting and reduce training time. The most common type of pooling is max pooling which just takes the max value in the pooling window (Figure 2.6).

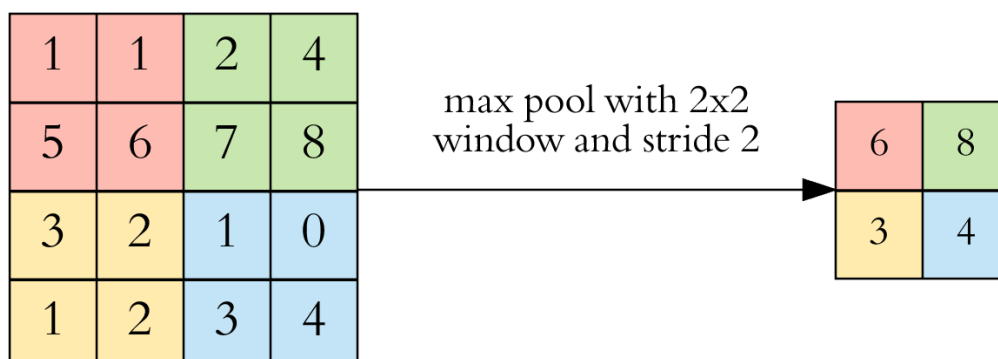


Figure 2.6 Illustrate of max pooling `2x2. It gets the max pixel value of each 2x2 windows around the image then the result still preserves characteristic of the image (Dernart, 2017).

2.5.2 Softmax layer

A softmax layer is the lastest layer of neural network which contain the activation function called softmax function. The softmax function transforms each value of output layer into probabilities of each class of the input image by use the equation in Figure 2.8.

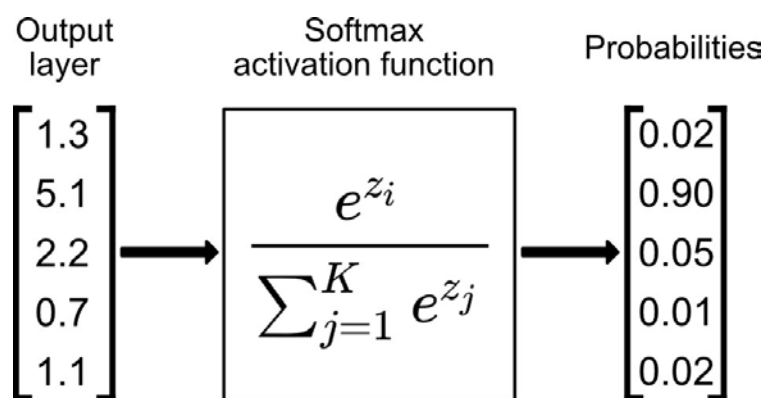


Figure 2.7 Processing of the Softmax activation function that use vector of output layer to calculate in the equation to get the probability of each class in detected object.

2.6 Faster R-CNN and ResNet-50

Faster R-CNN ResNet-50 model compose of ResNet-50 which is convolutional neural network (CNN), Faster R-CNN (Figure 2.9 (b)). The model is used for object detection task.

The input image is passed through the process in Figure 2.9 (b), then visualize the loss of classification and bounding box around the predicted class. For each part of the architecture will be explained in next topic.

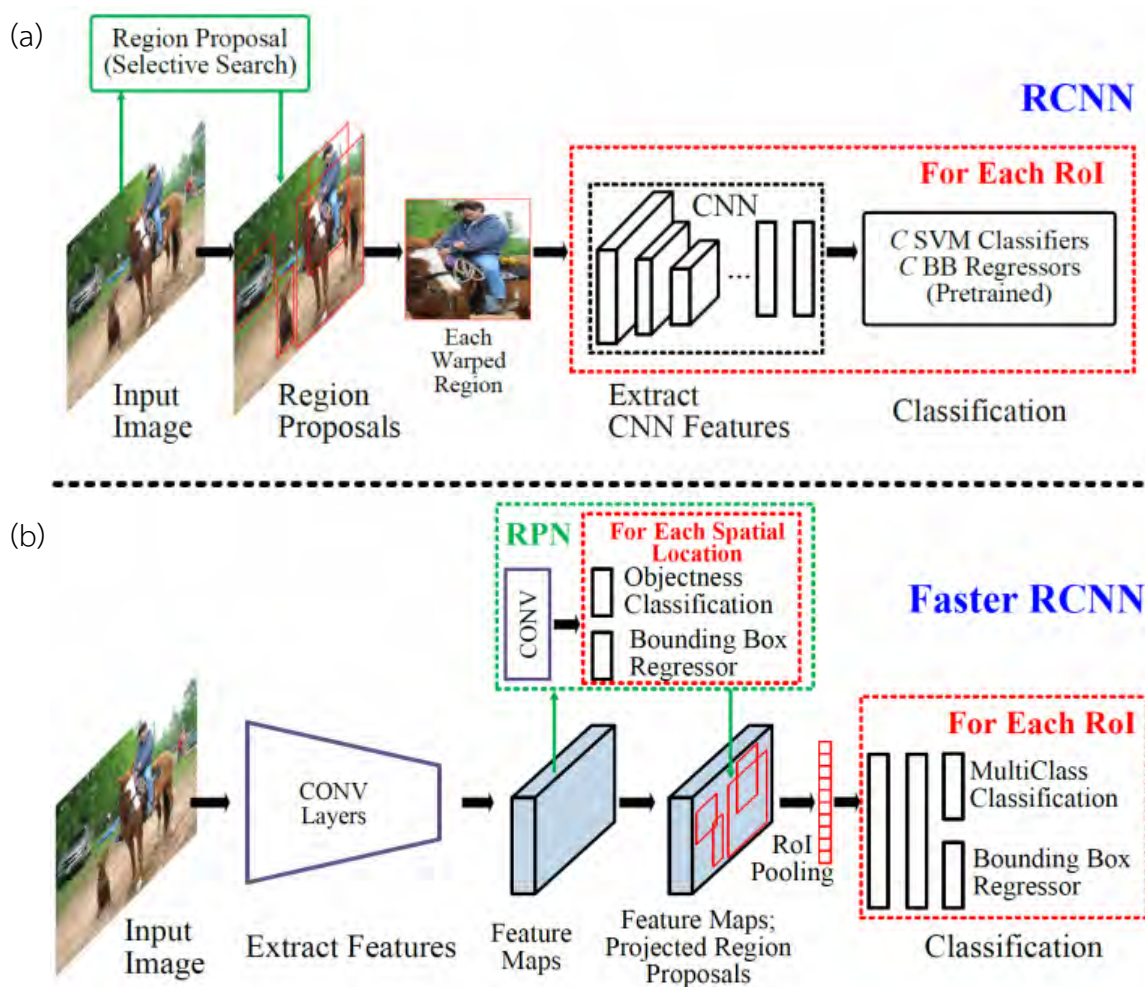


Figure 2.8 Schematic of RCNN and Faster R-CNN model architecture (Li et al.,2018)

2.6.1 Faster R-CNN

In the normal R-CNN model (Figure 2.9 (a)) is proposed by Girshick et al., 2014. The goal of the model is defining the objects in an image by bounding box. The process of R-CNN is work with following steps (Lui et al., 2018):

1. **Selective Search (Region Proposal)**, creating random bounding box in the input image in k boxes.
2. **Image Warping**, preparing image in each bounding box by resize each image into the same size.
3. **Extracted Features**, each bounding box are extracted features by CNN model.
4. **Classification**, each bounding box are classified to each class.
5. **Bounding box Regression**, If probability of class more than 0.5, the regressor will generate an accurate box of that class.

However, training time of R-CNN is too much because the CNN have to extract the feature of each k proposed box. To solve this problem Ren et al. propose Faster R-CNN which faster than normal R-CNN in 100 times.

A Faster R-CNN concept is proposed by Ren et al., 2017 that using CNN model before Region proposal process. This concept reduces a huge of training time because CNN is computed in 1 time, then get feature maps (Figure 2.8 (b)). The feature maps are be the input to Region Proposal Network (RPN) and Roi Pooling.

2.6.1.1 Region Proposal Network (RPN)

RPN is another CNN model of Faster R-CNN that come to solve of slowing training of selective search. The function of RPN is learning and propose bounding boxes from the feature maps.

2.6.1.2 RoI Pooling (Region of Interest Pooling)

The function of RoI pooling quite similar to max pooling. But difference between Max pooling and RoI Pooling is the RoI separates region proposal (Figure 2.10 (2)) into their size (In Figure 2.10 (3) is 2x2) and each section can contain many pixel values. Output of the RoI is max pixel values of each section (Figure 2.10 (4)).

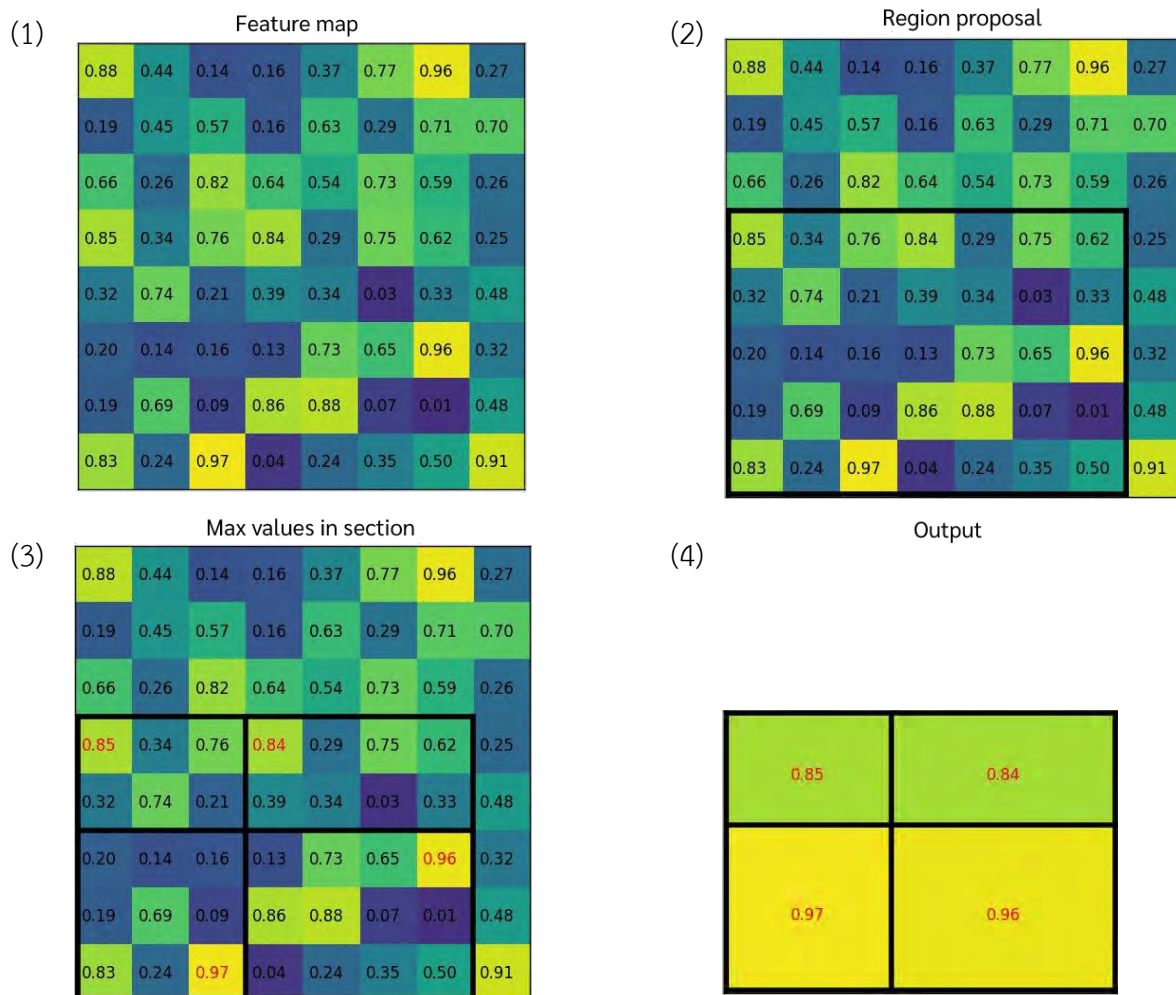


Figure 2.9 Illustrate of RoI pooling process ordered by 1-4 (<https://blog.deepsense.ai/>)

2.6.1.3 Intersection Over Union (IoU)

After RoI Pooling, the output (Figure 2.9 (4)) is passed through Classification layer like Soft-max and created bounding box of the class.

An accuracy of class prediction can be measured in binary term which is correct or wrong from labeled class. But the accuracy of bounding box has to use IoU method (Equation 2) to measure the accurate of model (Liu et al., 2018)

$$IoU(P, L) = \frac{|P \cap L|}{|P \cup L|} \quad (\text{Equation 2})$$

Where, P is area of predicted bounding boxes

L is area of labeled bounding boxes

2.6.2 ResNet-50

Residual Network (ResNet) is a deep convolutional neural network model (proposed by He et al., 2016) for solve the problem of deep neural network which is degradation and saturation of accuracy of a deep learning model when the depth of neural network is increased. ResNet used the skip connection to propagate information across layers to solve the problem.

Faster R-CNN ResNet-50 uses ResNet-50 to be the CNN model which compute the feature maps which contain the feature of labeled object of the input image such as color, shape, edge. ResNet-50 is a variant of ResNet model (Table 2) which has 48 Convolution layers along with 1 Max pooling and 1 Average pooling layer (Figure 2.10). It is a widely used ResNet model in deep learning and computer vision tasks base on PyTorch.

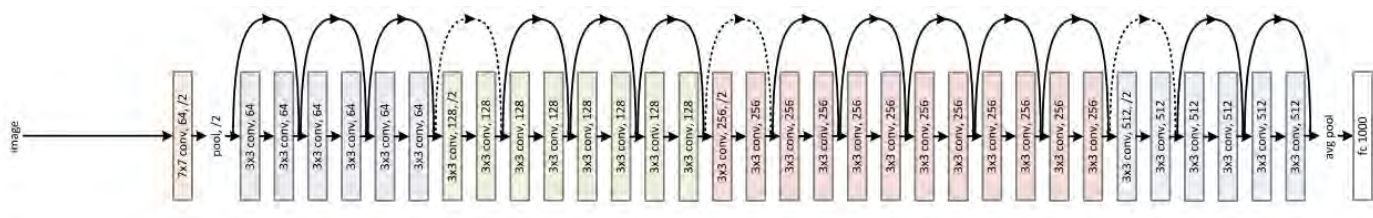


Figure 2.10 Illustrate of ResNet-50 architecture (He et al., 2016)

Table 2 Each ResNet architectures is different from the amount of convolution layers. (He et al., 2016)

		3×3 max pool, stride 2				
conv2_x	56×56	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$
conv3_x	28×28	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 8$
conv4_x	14×14	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 36$
conv5_x	7×7	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

2.7 RetinaNet-50

Traditionally, in computer vision, featurized image pyramids have been used to detect objects with varying scales in an image. Featurized image pyramids are feature pyramids built upon image pyramids. This means one would take an image and subsample it into lower resolution and smaller size images (thus, forming a pyramid).

With the advent of deep learning, these hand-engineered features were replaced by CNNs. Later, the pyramid itself was derived from the inherent pyramidal hierarchical structure of the CNNs. In a CNN architecture, the output size of feature maps decreases after each successive block of convolutional operations, and forms a pyramidal structure.

RetinaNet is proposed by Lin et al., 2017 which has four major components of a model architecture (Figure 2.11):

a) Bottom-up Pathway The backbone network in RetinaNet50 is ResNet-50 which was mentioned in 2.6.2.

b) Top-down pathway and Lateral connections The top down pathway upsamples the spatially coarser feature maps from higher pyramid levels, and the lateral connections merge the top-down layers and the bottom-up layers with the same spatial size.

c) Classification subnetwork It predicts the probability of an object being present at each spatial location for each anchor box and object class.

d) Regression subnetwork It's regresses the offset for the bounding boxes from the anchor boxes for each ground-truth object.

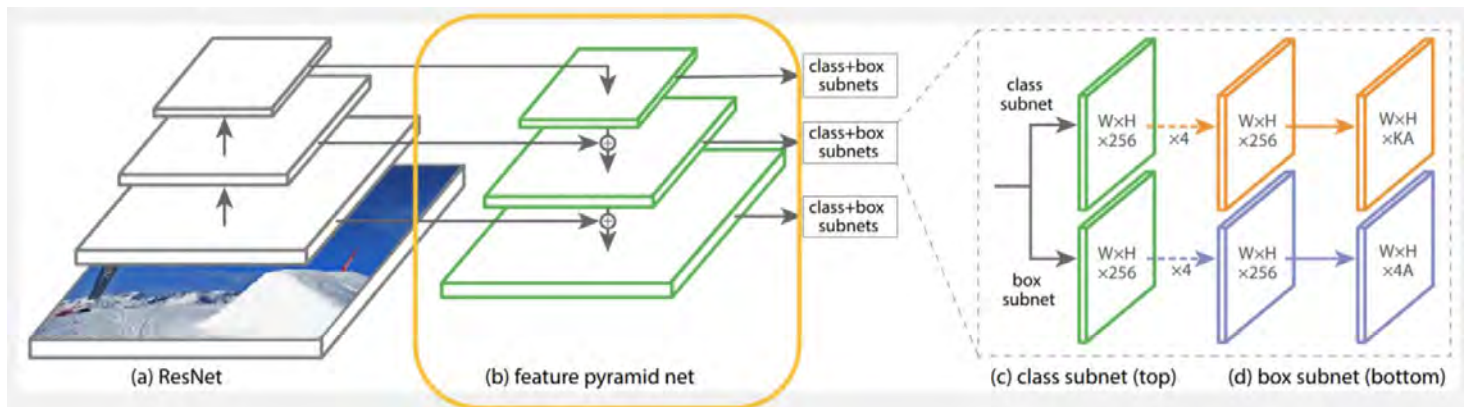


Figure 2.11 Schematic of RetinaNet model architecture (Lin et al, 2017)

Chapter3 Methodology

3.1 Overview of the workflow

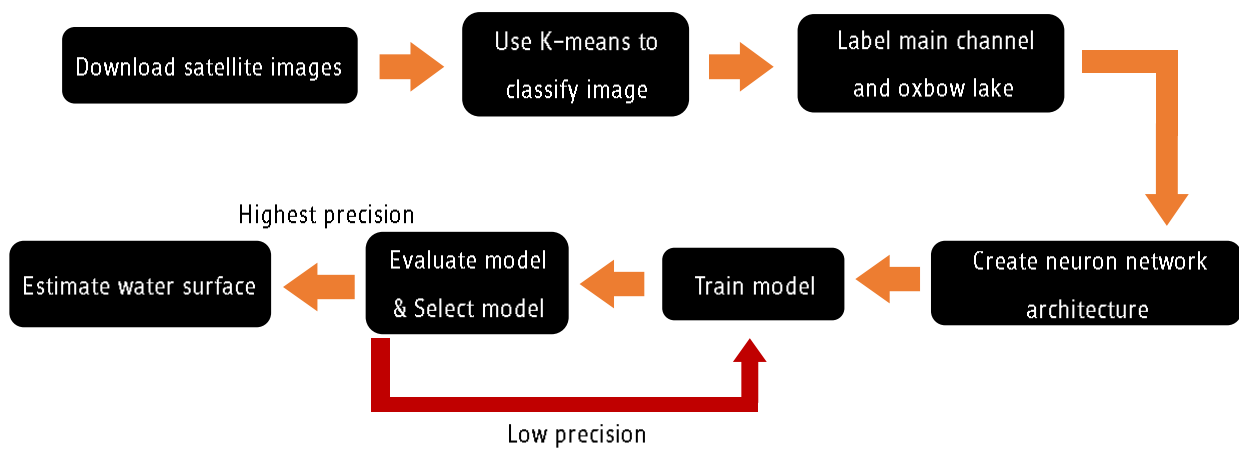


Figure 3.1 Schematic of workflow in this study

3.2 Data collection and feature engineering

3.2.1 Download satellite images from Google Earth Engine

Download 98 stacked satellite images of Sentinel-2 Level-2A in January – May 2020 along the Mun river from Google Earth Engine (Figure 3.2). Use the images in this period because it's summer season which doesn't have temporary water body and large clouds (Figure 3.3), this makes the classification process of K-means less complicated.

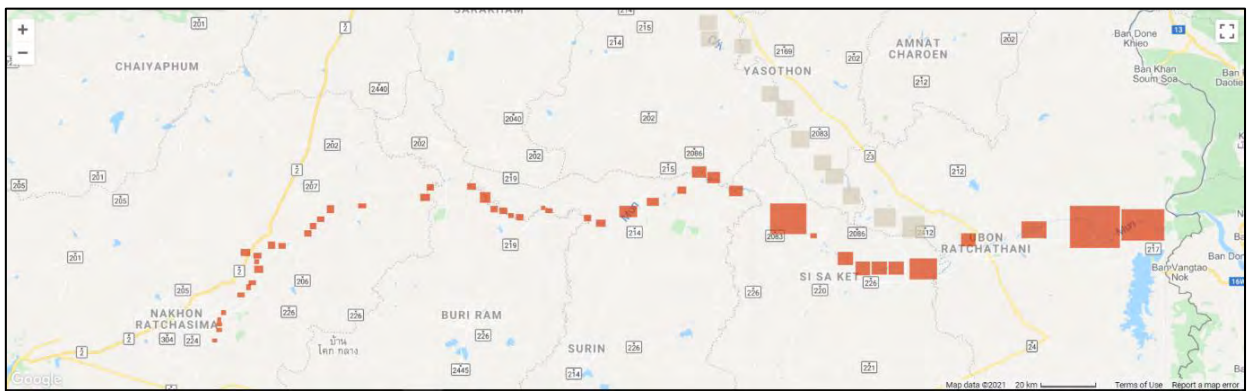


Figure 3.2 Study area in Google Earth Engine (GEE) with downloaded polygon

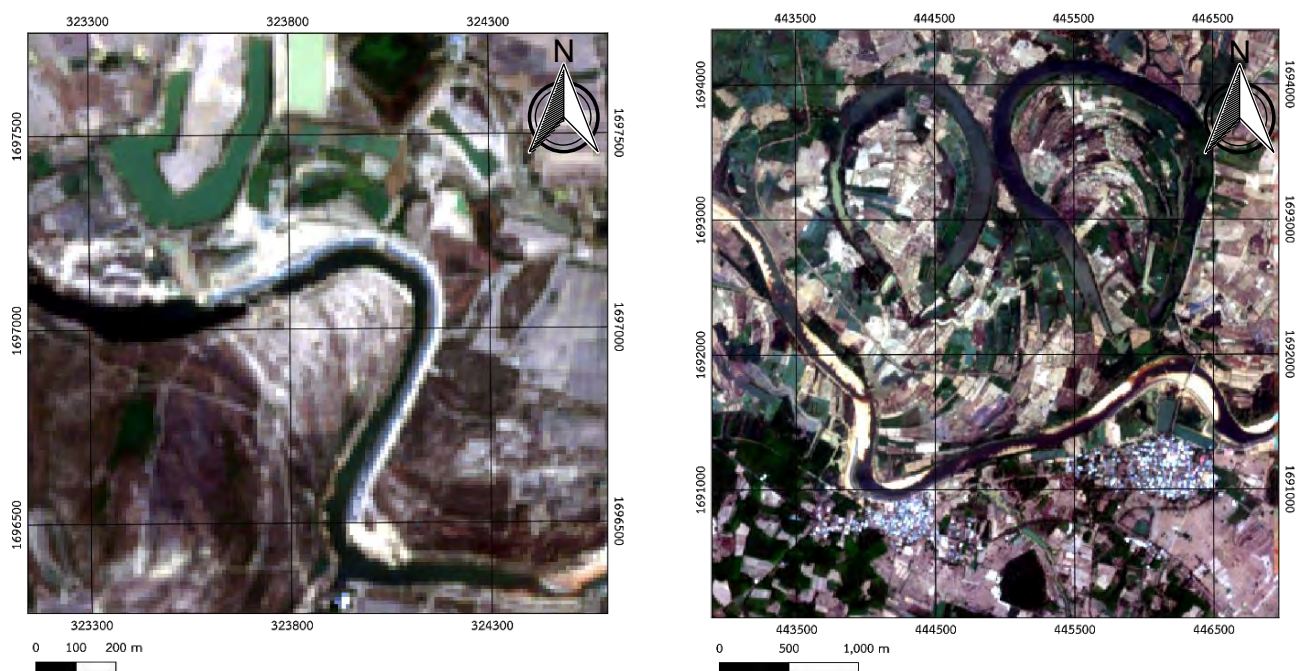


Figure 3.3 Two samples of band432 of downloaded images (left 061.tiff, right 077.tiff)

3.2.2 Using K-means clustering to classify water body

The classifying process of K-means clustering was previously mentioned in Chapter 2. This study uses K-means clustering from Scikit-learn to classify objects in the satellite images into two classes: water and land. In the classified process K-means will pull each band of the satellite images and give the result into the classified image. Then, the result of band 8 is the best band to classify water out of land because the difference of reflectance between water and land is high (Figure 3.4) in the wavelength of band 8. Therefore, in this project is going to use band 8 of the Sentinel-2 to implement in the next steps because band 8 of Sentinel-2 has a wavelength of 835.1 nm, which can classify water out of soil and vegetation because water does not reflect in this wavelength. But soil and vegetation have high reflectance at this length (Figure 3.5 and 3.6). Thus, K-means can identify water bodies easily (Figure 3.8).

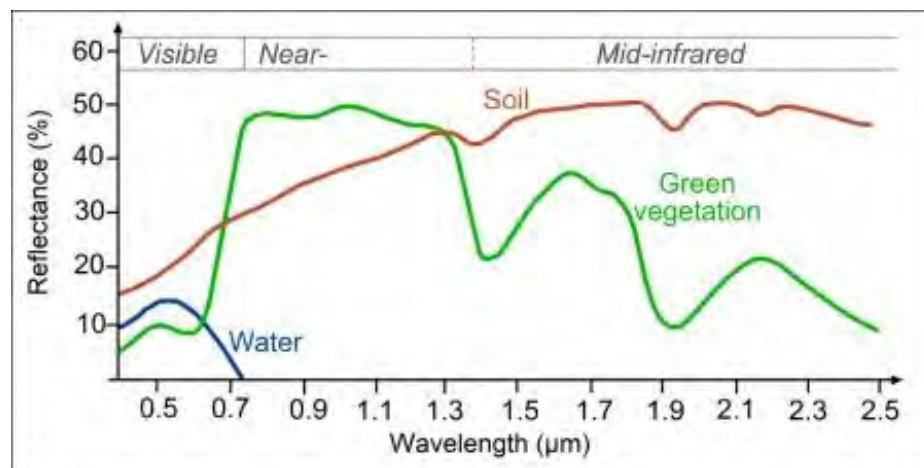


Figure 3.4 Reflectance curve of water (blue line), Vegetation (green line) and Soil (red line) along the wavelength in range 0-2.5 micrometer (Serco Italia SPA, 2017)

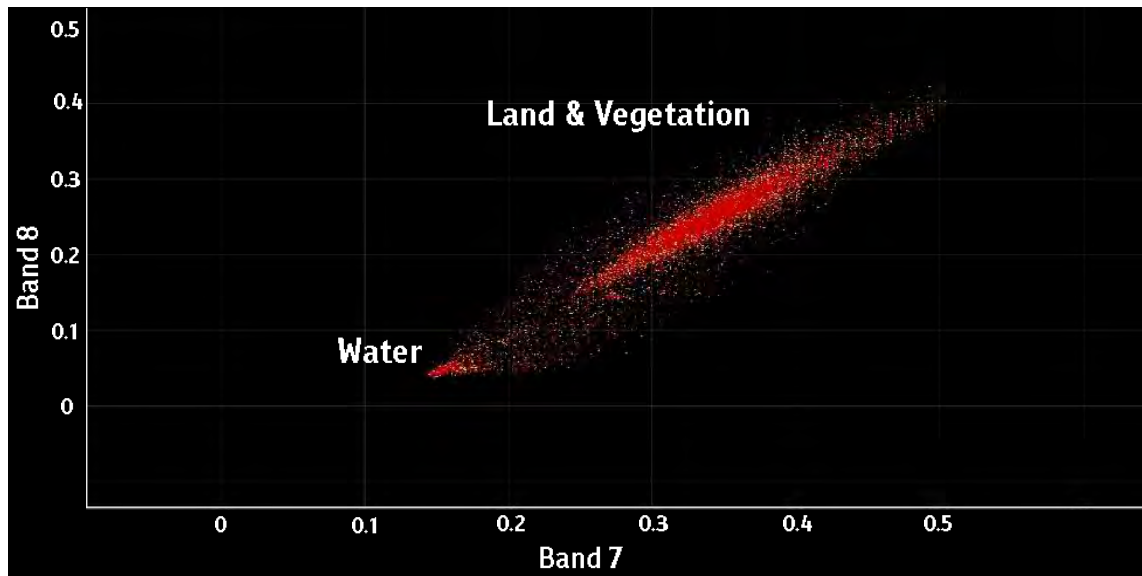


Figure 3.5 Comparing plot between Band 7 and Band 8 of image 066.tiff

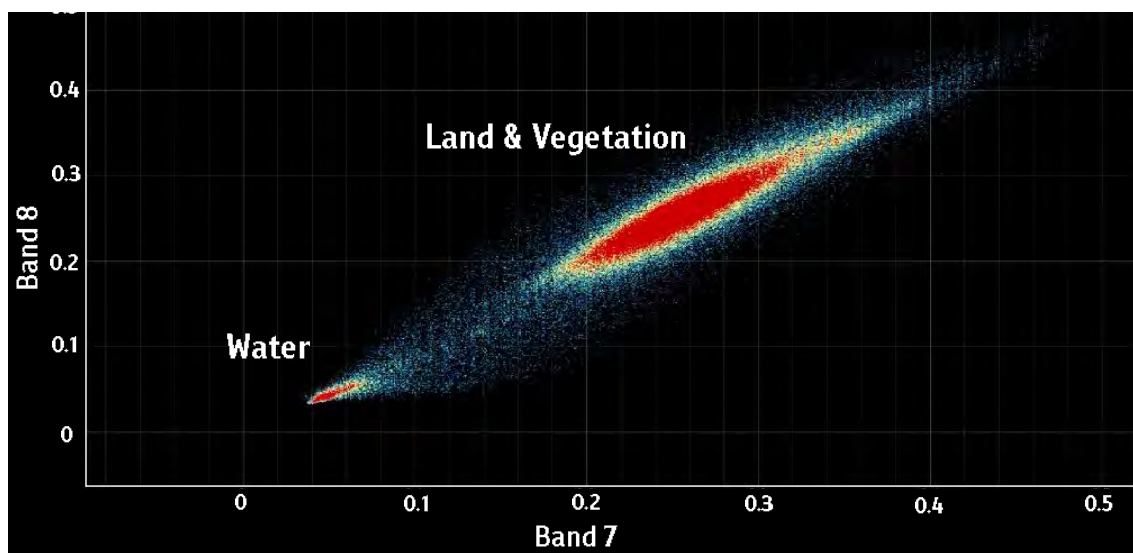


Figure 3.6 Comparing plot between Band 7 and Band 8 of image 071.tiff

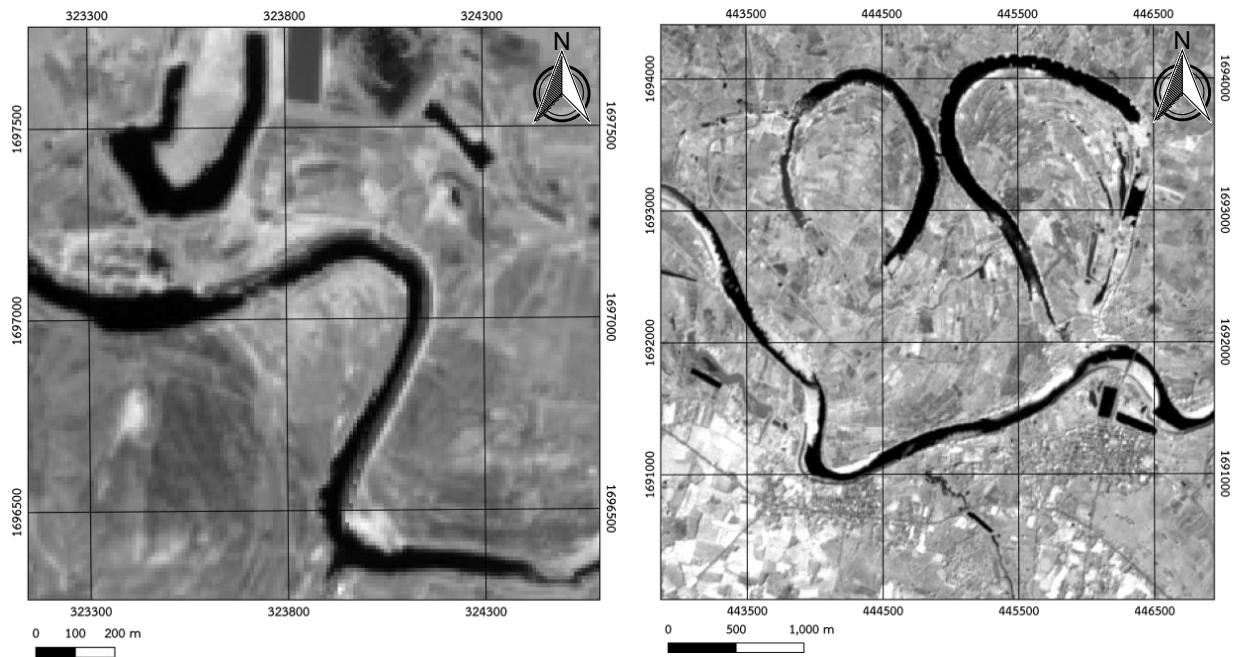


Figure 3.7 Two samples of band 8 of downloaded images

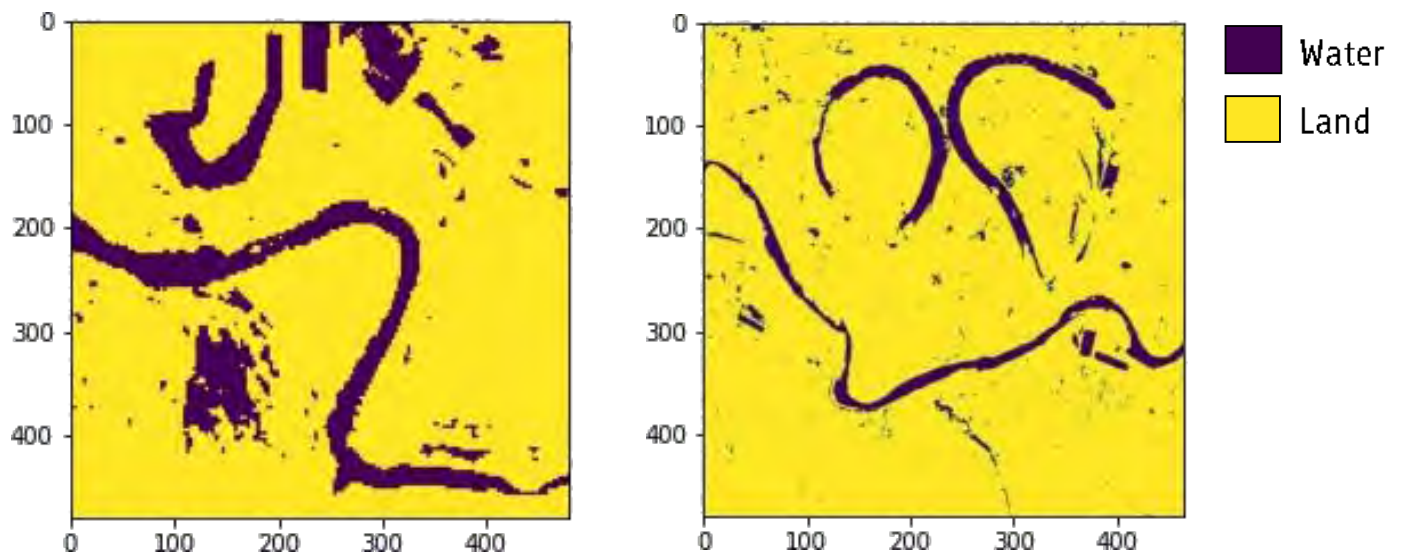


Figure 3.8 Two samples of classified images which is classified into two classes:
 1. Water (violet color), 2. Land & vegetation (yellow color)

3.3 Labeling

Labeling is drawing the bounding box (green rectangle in Figure 3.9) around the wanted class in the images. Labeling is the required method in object detection task that machine will learn and recognize the labeled object in the images in training process. If we have the more labeled object, the more accurate the model can be.

In this study uses opensource labeling tool named Labelling from Tzutalin, 2015 in Github to label oxbow lake and main channels in the classified images (Figure 3.9). After labeling of all 98 dataset images, the amount of Main channel class equal 102 objects and Oxbow lake equal 160 objects (Figure 3.10).

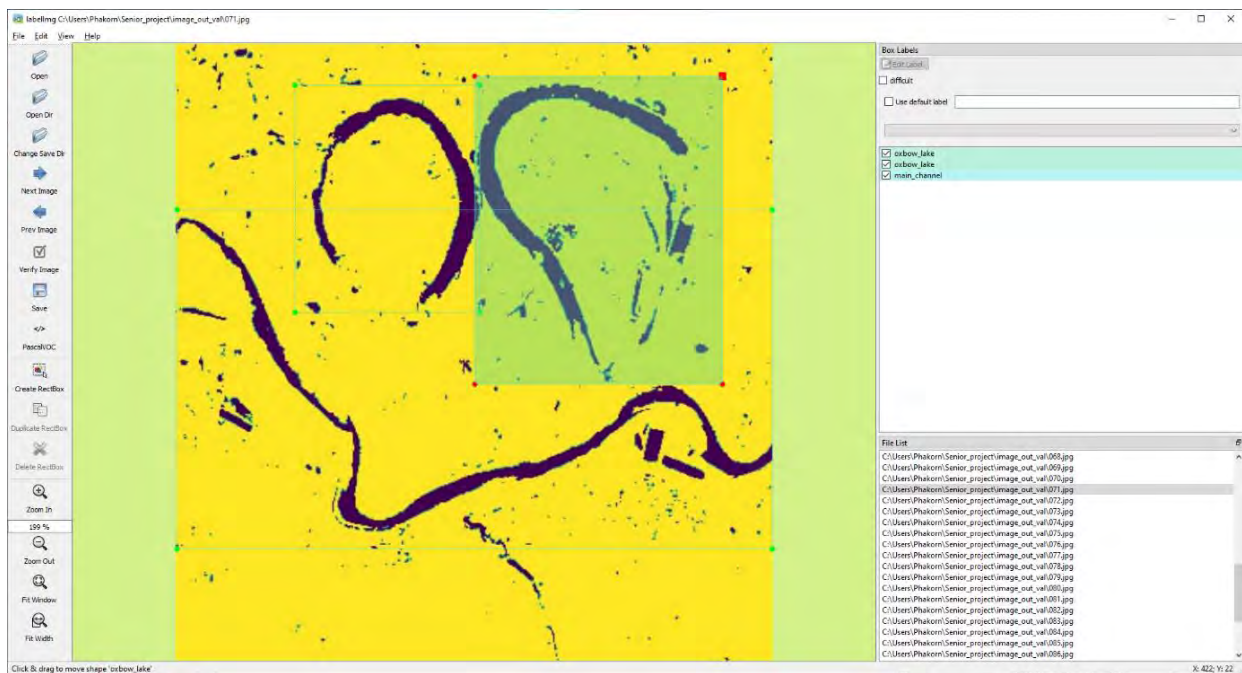


Figure 3.9 Labelling tool from Labeling from Tzutalin, 2015

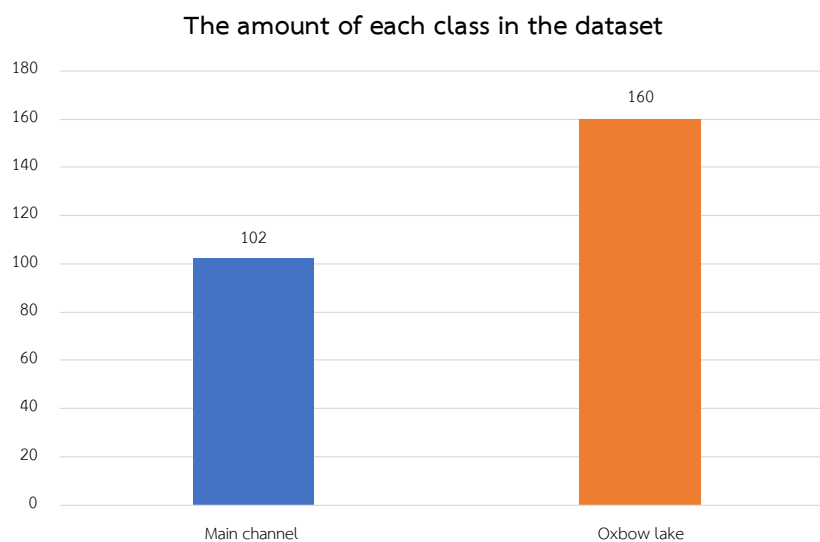


Figure 3.10 Illustrate of the amount of labeled object in each classes.

3.4 Building neuron network architecture

This study use MMDetection (Chen et al., 2019), which is open-source object detection toolbox based on PyTorch, to build and train Faster R-CNN ResNet-50 and RetinaNet50. PyTorch is an open-source machine learning library based on the Torch library, mainly developed by Facebook's AI Research lab for applications such as object detection and others computer vision.

Write python code to use the MMDetection API to retrieve both models from Pytorch and save it to a local host (Appendix A). The pipeline of the input images is passed through the model (Figure 3.11 and Figure 3.12), and the model learns and identifies the object in the image before computing and predicting the objects.

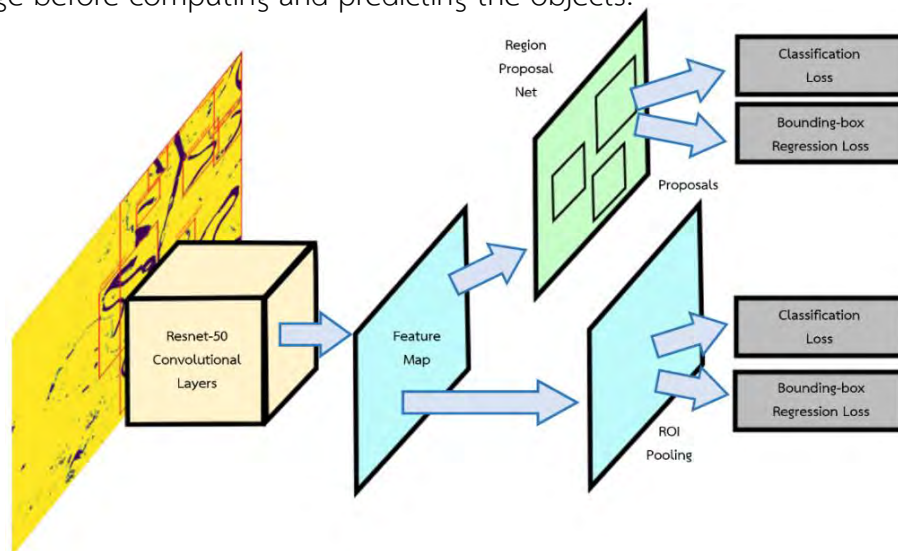


Figure 3.11 Illustrate of Faster R-CNN architecture. (Modify from Mohan and Vinayakumar et al., 2019)

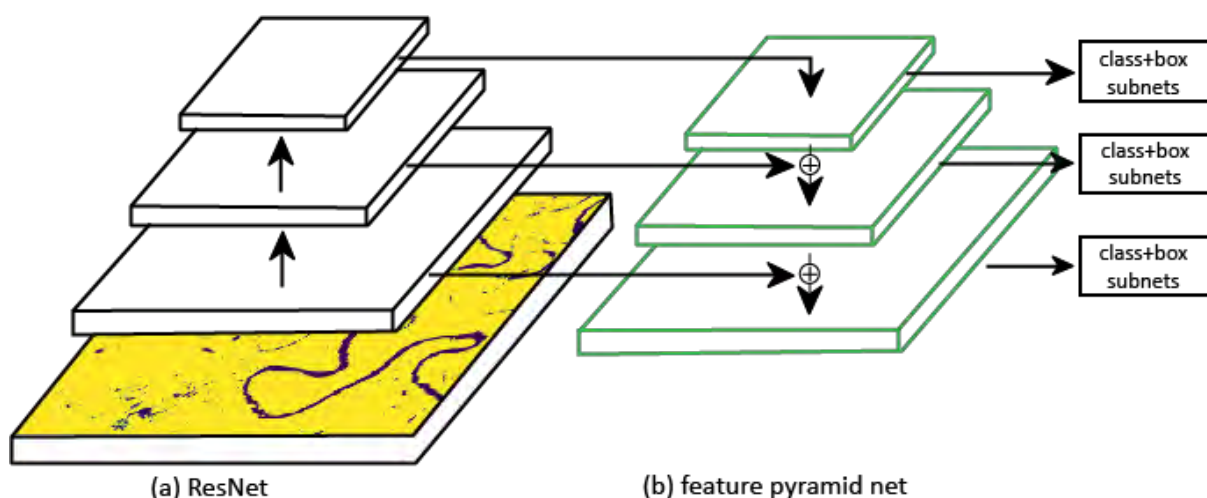


Figure 3.12 Illustrate of Retinanet architecture. (Modify from Zeng, 2019)

3.5 Training models

This process will separate dataset from 98 images into two part, 80% for Training set (78 images) 2. 20% for Validation set (20 images). Training set is used for the training process but Validation set is used for evaluate model after finished the training in each epoch.

The training model is a process that machine extracts and learns the features of the objects in the input images (Figure 3.13). The detail of the learning process of Faster R-CNN ResNet-50 and RetinaNet50 was previous mentioned in Chapter 2.

The training model has many algorithms to optimize weight parameters in the neural network such as Stochastic Gradient Descent (SGD), Adaptive Moment Estimation (Adam). In this study uses SGD because of research from Wilson et al., 2018 which has an experiment to compare training performance among SGD, Adam, RMSProp, and AdaGrad. The result of the research show SGD has better performance than the others in both training and testing process.

The SGD has three significant parameters:

- 1. Learning rate (lr)** controls how fast the optimizer decreases or increases the weight parameter in the neural network.
- 2. Weight decay (wd)** controls decaying of weight in each training epoch. If using appropriate value, the model can avoid the overfitting problem.
- 3. Momentum** is a parameter that plus into the gradient term in stochastic gradient descent to avoid machine get stuck on the local minima (Figure 3.14). Therefore, the Momentum is a parameter that improves both training speed and accuracy of the model.

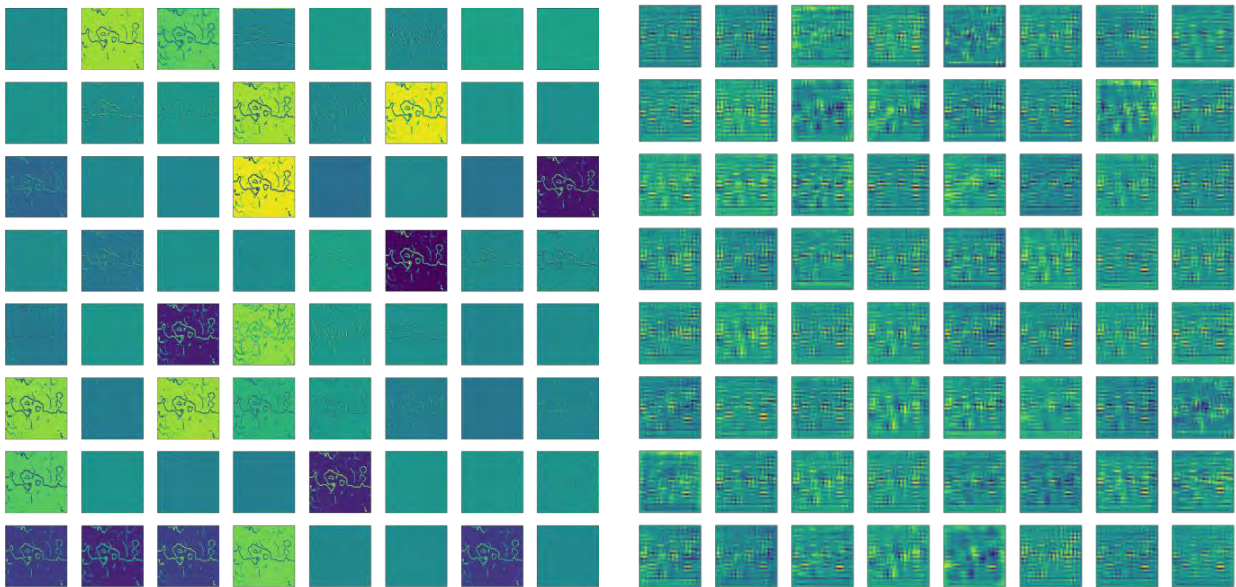


Figure 3.13 Result of features map in layer 1 (left) and layer 49 (right) when pass the classified image into Resnet-50

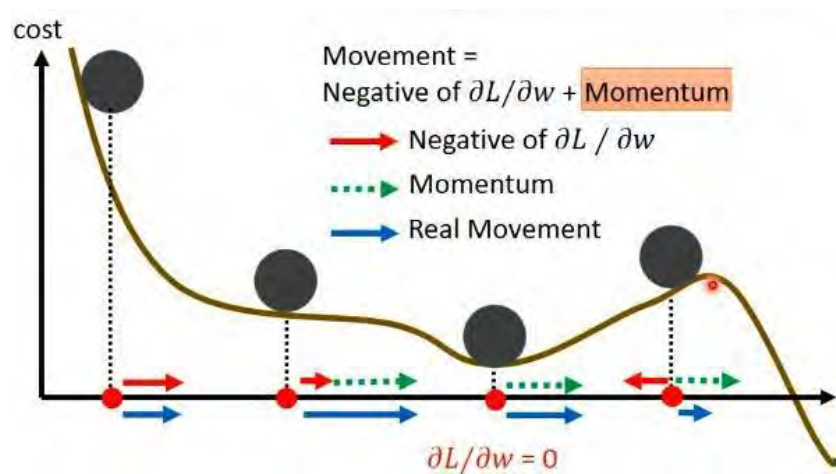


Figure 3.14 Functioning of Momentum parameter is helped the gradient of SGD to skip the local minima. (Dong, 2019)

After the end of the learning feature in Resnet-50 Convolutional Layers, the machine creates bounding boxes whole the image (Figure 3.15), then use the IoU method (Explained in Chapter 2) to filter the usable boxes which have $\text{IoU} > 0.5$. The usable boxes is predicted what is the class in there by using the Softmax function with weight parameter in each training Epoch.

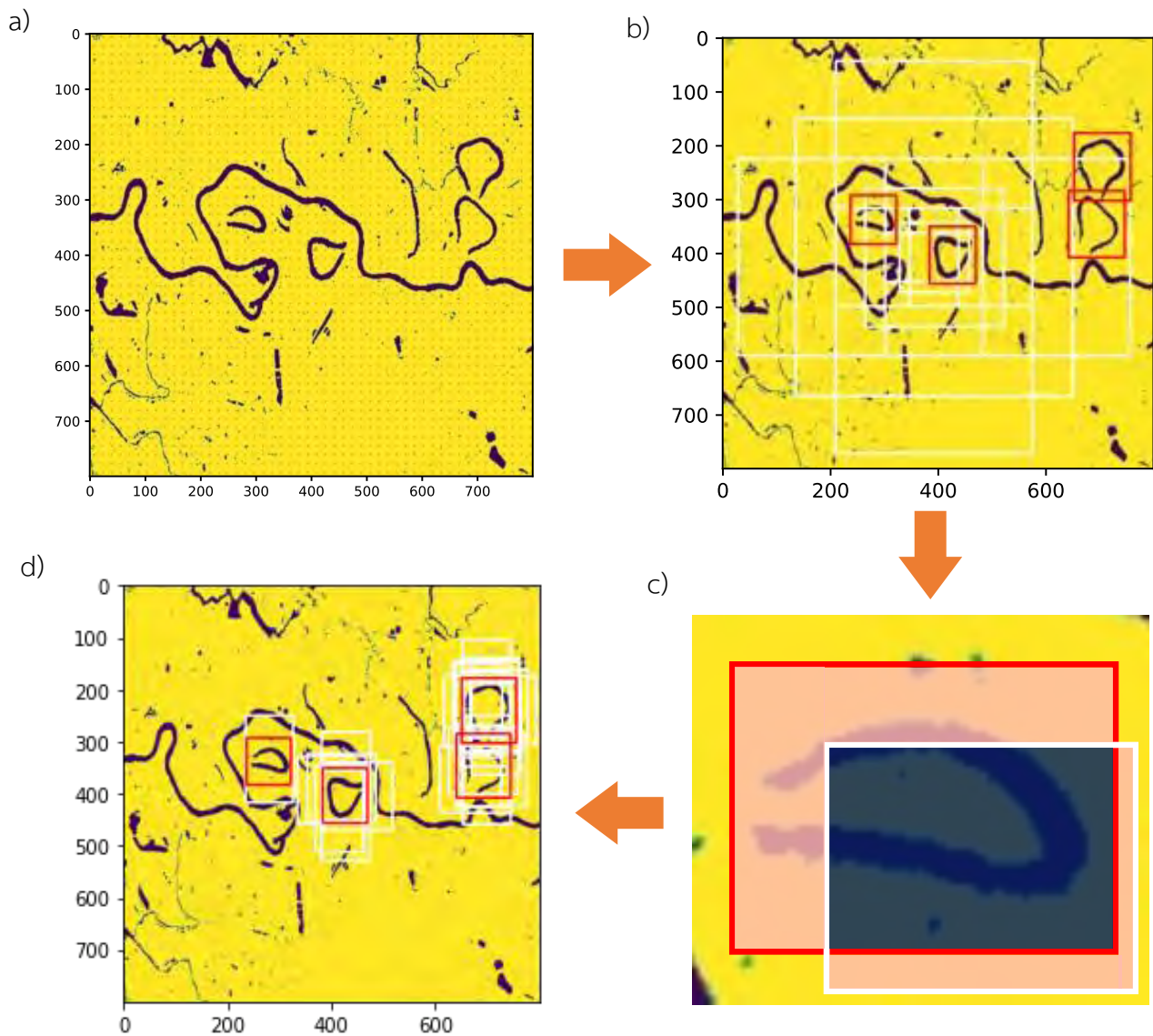


Figure 3.15 Illustrate of Region Proposal which is process of creating bounding boxes in object detection task. a) Create anchors whole of the image. b) Create bounding boxes from anchors. c) Filter the usable boxes by using IoU d) Filtered bounding boxes

Chapter4

Result and Discussion

4.1 Evaluate models

From several tests, we found that the best SGD's parameters yielded low error (CrossEntropy, L1) of the models are shown in table 3.

Table 3 The best SGD's parameters of each models

Model	Learning rate	Weight decay	Momentum
Faster R-CNN Resnet50	0.0025	0.0001	0.9
RetinaNet50	0.003	0.0001	0.9

Evaluation from the training set gets loss of class (CrossEntropy) and loss of bounding box (SmoothL1). The result of the two models is shown in Figure 4.1 and Figure 4.2. Loss of RetinaNet50 is higher than Faster R-CNN ResNet-50 in Early training Epoch but the end of training overall loss of RetinaNet50 is lower than Faster R-CNN ResNet-50. CrossEntropy and L1 are the loss function which is used for measure error from prediction of model. CrossEntropy (Equation 3) is an equation that calculate error of probability of predicted class compare with labeled class in softmax activation function. L1 (Equation 4) is an equation that calculate error of predicted bounding boxes compare with labeled boxes.

$$\text{Cross Entropy (i)} = -k \log \left(\frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \right) - (1 - k) \sum_{j \neq i} \log \left(\frac{e^{z_j}}{\sum_{j=1}^K e^{z_j}} \right) \quad (\text{Equation 3})$$

Where, k = number of classes

K = total columns of matrix in output layer

i,j = refer to row and column of matrix

Z = value of row i, column j in matrix

$$\text{L1 loss} = \text{MAE}(P, L) \quad (\text{Equation 4})$$

Where, MAE = Mean absolute error $(\frac{1}{n} \sum_{i=1}^n |P_i - L_i|)$

P = coordinate of predicted bounding box (xmin, ymin, xmax, ymax)

L = coordinate of labeled bounding box (xmin, ymin, xmax, ymax)

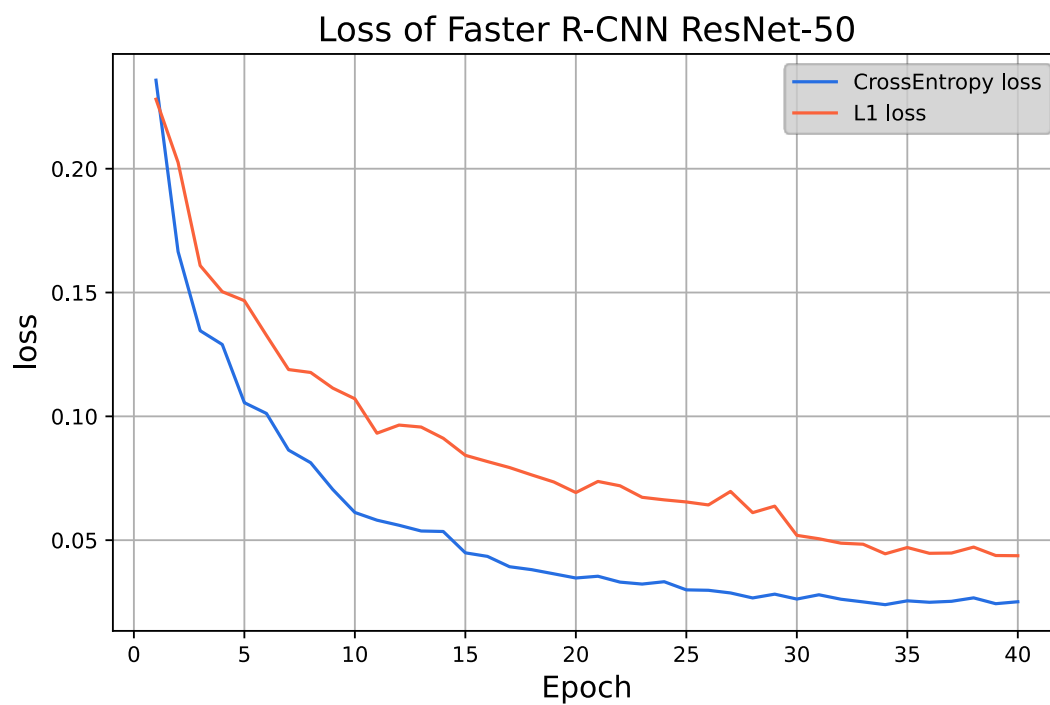


Figure 4.1 Illustrate of learning curve in training process of Faster R-CNN ResNet-50. The model gets loss of class around 0.02 and loss of bounding box around 0.05

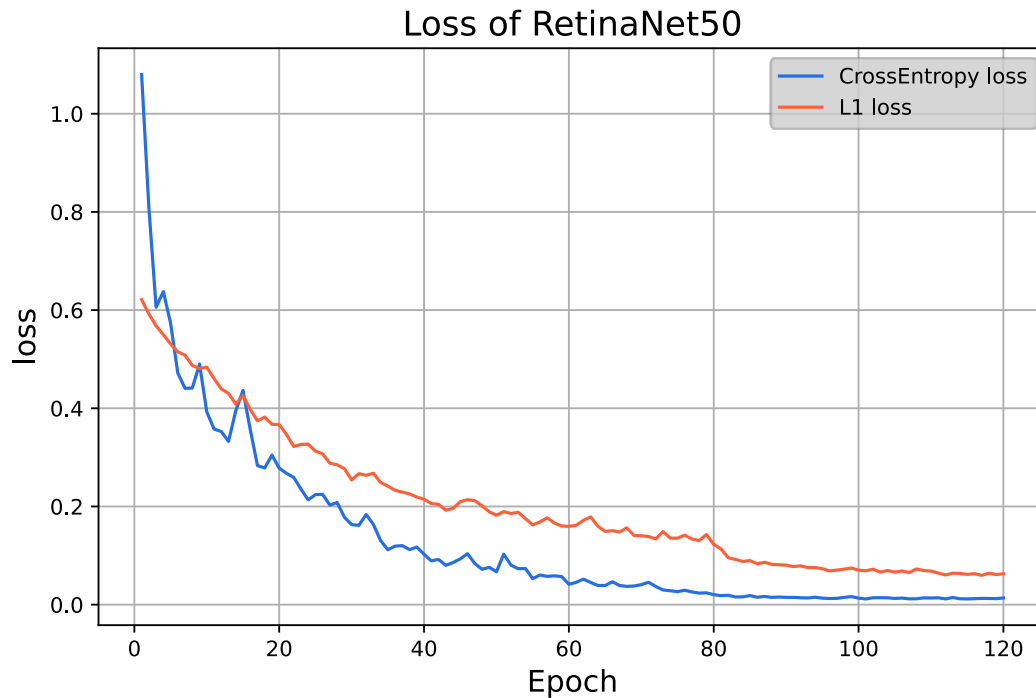


Figure 4.2 Illustrate of learning curve in training process of Faster R-CNN ResNet-50. The model gets loss of class around 0.01 and loss of bounding box around 0.06

Next, evaluate the models from the validation set. Even though the loss of Faster R-CNN ResNet-50 in the training set is lower than RetinaNet50 but in the validation set Faster R-CNN ResNet-50 has high value of mAP than RetinaNet50. The model can reach a value of mAP to 0.78 (Figure 4.3) compare with RetinaNet50 is 0.67. But the RetinaNet50 uses 120 Epoch to reach the highest mAP that has a long training time than Faster R-CNN ResNet-50.

Mean Average Precision (mAP) is a value which is summation of average precision (AP) of each class in the validation set (Equation 5).

$$mAP = \sum_{k=1}^{k=n} AP_k \quad (\text{Equation 5})$$

Where, AP_k = Average precision of class k

n = the number of classes

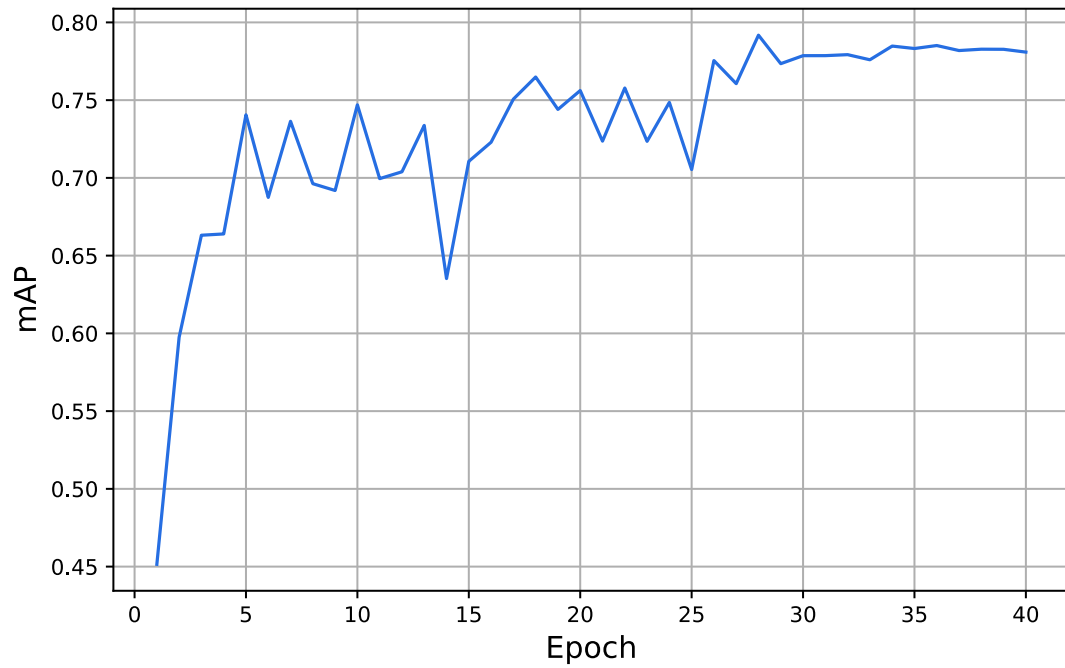


Figure 4.4 Illustrate of learning curve in training process of Faster R-CNN ResNet-50. The model gets mAP value around 0.78

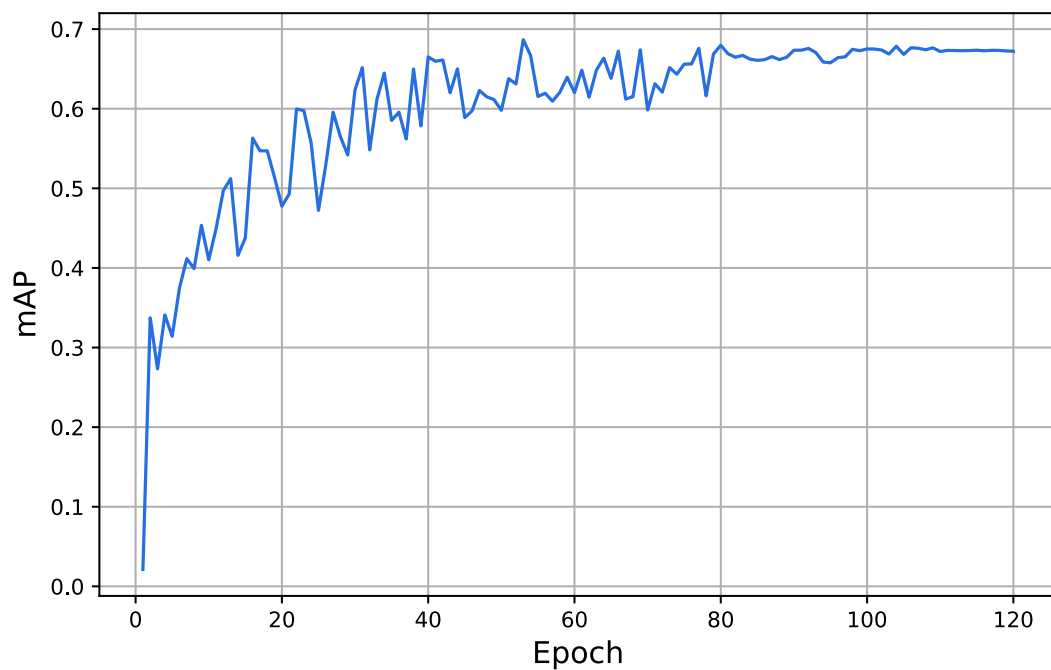


Figure 4.3 Illustrate of learning curve in training process of RetinaNet50. The model gets mAP value around 0.67

4.2 Testing models

After evaluating the models from the value of loss and mAP, this study uses a testing process imported from other images. We do not include the images in the dataset. The separation can avoid bias from training and validation models, which will lead to the proficiency of inference and predicting models. The result shows in Figure 4.5 and Figure 4.6. For Figure 4.5, both models can completely detect all objects (one oxbow lake and one main channel) with high probability. On the other hand, in Figure 4.6 shows the difference of both models clearly. Faster R-CNN ResNet50 is more accurate than RetinaNet50 because the model can detect all objects (4 oxbow lakes and 1 main channel) in the test image with high probability. Different from Retinanet50, none of all the objects can't be detected (2 oxbow lakes and 1 main channel). Moreover, the detected position of bounding boxes of Faster R-CNN ResNet-50 is also greater than RetinaNet50.

Therefore, Faster R-CNN ResNet-50 is selected to be the base model for implement the last process that is evaluate water surface areas in the detected objects.

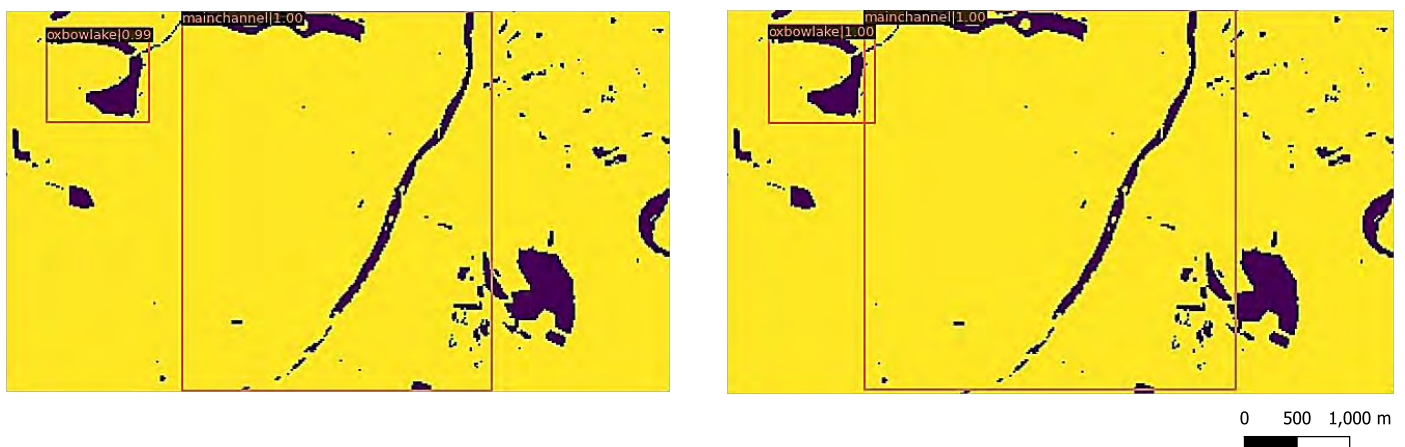


Figure 4.5 test image number test001.jpg. Result of prediction from Faster R-CNN Resnet50 (Left), and RetinaNet50 (Right).

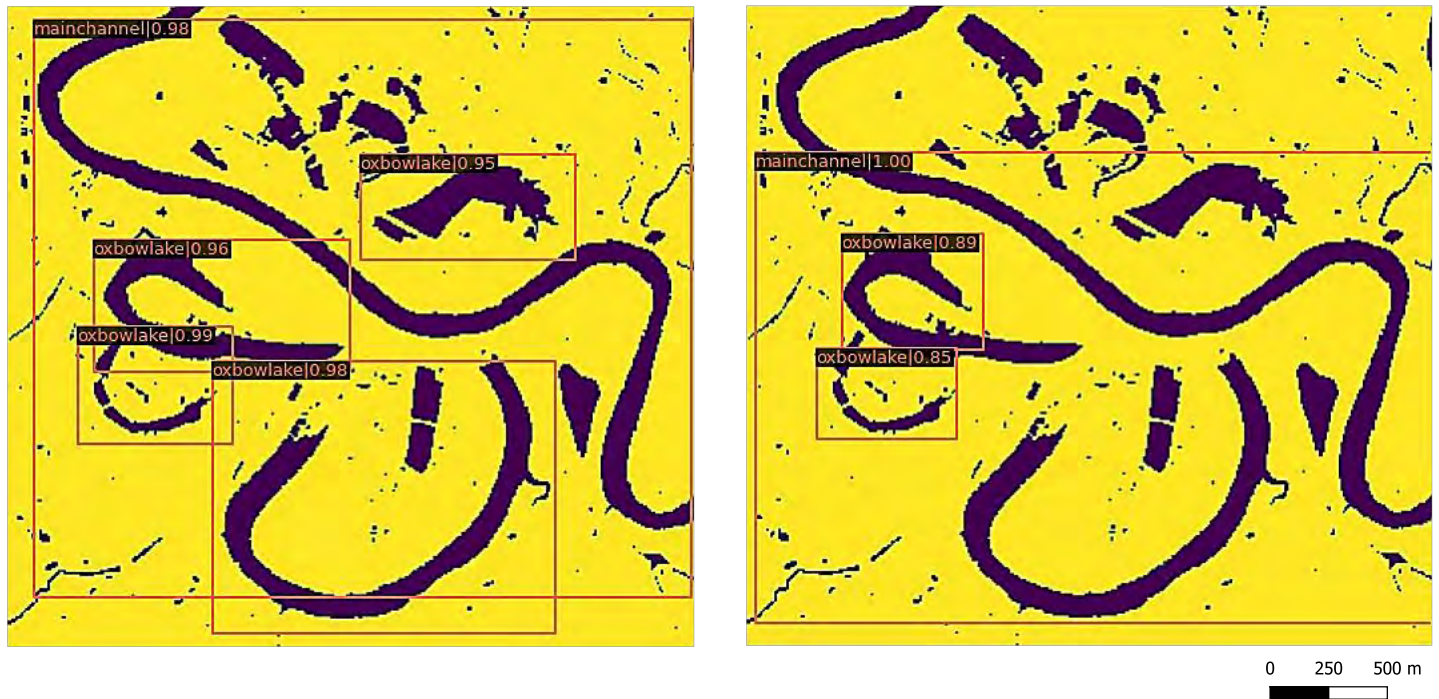


Figure 4.6 test image number test002.jpg. Result of prediction from Faster R-CNN ResNet50 (Left) is more accurate than RetinaNet50 (Right) because 4 oxbow lakes and 1 main channel in the image can be completely detected by Faster R-CNN ResNet-50. While Retinanet50 can detect just 2 oxbow lakes. Moreover, the precision of creating bounding. boxes of Faster R-CNN ResNet-50 is higher than Retinanet50.

4.3 The evaluation of water surface areas

Before evaluate water surface areas in each detected object, this study uses a coordinate (xmin, ymin, xmax, ymax) of the detected bounding boxes. Then use the coordinate eachs box to crop the object in the original image.

This process uses image segmentation (Thongsang et al., 2021) to clean noise, which refers to small ponds in the detected object automatically before calculate the water surface areas in the oxbow lake and main channel. Cleaning noises is using connectivity of pixel, if the amount of connected water pixels is low those pixels value will be turn into zero. The result of cleaning is shown in Figure 4.7- Figure 4.13

Calculating water surface areas is using count amount of water pixels (Purple color) after cleaning process. Then bring the amount of the pixels multiply by band 8 resolution (100 m^2) (Equation 5)

$$\text{water surface} = \text{water pixels} \times 10^2 \quad (\text{Equation 6})$$

Notice of the cleaning noises in the image. When the detected bounding box has more than one object in the box (Figure 4.12) like two oxbow lake in one box. the cleaning process can't delete the nearly oxbow lake that effect to value of water surface areas is over in calculating water surface areas process.

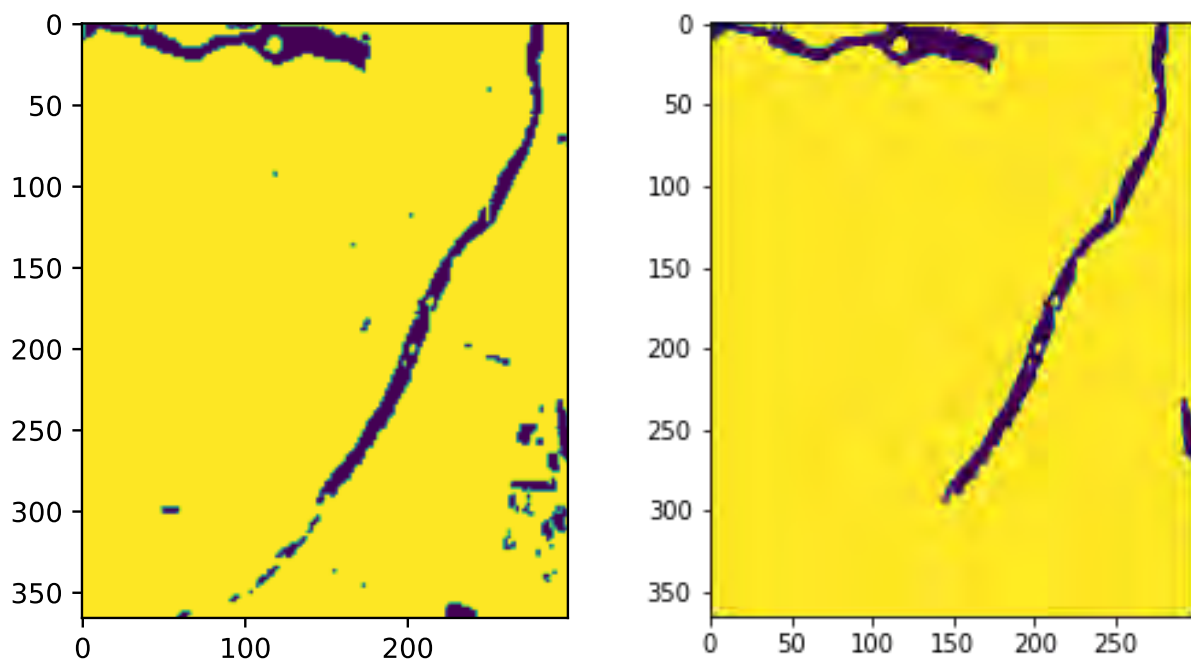


Figure 4.7 Cropped image of main channel from test001.jpg (Figure 4.5 (left)). Detected object (left) and cleaned image (right) that can be measured the water surface areas equal to 488,000 m²

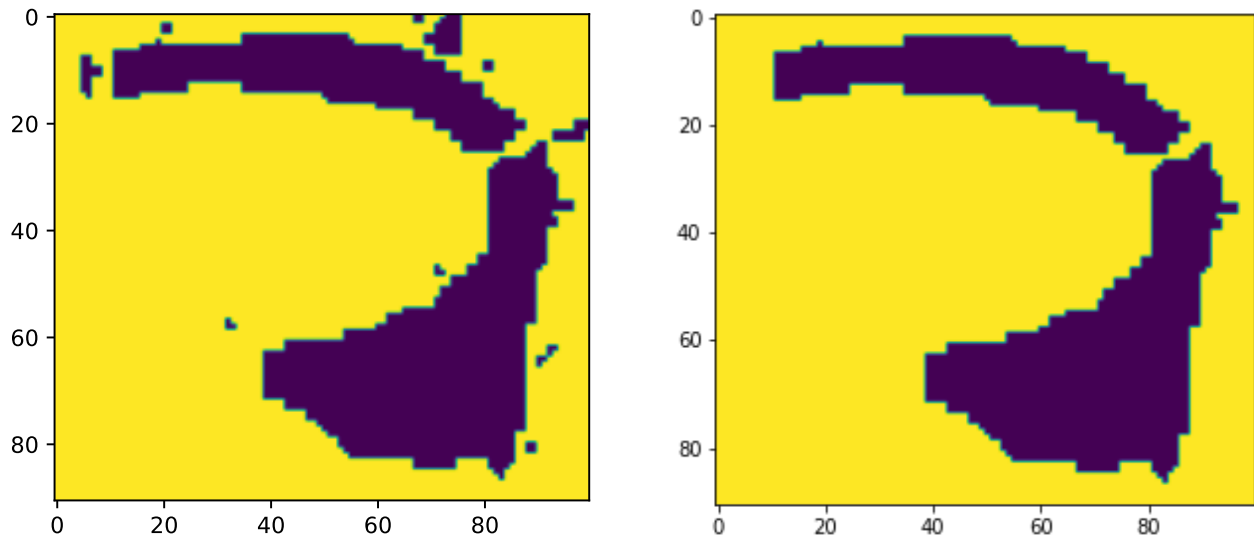


Figure 4.8 Cropped image of oxbow lake from test001.jpg (Figure 4.5 (left)). Detected object (left) and cleaned image (right) that can be measured the water surface areas equal to 231,800 m²

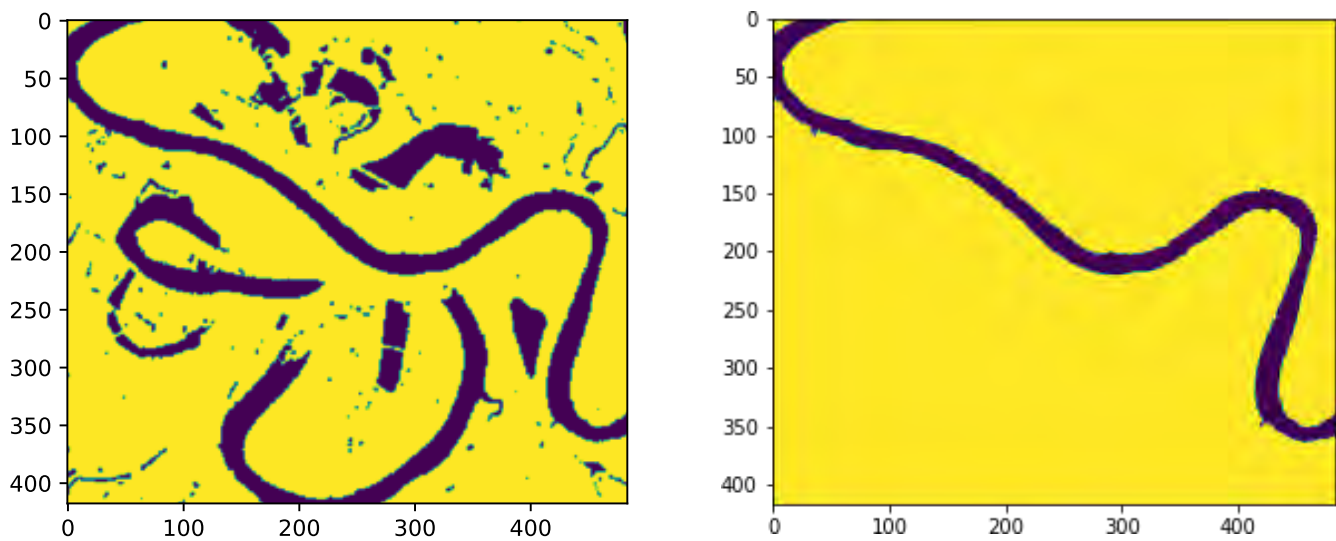


Figure 4.9 Cropped image of main channel from test002.jpg (Figure 4.6 (left)). Detected object (left) and cleaned image (right) that can be measured the water surface areas equal to 2,592,400 m²

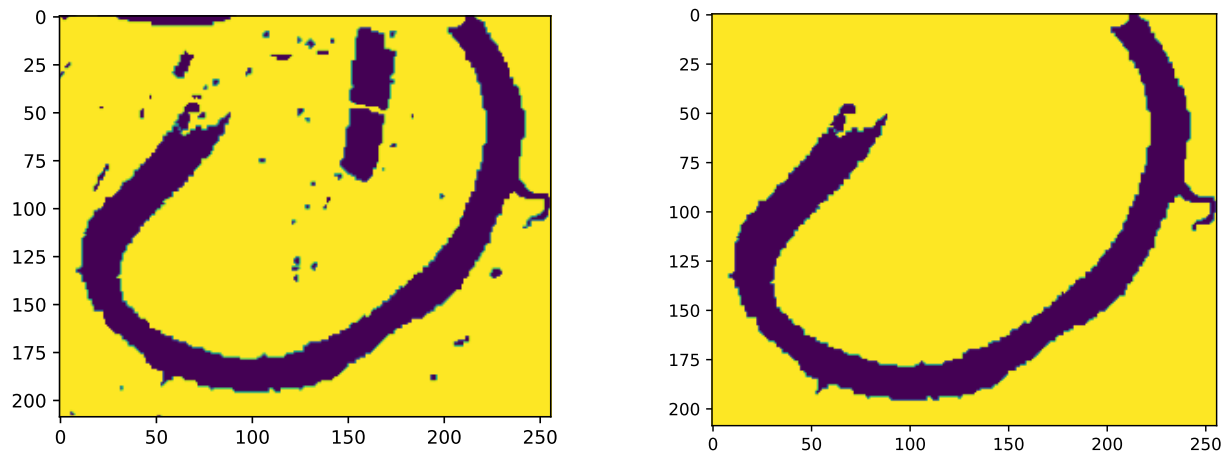


Figure 4.10 Cropped image of oxbow lake from test002.jpg (Figure 4.6 (left)). Detected object (left) and cleaned image (right) that can be measured a water surface areas equal to $877,700 \text{ m}^2$

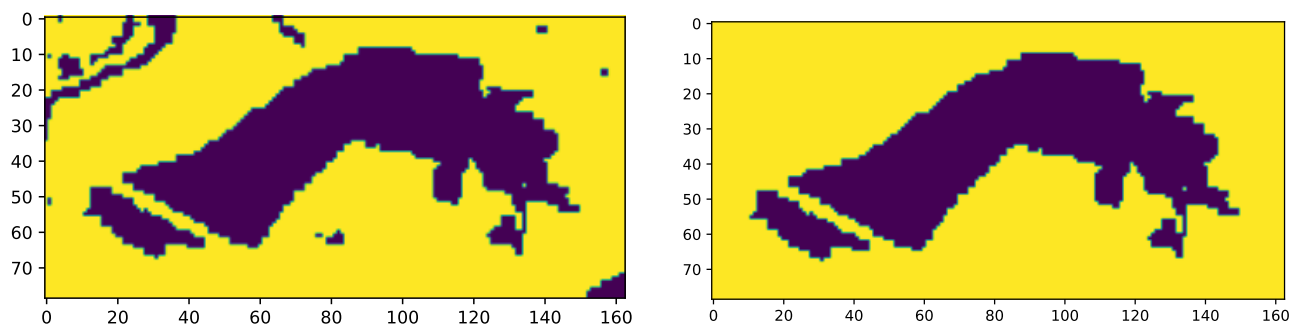


Figure 4.11 Cropped image of oxbow lake from test002.jpg (Figure 4.6 (left)). Detected object (left) and cleaned image (right) that can be measured the water surface areas equal to $375,400 \text{ m}^2$

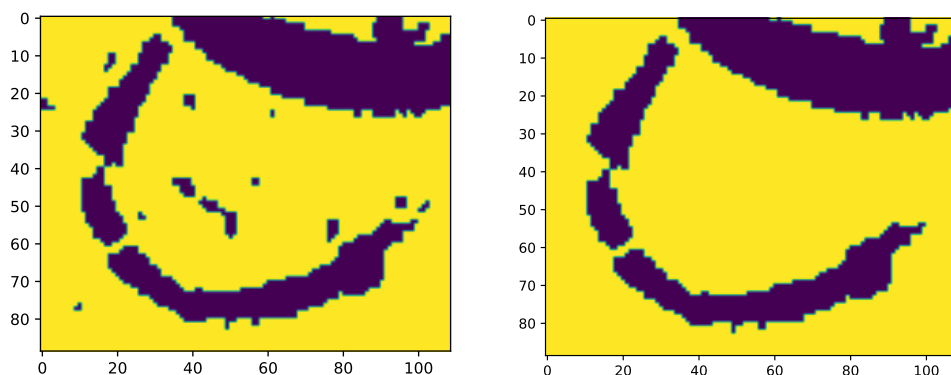


Figure 4.12 Cropped image of oxbow lake from test002.jpg (Figure 4.6 (left)) detected object (left) and cleaned image (right) that can be measured the water surface areas equal to $256,000 \text{ m}^2$

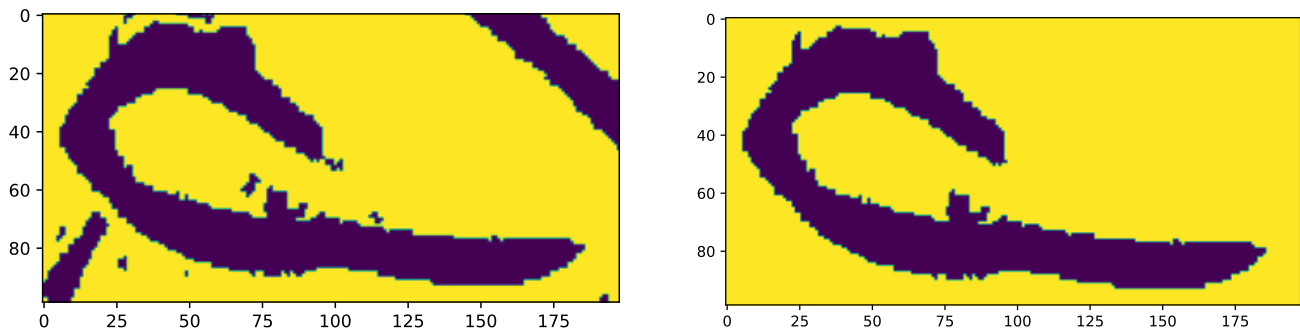


Figure 4.13 Cropped image of oxbow lake from test002.jpg (Figure 4.6 (left)) detected object (left) and cleaned image (right) that can be measured the water surface areas equal to 491,200 m²

To summarize, testing the models by using validation set, Faster R-CNN ResNet-50 has more accurate than RetinaNet50. Therefore, this study selects Faster R-CNN ResNet-50 to detect main channel and oxbow lake in the test image. Then, the detected objects is calculated water surface areas by using image segmentation and equation 6.

Chapter5

Conclusion

5.1 Model performance

Faster R-CNN ResNet-50 can extract oxbow lake and main channel in the validation set and the testing images with high performance than RetinaNet50 when compare by using value of mAP (Figure 5.1)

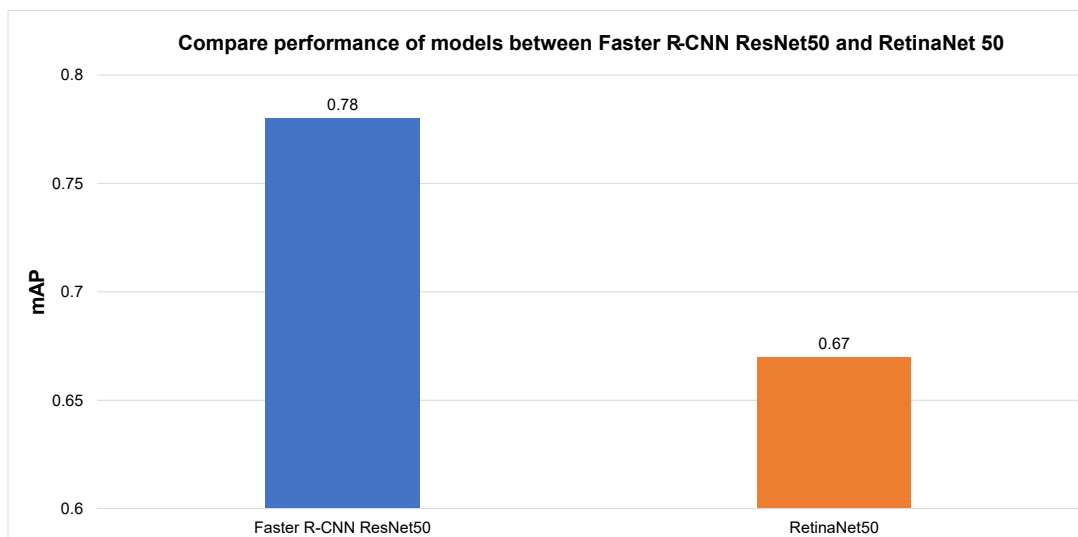


Figure 5.1 Bar plot compare the models performance by using value of mAP.

5.2 User's experience

First, author already uploads python source code to Github repository which contains tutorial python notebook that can be activated it on Google Colab and all python code for this study. Follow the link in Figure 5.2 for download source code and the tutorial notebook. Inside the tutorial notebook users can activate it in Google Colab by click the Open in Colab button (Figure 5.3)

After users opened the tutorial notebook in Google Colab already. Users have to copy the notebook into own Google Colab account (Figure5.4)

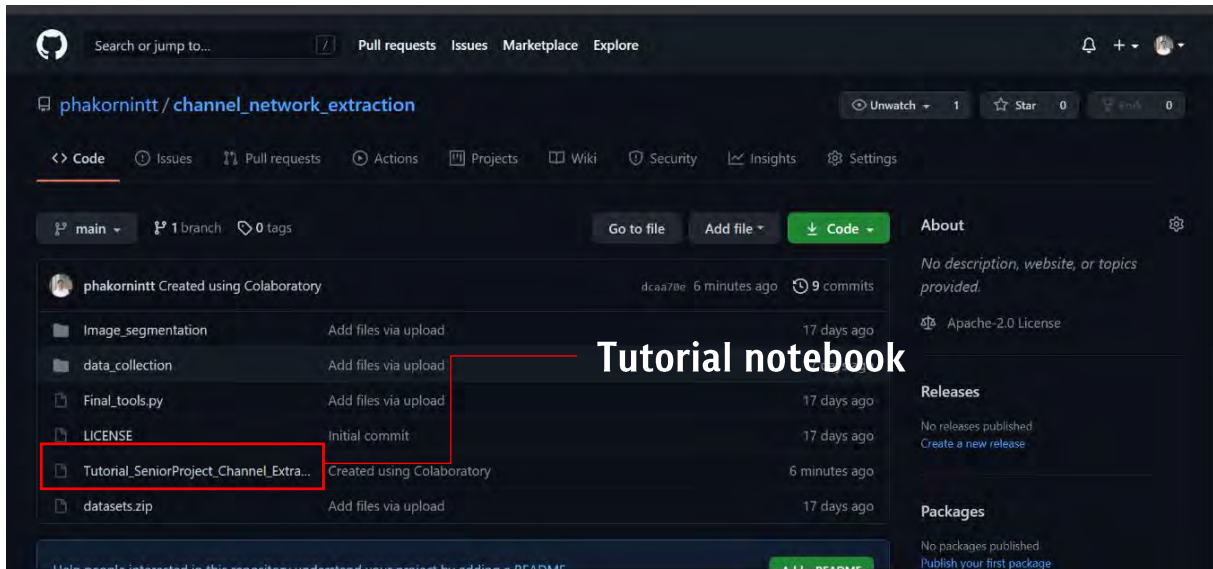


Figure 5.2 Author's Github repository for implement this study.

https://github.com/phakornintt/channel_network_extraction

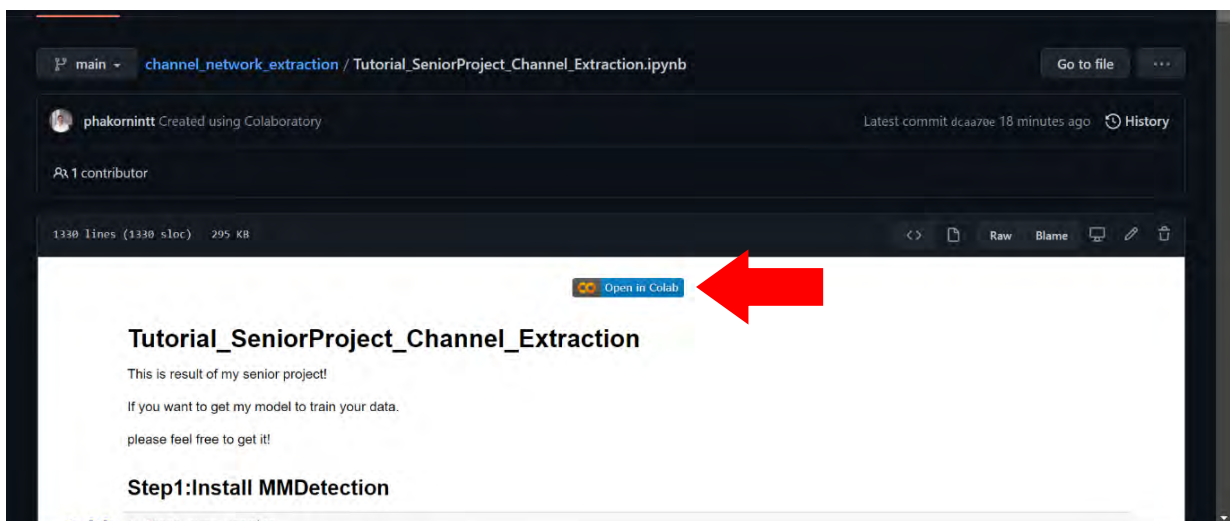


Figure 5.3 Illustrate of Tutorial notebook in Github. Author recommends new users enter the tutorial notebook in Google Colab.

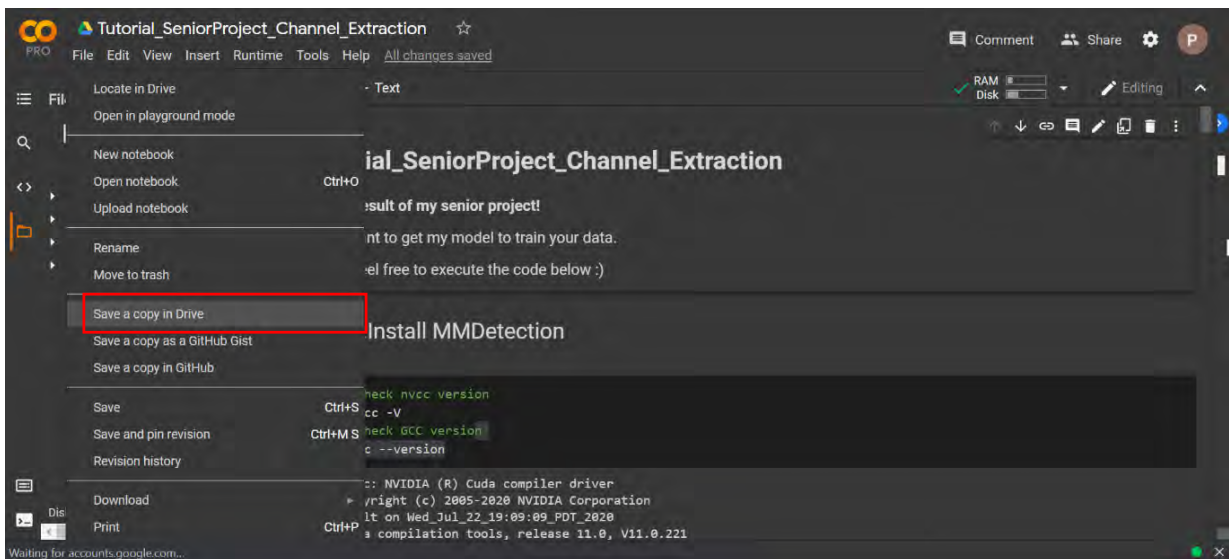


Figure 5.4 How to copy Google Colab Notebook to your own Google Colab.

Next, Users just download a satellite image (tiff file) then input into the Google Colab website by drag and drop (Figure 5.5). Then, change your filename in some line of python code in the notebook (Figure 5.6).

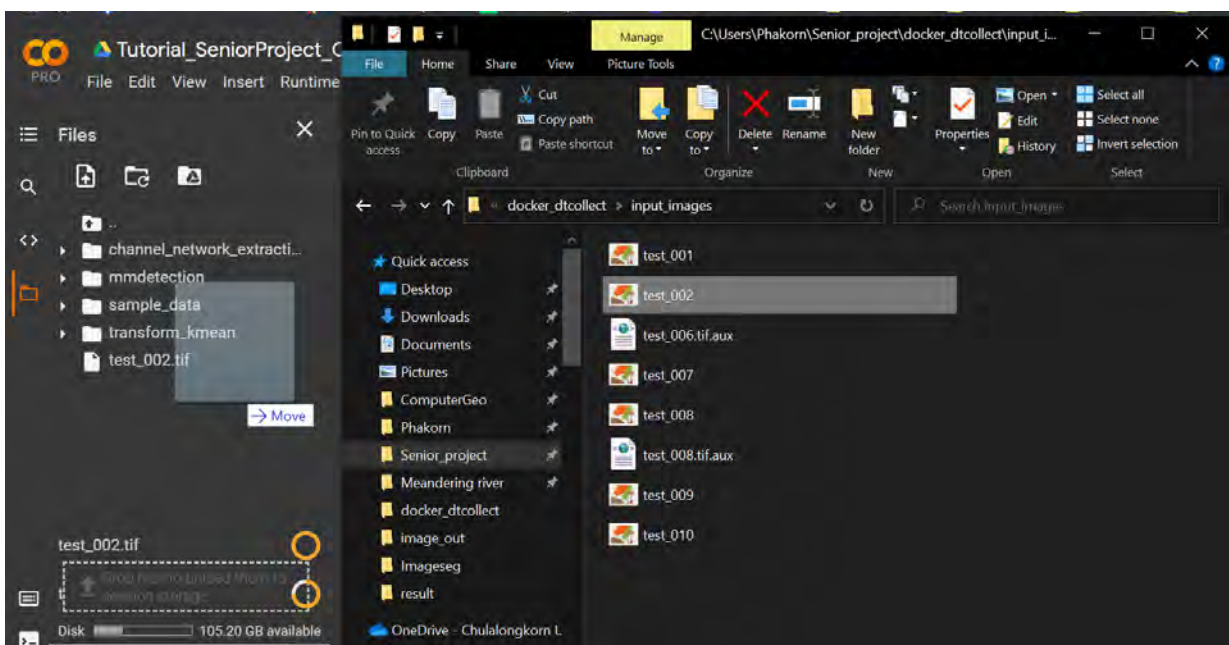


Figure 5.5 Drag and drop the input satellite image (tiff file) into Google Colab.

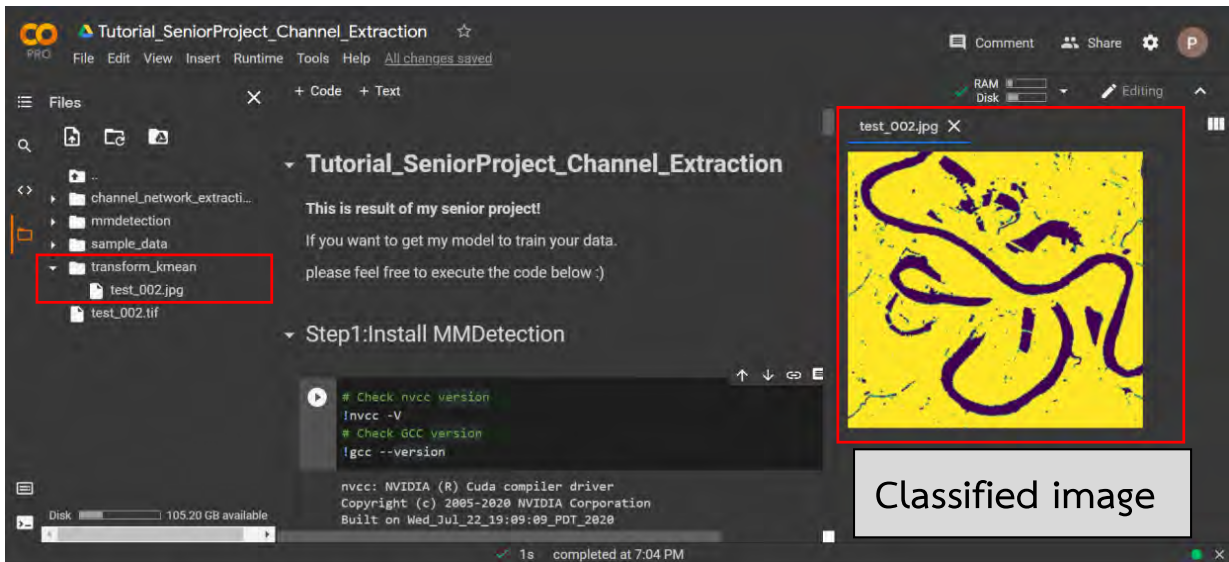


Figure 5.8 test_002.tif will be transformed to classified image (test_002.jpg) which is collected in transform_kmean folder.

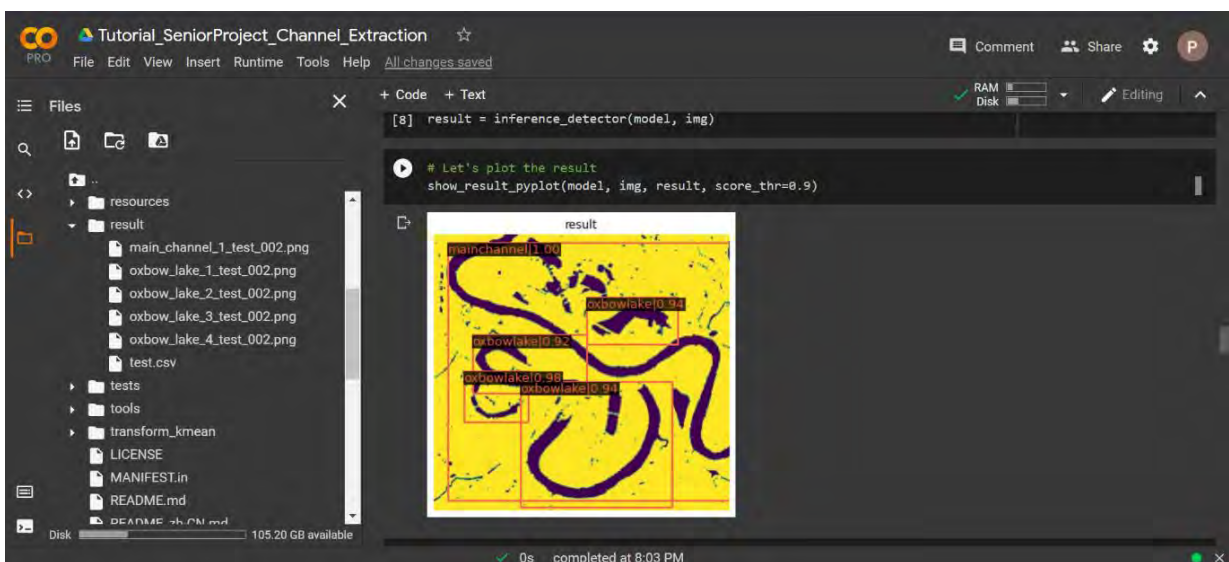


Figure 5.9 Main channel and oxbow lakes will be detected automatically.

Lastly, User will receive the cleaned images of detected objects and 1 csv file which contains object name and water surface areas column (Figure 5.10). If users want to download to the result to own local computer, users just click Download button (Figure 5.11).

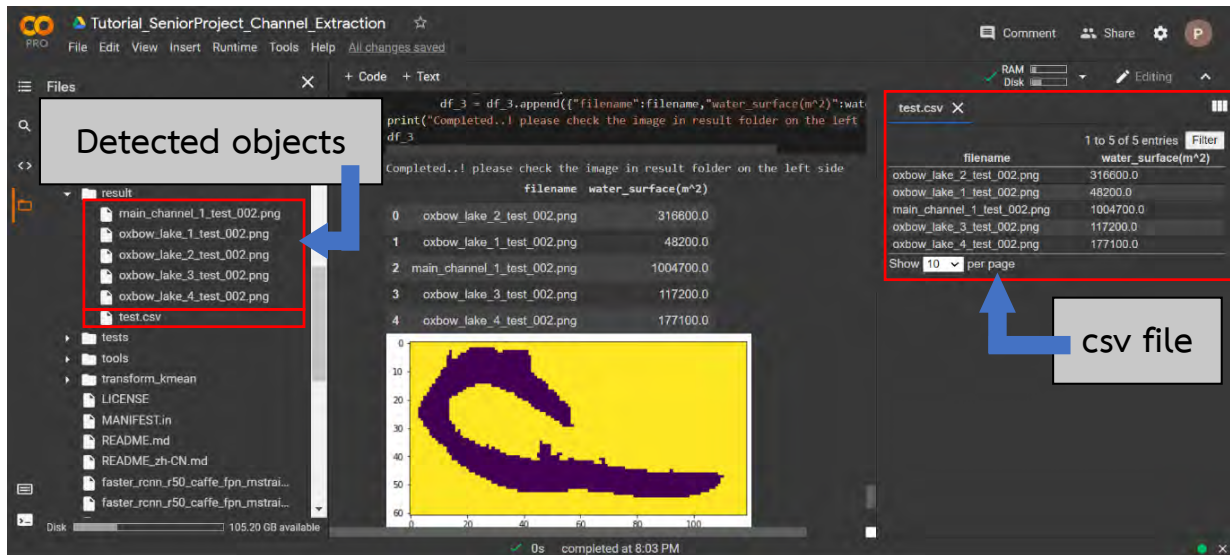


Figure 5.10 At the end of tutorial, users will receive all cleaned images of detected objects (a), and csv file (b) in result folder automatically.

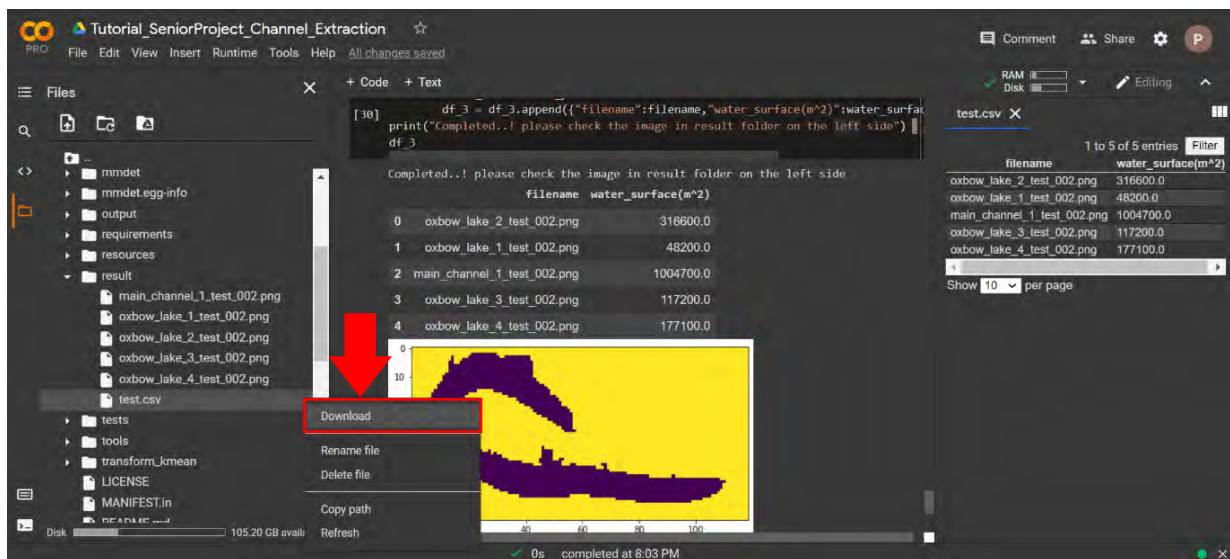


Figure 5.11 Users can download the result file into own local computer by click the Download button.

5.3 Limitation

1. This code version can't use the satellite images that have large cloud cover, low resolution (more than 10 m.) and the input image resolution is recommended in range 300 – 1000 pixels.
2. If the input images have more floodplain. The machine will lose accuracy detecting objects process.

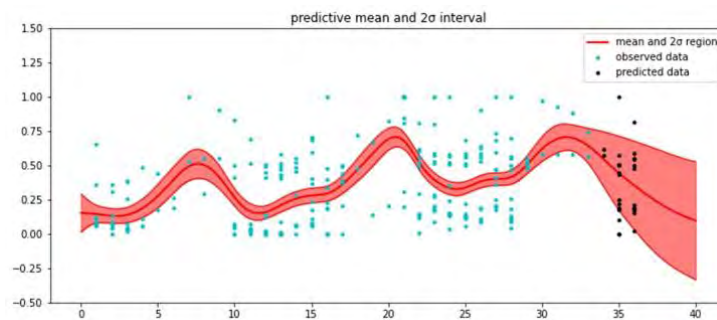
5.4 Upcoming projects



Increase dataset or other features



Create plug-in to serve the QGIS



Integrate **gaussian process regression to automated extraction and measurement of the changing water in time series**

Figure 5.12 Upcoming projects of this study:

1. Increase dataset to improve accuracy of model or add more detected features e.g. man-made ponds.
2. This tool is available especially python code. If transform code into open-source plug-in that is used for QGIS, it more accessible and friendly for the users.
3. Combine gaussian process regression to predict changing of water body in each detected objects in time series.

List of reference

- Camps-Valls, G., 2009. Machine learning in remote sensing data processing. *Proceedings of 2009 IEEE International Workshop on Machine Learning for Signal Processing*. 1-4 September 2009. Grenold, France.
- Chen, K., Wang, J., Pang, J., Cao, Y., Li, X., Sun, S., Feng, Z., Xu, J., Zhang, Z., Cheng, C., Cheng, T., Zhao, Q., Li, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C., & Lin, D., 2019, MMDetection: Open MMLab Detection Toolbox and Benchmark.
- Dertat, A., 2017. Applied Deep Learning - Part 4: Convolutional Neural Networks. [online]. Available at: <<https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2>> [Accessed 1 November 2020].
- Dong, Y., 2019. Hyper Parameter—Momentum. [online] Medium. Available at: <<https://medium.com/ai%C2%B3-theory-practice-business/hyper-parameter-momentum-dc7a7336166e>> [Accessed 12 May 2021].
- Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 580-587
- GIS Geography, 2020. Supervised and Unsupervised Classification in Remote Sensing - GIS Geography. [online] Available at: <<https://gisgeography.com/supervised-unsupervised-classification-arcgis>> [Accessed 28 October 2020].
- He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep Residual Learning for Image Recognition. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S. and Murphy, K., 2017. Speed/Accuracy Trade-Offs for Modern Convolutional Object Detectors. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

- Otuyama, J., 2000. Introduction to Convolutional Neural Networks [online] Available at: <<https://www.ime.usp.br/~otuyama/academic/cnn/index.html>> [Accessed 12 May 2021].
- Li, K., Wan, G., Cheng, G., Meng, L. and Han, J., 2020. Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS Journal of Photogrammetry and Remote Sensing*, 159, pp.296-307.
- Lin, T., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S., 2017, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2117-2125.
- Lin, T., Goyal, P., Girshick, R., He, K., Dollar, P., 2017 *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2980-2988.
- Maggiore, E., Tarabalka, Y., Charpiat, G. and Alliez, P., 2017. Convolutional Neural Networks for Large-Scale Remote-Sensing Image Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2), pp. 645-657.
- Radečić, D., 2020. Softmax Activation Function Explained. [online] Medium. Available at: <<https://towardsdatascience.com/softmax-activation-function-explained-a7e1bc3ad60>> [Accessed 12 May 2021].
- Ren, S., He, K., Girshick, R. and Sun, J., 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), pp.1137-1149.
- Richard, J. A. 2013. *Remote sensing digital image analysis: an introduction*, Berlin, Springer, pp. 247-341.
- Serco Italia SPA, 2017, Burned Area Mapping with Sentinel-2 (SNAP), Portugal, version 1.2, Available at: < <https://rus-copernicus.eu/portal/the-rus-library/learn-byyourself/>>
- Thongsang, P., Hu, H., Zhou, H. and Lau, A., 2021. Imaging Enhancement in Angle-Domain Common-Image-Gathers Using the Connected-Component Labeling Method. *Pure and Applied Geophysics*, 177(10), pp. 4897-4912.

Tzutalin D., LabelImg, 2015. Available at: < <https://github.com/tzutalin/labelImg> > [Accessed 12 May 2021].