Detection of Wagyu Beef Sources with Image Classification using Convolutional Neural Network

Mr. Nattakorn Kointarangkul

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science in Computer Science
Department of Computer Engineering
FACULTY OF ENGINEERING
Chulalongkorn University
Academic Year 2020
Copyright of Chulalongkorn University

การตรวจหาต้นกำเนิดเนื้อวากิวด้วยการจำแนกภาพแบบโครงข่ายประสาทคอนโวลูชัน

นายณัฏฐกร โคอินทรางกูร

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
สาขาวิชาวิทยาศาสตร์คอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2563

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Thesis Title	Detection of Wagyu Beef Sources with Image Classification using Convolutional Neural Network
By	Mr. Nattakorn Kointarangkul
Field of Study	Computer Science
Thesis Advisor	Associate Professor Yachai Limpiyakorn, Ph.D.

Accepted by the FACULTY OF ENGINEERING, Chulalongkorn University in Partial Fulfillment of the Requirement for the Master of Science

.... Dean of the FACULTY OF ENGINEERING
(Professor SUPOT TEACHAVORASINSKUN, D.Eng.)

THESIS COMMITTEE

.... Chairman
(Assistant Professor SUKREE SINTHUPINYO, Ph.D.)
.... Thesis Advisor
(Associate Professor Yachai Limpiyakorn, Ph.D.)
.... External Examiner
(Paskorn Apirukvorapinit, Ph.D.)

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ณัฏฐกร โคอินทรางกูร : การตรวจหาต้นกำเนิดเนื้อวากิวด้วยการจำแนกภาพแบบโครงข่ายประสาทคอนโวลูชัน. (Detection of Wagyu Beef Sources with Image Classification using Convolutional Neural Network) อ.ที่ปรึกษาหลัก : รศ. ดร.ญาใจ ลิ่มปิยะกรณ์

เนื้อวากิวมีต้นกำเนิดในญี่ปุ่น อย่างไรก็ตาม ยังมีเนื้อวากิวประเภทต่างๆจำหน่ายในตลาดทั่วโลก แหล่งผลิตหลักๆ อาทิ ออสเตรเลีย สหรัฐอเมริกา แคนาดา และอังกฤษ ทั้งนี้ เนื้อวากิวญี่ปุ่นแท้จะเป็นที่รู้จักกันดีสำหรับลายเนื้อหินอ่อนที่หนาแน่น ความนุ่มและรสชาติที่ฉ่ำลิ้น เป็นที่สังเกตว่า เนื้อวากิวจากแหล่งผลิตที่ต่างกันจะมีรสชาติ เนื้อสัมผัส และคุณภาพที่แตกต่างกัน งานวิจัยนี้นำเสนอแนวทางบนพื้นฐานปัญญาประดิษฐ์เพื่อระบุแหล่งผลิตเนื้อวากิวด้วยการจำแนกประเภทภาพ ข้อมูลนำเข้าเป็นภาพที่รวบรวมจากแหล่งที่น่าเชื่อถือบนอินเทอร์เน็ต และเพิ่มจำนวนภาพด้วยดีซีแกน โครงข่ายประสาทเชิงลึกซีเอ็นเอ็นถูกสร้างขึ้นเพื่อตรวจหาแพตเทิร์นลายไขมันหินอ่อนของเนื้อวากิวสองแหล่ง คือ ญี่ปุ่นและออสเตรเลีย ผลการทำนายให้ค่าความแม่นยำสูงที่ 94.2% การดำเนินการทดลองขั้นต่อไปได้เพิ่มเนื้อวากิวสหรัฐอเมริกาสำหรับการจำแนกหลายประเภท แบบจำลองการตรวจหาวัตถุได้ถูกสร้างขึ้นโดยการฝึกสอนซีเอ็นเอ็นเชิงภูมิภาคหรืออาร์ซีเอ็นเอ็น ซึ่งให้ผลลัพธ์การทำนายที่ความแม่นยำ 79.8% พบว่า แบบจำลองการเรียนรู้โครงข่ายประสาทคอนโวลูชันทั้งสองเป็นวิธีที่หวังว่าสามารถใช้สำหรับเรียนรู้ลักษณะแพตเทิร์นของเอกลักษณ์ชั้นลายไขมันหินอ่อนได้อย่างรวดเร็ว ซึ่งจักเป็นประโยชน์กับลูกค้าสำหรับการเลือกซื้อเนื้อวากิวในราคาที่เหมาะสม

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สาขาวิชา วิทยาศาสตร์คอมพิวเตอร์ ลายมือชื่อนิสิต ....
ปีการศึกษา 2563 ลายมือชื่อ อ.ที่ปรึกษาหลัก ....

6270076621 : MAJOR COMPUTER SCIENCE

KEYWORD: Convolutional neural network, GANs, Image classification, Wagyu beef, Marbling analysis

Nattakorn Kointarangkul : Detection of Wagyu Beef Sources with Image Classification using Convolutional Neural Network. Advisor: Assoc. Prof. Yachai Limpiyakorn, Ph.D.

Wagyu beef originated in Japan. However, there are many types of Wagyu beef in the market around the globe. Primary sources include Australia, USA, Canada and the United Kingdom. The authentic Japanese Wagyu is well known for its intense marbling, juicy rich flavor and tenderness. Observing that there are differences in flavor, texture, and quality between distinct sources of Wagyu. This research presents an AI-based approach to identify Wagyu beef sources with image classification. The input images were collected from reliable sources on the internet and augmented with DCGAN. Deep neural networks, CNN, was constructed to detect the marbled fat patterns of two sources, Japanese Wagyu and Australian Wagyu. The prediction of Wagyu sources achieved high accuracy of 94.2%. Further experiment was conducted for multi-classification with the additional source of the US wagyu. The object detection model was trained using Region-based CNN, R-CNN, providing the prediction accuracy of 79.8%. The learning models of Convolutional Neural Networks were found to be promising methods for the rapid characterization of the unique patterns of marbled fat layers. These classifiers would benefit the customers for buying Wagyu beef at reasonable prices.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Field of Study: Computer Science Student's Signature ....
Academic Year: 2020 Advisor's Signature ....

ACKNOWLEDGEMENTS

The completion of this thesis could not have been possible without the assistance and participation of so many people whose names may not all be enumerated. Their contributions are sincerely appreciated and gratefully acknowledged. However, the group would like to express deep appreciation and indebtedness particularly to the following.

Assoc. Prof. Dr. Yachai Limpiyakorn, Asst. Prof. Dr. Sukree Sinthupinyo and Dr. Paskorn Apirukvorapinit for their time, endless effort and generous advice during the presentation.

Last but not least, to all relatives, friends, others who in one way or another shared their support, either morally, financially and physically, thank you.

Nattakorn Kointarangkul

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

TABLE OF CONTENTS

Page

.... iii

ABSTRACT (THAI) iii

.... iv

ABSTRACT (ENGLISH) iv

ACKNOWLEDGEMENTS v

TABLE OF CONTENTS vi

LIST OF TABLES viii

LIST OF FIGURES ix

CHAPTER 1 16

1.1 Statement of the problems 16

1.2 Objective 17

1.3 Scope of Study 17

1.4 Research Methodology 17

1.5 Outcomes 18

1.6 Thesis Publication 18

CHAPTER 2 19

2.1 Convolutional Neural Network (CNN) 19

2.2 Deep Convolutional Generative Adversarial Network (DCGAN) 20

2.3 Felzenszwalb Segmentation 22

2.4 Selective Search 23

2.5 Insection over Union (IoU) 27

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.6 Region-based Convolutional Neural Network (R-CNN) 29
CHAPTER 3 31
3.1 Methodology and Working Environments 31
3.2 Data Preparation 35
3.3 Phase 1 37
3.4 Phase 2 38
CHAPTER 4 39
4.1 Results of generating images via DCGAN 39
4.2 Results of Phase 1 40
4.2.1 Results of Phase 1.1 40
4.2.2 Results of Phase 1.2 45
4.3 Results of Phase 2 49
4.3.1 Results of Phase 2.1 49
4.3.2 Results of Phase 2.2 62
4.4 Overall Test Results 74
CHAPTER 5 77
REFERENCES 78
APPENDIX 80
VITA 89

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

LIST OF TABLES

Page

Table 1. Size of dataset containing images of Japanese and Australian Wagyu for binary classification. 35

Table 2. Size of dataset containing images of Japanese, Australian, and US Wagyu for multi-classification. 35

Table 3 Comparison of accuracy of CNN binary classification 74

Table 4 Comparisons of accuracy of CNN multi-classification 74

Table 5 Comparisons of accuracy of R-CNN binary classification. 74

Table 6 Comparison of accuracy of R-CNN multi-classification 74

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

LIST OF FIGURES

Page

Figure 1. The process of calculating convolutional layers and the architecture of the network [3]. 20

Figure 2. How both the generator and the discriminator improve over time [6]........ 21

Figure 3. The network architecture of the generator of DCGANs [5] 21

Figure 4. An example of an image transformed by Felzenszwalb Segmentation......... 23

Figure 5. A result in steps when Selective Search is implemented........ 24

Figure 6. An illustration of selective search capturing the possible target objects [10]. 25

Figure 7. An example of possible proposals generated by Selective Search [9]. 28

Figure 8. The formula used for calculating IoU. 28

Figure 9. Examples of IoU scores generated by various positions of bounding box [9]. 29

Figure 10. The process of calculating Region-based CNN [9]. 30

Figure 11. An example NVIDIA CUDA ,which could be used as GPU. 33

Figure 12. An example of NVIDIA CUDA command to trigger its GPU. 34

Figure 13. A format of constructing a data folder for training via Keras Tensorflow.... 36

Figure 14. The architecture of trained CNNs. 37

Figure 15. The architecture of Region-based CNN. 38

Figure 16. How DCGAN images evolved over the period trained. 39

Figure 17. Original Japanese Wagyu images (Left) and Japanese Wagyu images augmented with DCGAN (Right). 40

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Figure 18. Original Australian Wagyu images (Left) and Australian Wagyu images augmented with DCGAN (Right). 40

Figure 19. Accuracy (Left) and loss (Right) of the 1st fold of 5-fold cross validation of the CNN binary classification without DCGAN images. 41

Figure 20. Accuracy (Left) and loss (Right) of the 2nd fold of 5-fold cross validation of the CNN binary classification without DCGAN images. 41

Figure 21. Accuracy (Left) and loss (Right) of the 3rd fold of 5-fold cross validation of the CNN binary classification without DCGAN images. 41

Figure 22. Accuracy (Left) and loss (Right) of the 4th fold of 5-fold cross validation of the CNN binary classification without DCGAN images. 42

Figure 23. Accuracy (Left) and loss (Right) of the 5th fold of 5-fold cross validation of the CNN binary classification without DCGAN images. 42

Figure 24. Accuracy (Left) and loss (Right) of the 1st fold of 5-fold cross validation of the CNN binary classification with DCGAN images. 43

Figure 25. Accuracy (Left) and loss (Right) of the 2nd fold of 5-fold cross validation of the CNN binary classification with DCGAN images. 43

Figure 26. Accuracy (Left) and loss (Right) of the 3rd fold of 5-fold cross validation of the CNN binary classification with DCGAN images. 44

Figure 27. Accuracy (Left) and loss (Right) of the 4th fold of 5-fold cross validation of the CNN binary classification with DCGAN images 44

Figure 28. Accuracy (Left) and loss (Right) of the 5th fold of 5-fold cross validation of the CNN binary classification with DCGAN images. 44

Figure 29. Accuracy (Left) and loss (Right) of the 1st fold of 5-fold cross validation of the CNN multi-classification without DCGAN images. 45

Figure 30. Accuracy (Left) and loss (Right) of the 2nd fold of 5-fold cross validation of the CNN multi-classification without DCGAN images. 45

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Figure 31. Accuracy (Left) and loss (Right) of the 3rd fold of 5-fold cross validation of the CNN multi-classification without DCGAN images. 46

Figure 32. Accuracy (Left) and loss (Right) of the 4th fold of 5-fold cross validation of the CNN multi-classification without DCGAN images. 46

Figure 33. Accuracy (Left) and loss (Right) of the 5th fold of 5-fold cross validation of the CNN multi-classification without DCGAN images. 46

Figure 34. Accuracy (Left) and loss (Right) of the 1st fold of 5-fold cross validation of the CNN multi-classification with DCGAN images........ 47

Figure 35. Accuracy (Left) and loss (Right) of the 2nd fold of 5-fold cross validation of the CNN multi-classification with DCGAN images........ 47

Figure 36. Accuracy (Left) and loss (Right) of the 3rd fold of 5-fold cross validation of the CNN multi-classification with DCGAN images........ 48

Figure 37. Accuracy (Left) and loss (Right) of the 4th fold of 5-fold cross validation of the CNN multi-classification with DCGAN images........ 48

Figure 38. Accuracy (Left) and loss (Right) of the 5th fold of 5-fold cross validation of the CNN multi-classification with DCGAN images........ 48

Figure 39. Accuracy of the 1st fold of 5-fold cross validation of the R-CNN binary classification without DCGAN images........ 49

Figure 40. Loss of the 1st fold of 5-fold cross validation of the R-CNN binary classification without DCGAN images........ 50

Figure 41. Accuracy of the 2nd fold of 5-fold cross validation of the R-CNN binary classification without DCGAN images........ 50

Figure 42. Loss of the 2nd fold of 5-fold cross validation of the R-CNN binary classification without DCGAN images........ 51

Figure 43. Accuracy of the 3rd fold of 5-fold cross validation of the R-CNN binary classification without DCGAN images........ 51

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Figure 44. Loss of the 3rd fold of 5-fold cross validation of the R-CNN binary classification without DCGAN images. 52

Figure 45. Accuracy of the 4th fold of 5-fold cross validation of the R-CNN binary classification without DCGAN images. 52

Figure 46. Loss of the 4th fold of 5-fold cross validation of the R-CNN binary classification without DCGAN images. 53

Figure 47. Accuracy of the 5th fold of 5-fold cross validation of the R-CNN binary classification without DCGAN images. 53

Figure 48. Loss of the 5th fold of 5-fold cross validation of the R-CNN binary classification without DCGAN images. 54

Figure 49. Accuracy of the 1st fold of 5-fold cross validation of the R-CNN binary classification with DCGAN images. 55

Figure 50. Loss of the 1st fold of 5-fold cross validation of the R-CNN binary classification with DCGAN images. 55

Figure 51. Accuracy of the 2nd fold of 5-fold cross validation of the R-CNN binary classification with DCGAN images. 56

Figure 52. Loss of the 2nd fold of 5-fold cross validation of the R-CNN binary classification with DCGAN images. 56

Figure 53. Accuracy of the 3rd fold of 5-fold cross validation of the R-CNN binary classification with DCGAN images. 57

Figure 54. Loss of the 3rd fold of 5-fold cross validation of the R-CNN binary classification with DCGAN images. 57

Figure 55. Accuracy of the 4th fold of 5-fold cross validation of the R-CNN binary classification with DCGAN images. 58

Figure 56. Loss of the 4th fold of 5-fold cross validation of the R-CNN binary classification with DCGAN images. 58

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Figure 57. Accuracy of the 5th fold of 5-fold cross validation of the R-CNN binary classification with DCGAN images. 59

Figure 58. Loss of the 5th fold of 5-fold cross validation of the R-CNN binary classification with DCGAN images. 59

Figure 59 Example 1 illustrating how R-CNN could recognize the marbled fat layers of Japanese Wagyu 60

Figure 60 Example 2 illustrating how R-CNN could recognize the marbled fat layers of Japanese Wagyu 60

Figure 61. Example 1 illustrating how R-CNN could recognize the marbled fat layers of Australian Wagyu 61

Figure 62. Example 2 illustrating how R-CNN could recognize the marbled fat layers of Australian Wagyu 61

Figure 63. Illustrating how the machine could not recognize the pattern of the marbled fat layers of the US Wagyu beef 62

Figure 64. Accuracy of the 1st fold of 5-fold cross validation of the R-CNN multi-classification without DCGAN images 63

Figure 65. Loss of the 1st fold of 5-fold cross validation of the R-CNN multi-classification without DCGAN images. 63

Figure 66. Accuracy of the 2nd fold of 5-fold cross validation of the R-CNN multi-classification without DCGAN images. 64

Figure 67. Loss of the 2nd fold of 5-fold cross validation of the R-CNN multi-classification without DCGAN images. 64

Figure 68. Accuracy of the 3rd fold of 5-fold cross validation of the R-CNN multi-classification without DCGAN images. 65

Figure 69. Loss of the 3rd fold of 5-fold cross validation of the R-CNN multi-classification without DCGAN images. 65

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Figure 70. Accuracy of the 4th fold of 5-fold cross validation of the R-CNN multi-classification without DCGAN images. 66

Figure 71. Loss of the 4th fold of 5-fold cross validation of the R-CNN multi-classification without DCGAN images. 66

Figure 72. Accuracy of the 5th fold of 5-fold cross validation of the R-CNN multi-classification without DCGAN images. 67

Figure 73. Loss of the 5th fold of 5-fold cross validation of the R-CNN multi-classification without DCGAN images. 67

Figure 74. Accuracy of the 1st fold of 5-fold cross validation of the R-CNN multi-classification with DCGAN images. 68

Figure 75. Loss of the 1st fold of 5-fold cross validation of the R-CNN multi-classification with DCGAN images. 68

Figure 76. Accuracy of the 2nd fold of 5-fold cross validation of the R-CNN multi-classification with DCGAN images. 69

Figure 77. Loss of the 2nd fold of 5-fold cross validation of the R-CNN multi-classification with DCGAN images. 69

Figure 78. Accuracy of the 3rd fold of 5-fold cross validation of the R-CNN multi-classification with DCGAN images. 70

Figure 79. Loss of the 3rd fold of 5-fold cross validation of the R-CNN multi-classification with DCGAN images. 70

Figure 80. Accuracy of the 4th fold of 5-fold cross validation of the R-CNN multi-classification with DCGAN images. 71

Figure 81. Loss of the 4th fold of 5-fold cross validation of the R-CNN multi-classification with DCGAN images. 71

Figure 82. Accuracy of the 5th fold of 5-fold cross validation of the R-CNN multi-classification with DCGAN images. 72

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Figure 83. Loss of the 5th fold of 5-fold cross validation of the R-CNN multi-classification with DCGAN images. 72

Figure 84. Illustrating how the machine could recognize the pattern of marbled fatty layers of the US Wagyu Beef 73

Figure 85 Illustrating how the R-CNN could not recognize the pattern of marbled fatty layers of artificial marbling beef. 73

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

CHAPTER 1

Introduction

1.1 Statement of the problems

According to the research and questionnaires conducted, many Wagyu Beef related occupations, for instance, Chefs and Main Kitchen Buyers, usually struggle to tell the difference between the true sources of the Wagyu Beef if they were not labelled [1]. When purchasing the Wagyu Beef, if the meat was wrongly labelled, more than often, the buyers would end up buying the Wagyu they do not desire. Since the sources of the Wagyu Beef could not be distinguished through eyesight, it leaves the buyers no choice but to blindfold-purchase them on some occasions. Whether the case was by intention or not, it often leaves the buyers in confusion, and more than likely, lead them to obtain lower graded Wagyu Beef as opposed to buying premium one. As one of the solutions to this problem, since each kind of Wagyu Beef have very distinguishable fatty marble layers, we shall play this feature to our advantage via deep learning (computer vision based). These fatty layers are expected to have different patterns for each source; therefore, computer vision would be a suitable tool in order to perform the classification task on this specific topic. However, computer vision is considered a very broad subject; in order to be more specific, the model used for this computer vision assignment would be Convolutional Neural Network or CNN. Later in the next section, CNN's architecture and implementations will be illustrated and explained in detail. Although, CNN has been invented and existed for some time, yet it remains as one the most powerful tool for deep learning-based approach. As a result, CNN architecture would be supporting this research as the backbone throughout. Although, some adjustments may have to be made and modified; the main and unique

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

characters of CNN would still hold, and inevitably, serve as the base for several models used to train the data sets of this research.

1.2 Objective

To present and implement an AI-based approach for classifying the true origins of the given Wagyu beef images.

1.3 Scope of Study

1. The chosen sources of Wagyu beef images will be binary based and expand to multi-classification; including out of distribution classification

2. The sources of datasets come from the internet, e.g.

- Australian Wagyu Association [1]
- American Wagyu Association [2]
- The Wagyu Shop [3]

3. The model performance will be evaluated by the accuracy on the test set after the model has been trained and satisfies the requirements specification.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

1.4 Research Methodology

1. Study several related works about CNNs and GANs.

2. Collect and preprocessing the datasets of Wagyu beef images.

3. Train and fine-tune the model.

4. Analyze the findings.

5. Evaluate and summarize the outcome.

5. Publish the article.

6. Complete thesis writing.

1.5 Outcomes

Gain an AI-based visual aid for classifying the authentic origin of a Wagyu beef image in order to encourage and promote a fair trade in the Wagyu Beef exchanged market.

1.6 Thesis Publication

Parts of the thesis had been published in the conference as following:

N. Kointarangkul and Y. Limpiyakorn, “Detection of Wagyu beef sources with image classification using convolutional neural network”, in Proceedings of the 6th International Workshop on Pattern Recognition, 2021, Chengdu, China.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

CHAPTER 2

Literature Review

2.1 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) [4] was chosen as the network architecture in this research due to its ability to handle multidimensional data. For cases associated with images, a CNN will take in a total of 3-dimensional data (height x width x depth) and then arrange the same number of neurons in order to proceed further to latter stages as shown in Figure 1. Furthermore, each neuron, within the given CNN, will have full connection to all elements of the previous layer, and would also have access to some elements of the neighbouring region through the “receptive field (filter size)”. The CNN’s neurons share the same weights and bias throughout the network. The so-called convolutional layers would act as a filter (kernel) in order to modify the image, which in the end, would lead to a feature transformation. Figure 1 illustrates an example of convolution operation computed on the given matrix. The formal mathematical term for the previous operation is ‘cross-correlation’ [3]. The cross-correlation of a matrix, x, with a filter, w, is provided in equation (1). The convolution results could be obtained from the formula below by flipping the kernels before it.

$$\forall i \in [0, H_o - 1] \text{ and } \forall j \in [0, W_o - 1]$$

$$(w * x)_{i,j} = \sum_{l=0}^{k-1} \sum_{m=0}^{k-1} w_{l,m} \cdot x_{i+l,j+m} \tag{1}$$

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

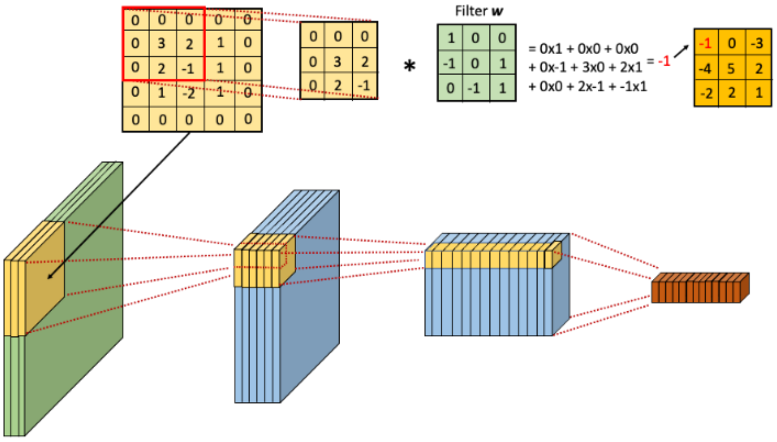


Figure 1. The process of calculating convolutional layers and the architecture of the network [3].

2.2 Deep Convolutional Generative Adversarial Network (DCGAN)

Before exploring the DCGAN [5], the standard Generative Adversarial Networks (GANs) [6] would be looked at first. GANs are divided into 2 parts: 1) A generator, and 2) A discriminator.

The generator ('the artist') creates fake images that look as close to the real ones as possible and will always try to convince the discriminator ('the art critic') that the images created are real [7]. Whereas the discriminator will act as a detective and try to tell the difference between real images and fake images.

As the training processes illustrated in Figure 2, the generator will become so much better at creating images and at the same time the discriminator will also become so much better at catching fake images. The process would keep continuing until it reaches the equilibrium point where the discriminator could no longer tell the difference between real images and fake images.

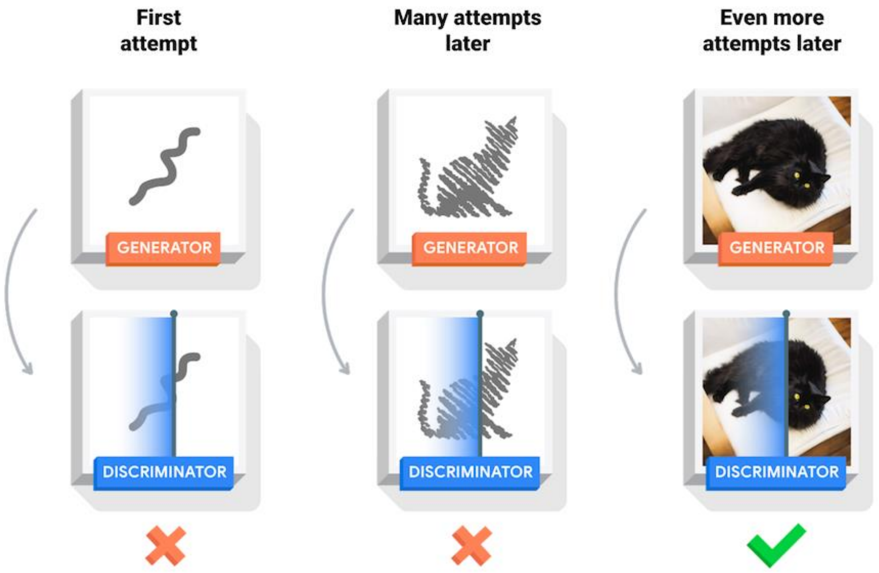


Figure 2. How both the generator and the discriminator improve over time [6]

However, the DCGANs [5] are an adjusted model based on the original GANs. Figure 3 depicts the architecture of the generator of DCGAN. Basically, DCGANs are the ‘image version’ of the most fundamental implementation of GANs. The deep convolutional neural networks are embedded deep within the system of Generator-Discriminator Framework, in which, the images generated are created from the given distribution of noisy data [7].

CHULALONGKORN UNIVERSITY

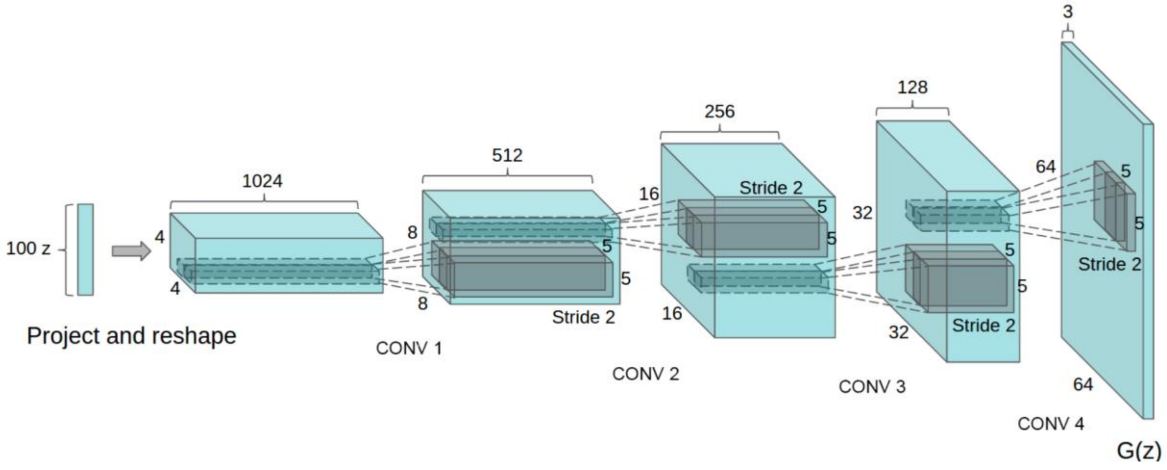


Figure 3. The network architecture of the generator of DCGANs [5]

These points are the adjustments made based on the original GANs [8]:

- Replace all max pooling with convolutional stride
- Use transposed convolution for upsampling
- Eliminate fully connected layers
- Use batch normalization except the output layer for the generator and input layer of the discriminator
- Use ReLU in the generator except for the output which uses tanh
- Use LeakyReLU in the discriminator

2.3 Felzenszwalb Segmentation

Felzenszwalb [9] is brought in to extract the segments; all of these components in the image chosen would be taken into consideration via the procedure.

- Color
- Texture
- Size
- Shape Compatibility

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

An example of Felzenszwalb is illustrated below in Figure 4.

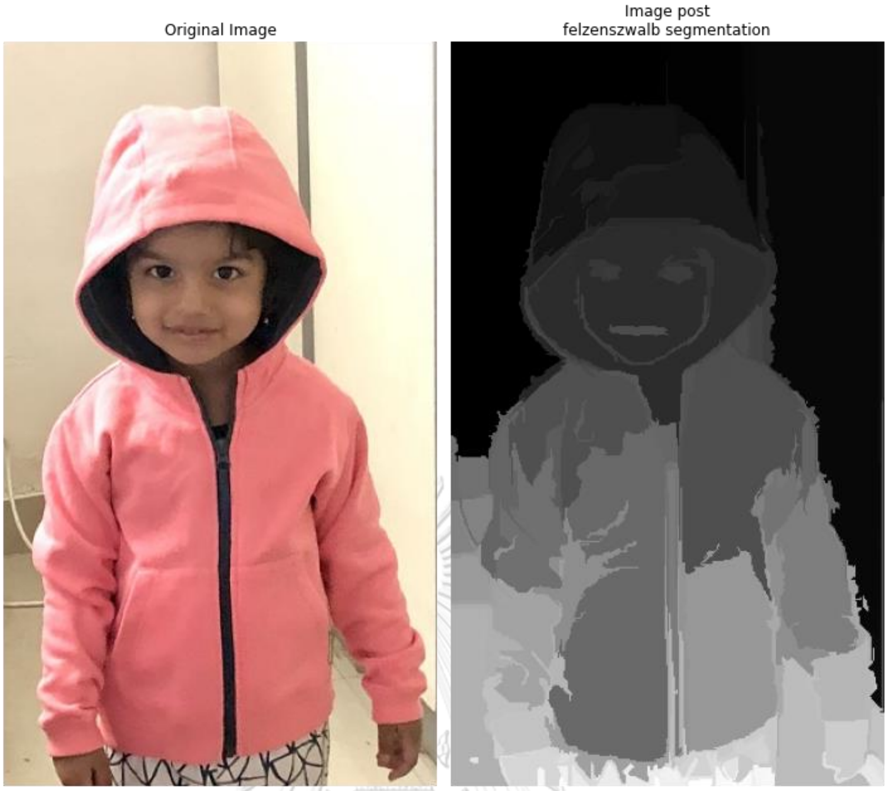


Figure 4. An example of an image transformed by Felzenszwalb Segmentation.

2.4 Selective Search

Selective Search is an algorithm created based on the distinction of color, texture, size and shape compatibility of an image given [9]. It is designed to have a very high recall along with quick response when implemented and is widely used as an algorithm for object detection.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Selective search is nitiated by implementing the graph-based segentation method by Felzenzwalb and Huttenlocher. Figure 5 shows the result after transforming the input image with the segmentation method stated. Then, the intensity of the pixels is over-segmented by the algorithm.

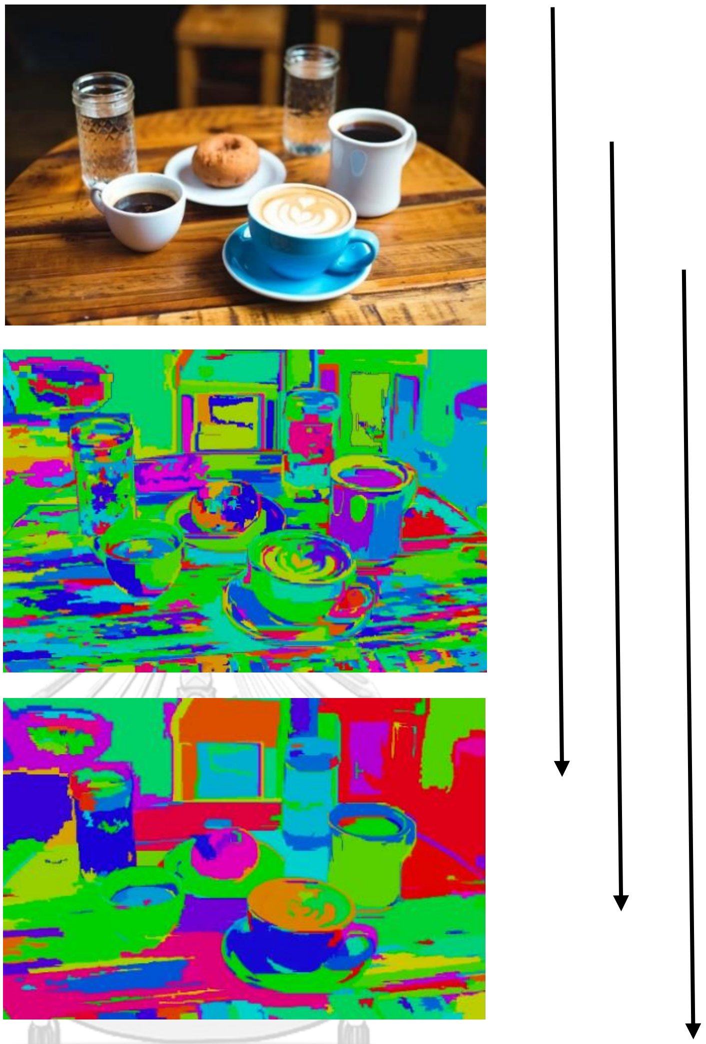


Figure 5. A result in steps when Selective Search is implemented

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

The selective search algorithm would then take these oversegments as its starting point and continue via the following steps:

1. Add all bounding boxes corresponding to the divided segment parts to the list of regional proposal
2. Group adjacent segment parts based on similarity
3. Go back to step 1 and iterate

After each iteration, larger segments will be formed and added to the list of region proposals. Therefore, as the algo iterates the segments will keep getting larger by forming and adding to the list of region proposals. This is seen as a bottom-up approach by many [10], since the structures are originate formed from small segments and piling up to become larger ones as shown in Figure 6.

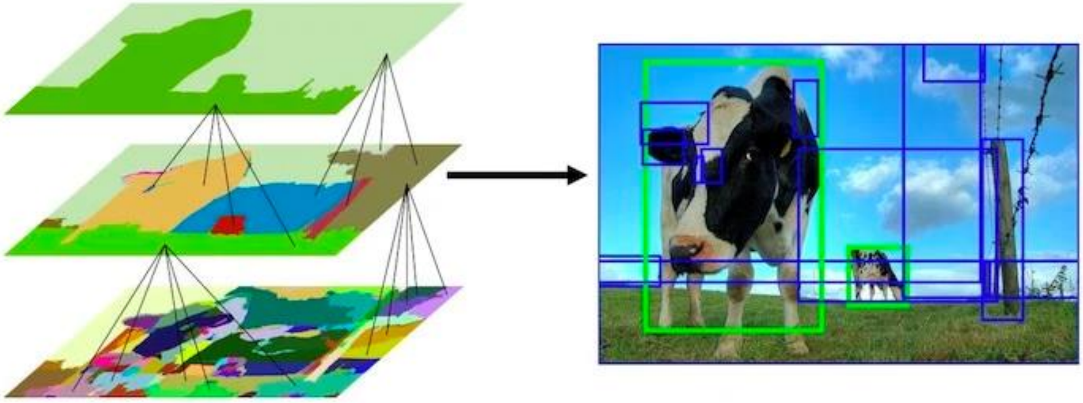


Figure 6. An illustration of selective search capturing the possible target objects [10].

In order to iterate successfully, the selective search must rely on its unique way of calculating similarity between two regions. The method could be examined and classified into 4 categories as follow:

- Color Similarity:- A color histogram of 25 bins is figured for each channel of the image [10]. Afterwards, the all of the channels of the histograms are concatenated to form a color descriptor resulting into a 25x3 = 7-dimensional color descriptor.

  The calculation of color similarity of 2 regions based on histogram intersection is processed as Equation (2):

$$s_{color}(r_i, r_j) = \sum_{k=1}^{n} min(c_i^k, c_j^k)$$

where c^k_i is the histogram value for the k^{th} bin in color descriptor (2)

- Texture Similarity:- Texture features are performed by extracting Gaussian derivatives at 8 orientations for each channel [10]. Within a single color channel for each channel, a 10-bin histogram is calculated and formed a 10x8x3 = 240-dimensional feature descriptor.

  Texture Similarity of 2 regions is processed by using histograms intersection as shown in Equation (3):

$$s_{texture}(r_i, r_j) = \sum_{k=1}^{n} min(t_i^k, t_j^k)$$

where t^k_i is the histogram value for the kth bin in texture descriptor (3)

- Size Similarity:- This feature ensures that small regions merge early. It also guaranteed that the region proposals at all scales and sizes are formed at every parts of the image given [10]. However, if this so-called size similarity does not exist. The region proposals at multiple scales would be generated at one location only, since that particular region will keep concatenating all the small regions one by one.

  Size Similarity is defined by Equation (4):

$$s_{size}(r_i, r_j) = 1 - \frac{size(r_i) + size(r_j)}{size(im)}$$

where size (im) is the size of image in pixels (4)

- Shape Compatibility:- This feature is designed for measuring how well the 2 regions(ri and rj) concatenate and fit into each other [10]. If ri fits into rj they would be merged into each other and all the gaps would be filled. However, if the 2 regions are not overlapping or even touching each other, the merging process would be terminated and cancelled.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Shape Compatibility is defined by Equation (5):

$$s_{fill}(r_i, r_j) = 1 - \frac{size(BB_{ij}) - size(r_i) - size(r_j)}{size(im)}$$

where size BB_{ij} is a bounding box around ri and rj (5)

- Final similarity:- Previously explained, the final similarity between the given two regions is defined as a linear combination of the four similarities [10], shown in Equation (6):

$$s(r_i, r_j) = a_1 s_{color}(r_i, r_j) + a_2 s_{texture}(r_i, r_j) + a_3 s_{size}(r_i, r_j) + a_4 s_{fill}(r_i, r_j) \quad (6)$$

where ri and rj are two regions or segments in the image,

and ai ∈ [1] denotes if the similarity measure is used or not.

- Results

  Usually, 1000-1200 proposals are good enough to get all the right regions proposals [9] as shown in Figure 7. When presenting the image as a result, a given range of top region is often set to crop the desired features and objects.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.5 Insection over Union (IoU)

Intersection over Union [9] is a tool built for detecting the accuracy of the predictions of bounding boxes created for the target objects. It is designed to measure how overlapping the predicted and actual bounding boxes are, while at the same time, would also measure the overall space possible for overlap. Basically, IoU is the ration between the overlapping region of the 2 bounding boxed and the combined region of both bounding boxes. Figure 8 was installed to explain the procedure previously mentioned.

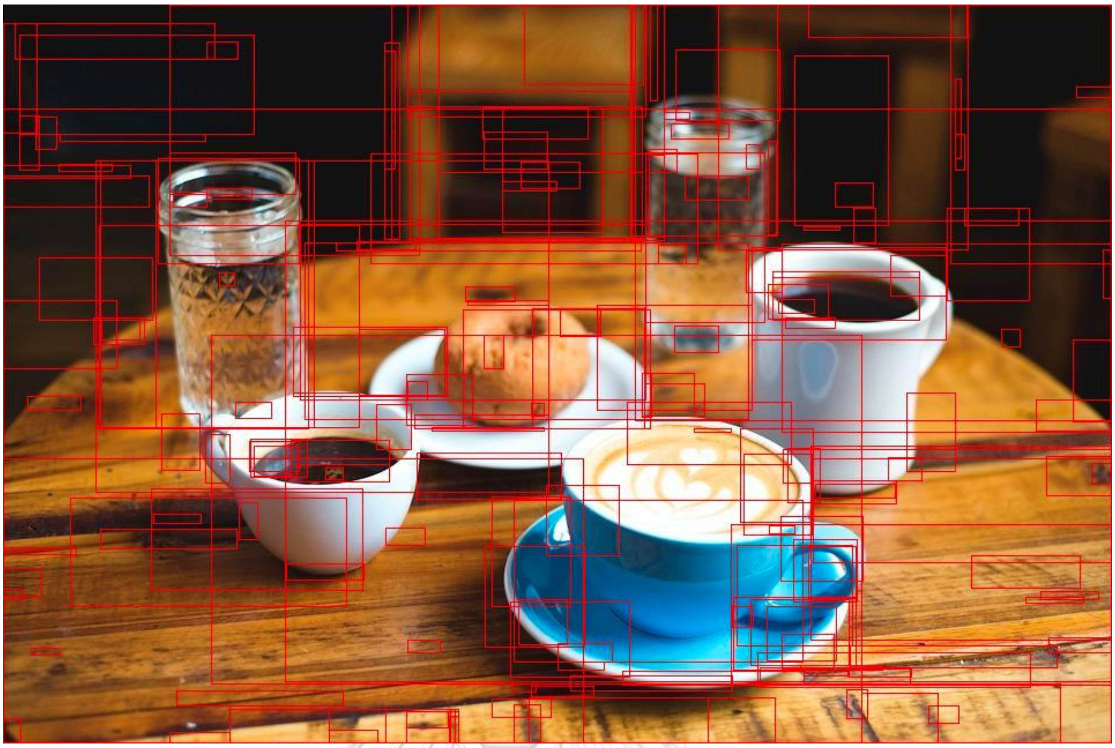


Figure 7. An example of possible proposals generated by Selective Search [9].

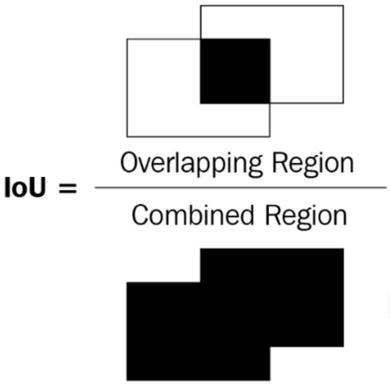


Figure 8. The formula used for calculating IoU.

The ration explained (IoU) could be examined and illustrated by Figure 9. By considering the left bounding box as the ground truth (the exact location of the object; Xmin, Xmax, Ymin, Ymax) and the right bounding box as the predicted location of the object, the result would be a ratio ranging from zero to one, with one representing the highest possible accuracy and zero representing lowest possible accuracy (no overlapping). In other words, as IoU decreases, the overlap will inevitably decrease as well. Hence, no overlapping would mean that the IoU metric is equivalent to zero.

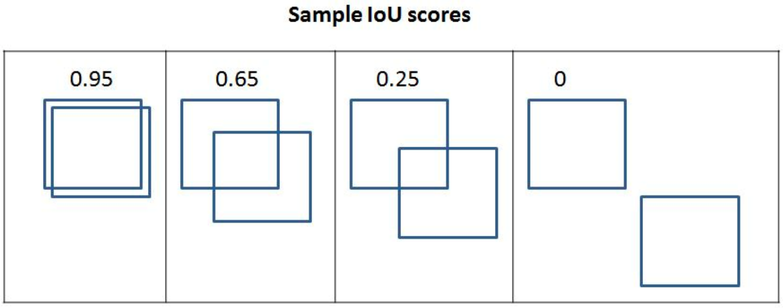


Figure 9. Examples of IoU scores generated by various positions of bounding box [9].

2.6 Region-based Convolutional Neural Network (R-CNN)

Region-based algorithm [9] is used in order to sync with the region proposals of the target object. Figure 10 illustrates the procedure of executing R-CNN visually.The steps to implement R-CNN are as follow.

- Extract the region proposals from the image given
- Resize (warp) all of the extracted regions to get all the images align by having the size
- Pass along the altered-images with newly reconfigured sizes through the network.
- Create data intended for model training, where the input features are extracted by passing the region proposals through the model, and the outputs are the classes according the to the regions given
- Connect 2 output heads, one is the class of the image and the other would be the bounding box, which corresponds to the offset of the region proposal with the ground truth bounding box
- Train the model
- Make prediction of the class of the object cropped by the bounding box.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

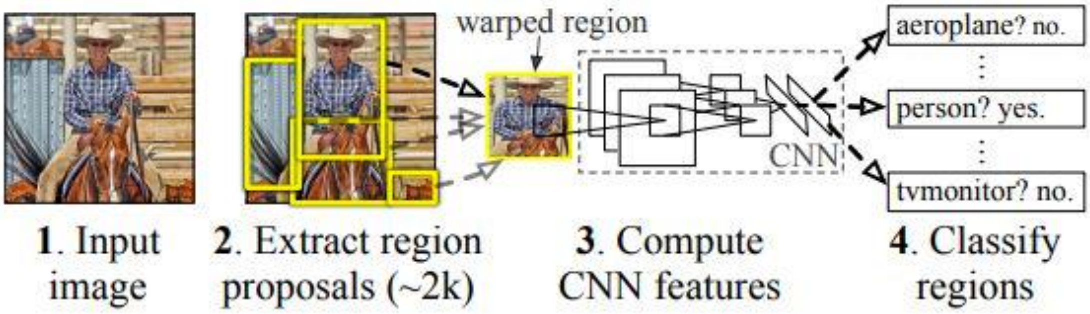


Figure 10. The process of calculating Region-based CNN [9].

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

CHAPTER 3

METHODOLOGY

3.1 Methodology and Working Environments

This research is separated into two phases:

- Phase 1: Model construction with CNN for Classification problem containing
  - 1.1 Binary class
  - 1.2 Multi-class
- Phase 2: Model construction with R-CNN for Object Detection problem containing
  - 2.1 Binary class
  - 2.2 Multi-class

The first phase was conducted to test whether the assumption is valid. Since the method was original and therefore, the outcome was unpredictable to some extent. Hence, this first phase was initiated to test out the algorithm and models, if achieving the desired result as intended initially.

The test started off with a binary classification between Japanese Wagyu Beef and Australian Wagyu Beef, since Japanese Wagyu is considered as the most of premium of all Wagyu Beef existed and, on the other hand, the Australian Wagyu Beef is considered as the least premium. Therefore, the binary-classification conducted was expected to be the easiest out of all the possible binary classifications between the Wagyu Beef from different sources.

The model in this first phase was trained on a *balanced* dataset containing images of Wagyu Beef from two distinct sources: Japanese and Australian via the CNN architecture described previously. The images of various sizes were scaled down to 256x256 pixels (RGB color PNG files) via the Imagedatagenerator function attached with

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Tensorflow. The model was implemented using Keras Tensorflow 2.0 library. The computer's CPUs were 2 GHs Quad-Core Intel Core i5 with RAM of 16GB. The python version was 3.6 along with Keras Tensorflow version 2.0. The last layer, Fully Connected layer, was implemented with 'sigmoid' activation function, producing the output value ranging from 0.0 to 1.0. In this work, the threshold was set to 0.5; that is, the output value < 0.5 (closer to 0.0) would be classified as Australian Wagyu, otherwise it would be classified as Japanese where the output value no less than 0.5 (closer to 1.0).

As the experiment turned out to be successful by reaching the expected accuracy of above 90%, the test was then expanded from a binary-classification to multiclass including. Australian Wagyu, Japanese Wagyu, and US Wagyu. The test ran on the same CNN architecture platform and the same computer configuration.

After training the model with the same balanced data set, the accuracy turned out to be lower than the required level (90%). The main issue was the missing of a powerful enough GPUs such as NVIDIA CUDA. By having NVIDIA CUDA embedded within the system, it would allow the system to perform complex computer vision tasks, especially when a multi-classification was preferred over a binary one.

As a result, the second phase protocol was initiated to serve the need of a more complex computer vision tasks. This second phase was trained on a Window platform with GPU enabled ability. Since the code and algorithms written, involved a CUDA command, which required a GPU from NVIDIA CUDA (Figure 11) to perform such task, and that only Window and Ubuntu platforms are capable of executing NVIDIA GPU. Hence, the macOS system was dropped and replaced with the Window platform (GPU enabled) platform.

Figure 11. An example NVIDIA CUDA ,which could be used as GPU.

This latter phase was designed to tackle the multi-classification problem including the Out-of-Distribution problem (OOD) via the R-CNN model to build an object detection framework. Furthermore, this second attempt was trained on the same balanced datasets as the first one, which means that the DCGAN images were included along with the original files.

The test started off by multi-classifying Australian Wagyu, Japanese Wagyu, and OOD (Out of Distribution class such as other types of meat, which were not Wagyu related). After training the data via the Region-Based CNN architecture, the model reached the accuracy above the required level of 90%, and consequently, proving that Region Based CNN, together with, CUDA enabled platform were capable of performing such tasks.

After previous success, the test was then expanded, including the US Wagyu Beef in order to make the model more robust and more complete in terms of a multi-classification CNN platform. The accuracy proved was satisfying and rose above the desired level of accuracy (90%) and brought the test to an end.

The main programming language used in this work was mainly Python3. The code written was divided into 4 main parts:

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

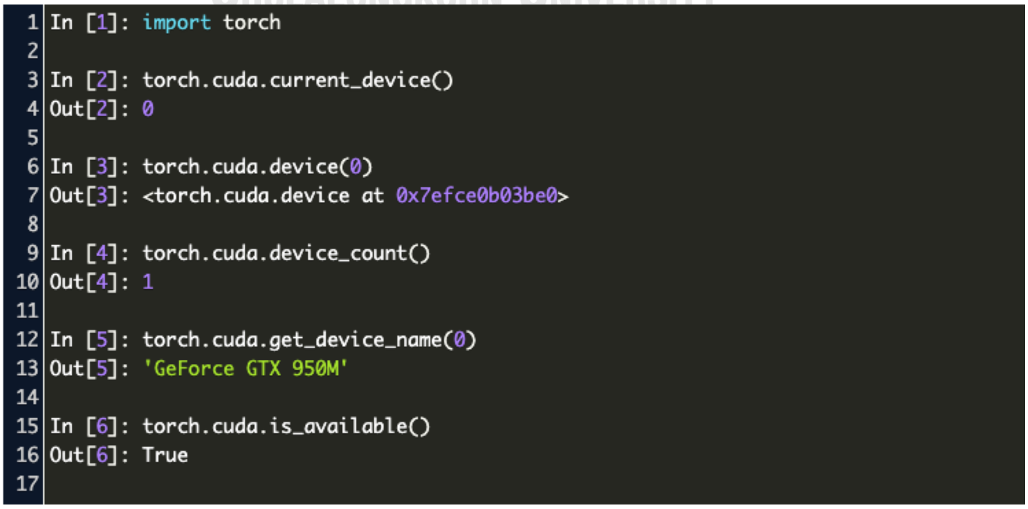
- Data preparation and customization
- Construction of the network architecture (CNN and Region based CNN)
- Model training via the architecture built
- Evaluation of the outcome

Both PyTorch and Keras Tensorflow were chosen to perform the neural network and deep learning algorithm. To be precise, Keras Tensoflow was used for the first phase (CNN) since it is a more user-friendly platform than PyTorch and was therefore a good starting point in order to gain knowledge from the experiment during the initial stages. This particular framework required external libraries as follow:

- Numpy to create and contain elements in an array
- Pandas for performing data mining and analysing tasks
- Pillow to read images as input and output
- Keras Tensorflow 2.0 to perform CNN algorithm

However, for Phase 2, PyTorch was chosen to run the model since it had the capability of handling NVIDIA CUDA. Figure 12 shows an example of how to check whether CUDA is enabled in PyTorch framework via Python Scripts.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY



```
In [1]: import torch

In [2]: torch.cuda.current_device()
Out[2]: 0

In [3]: torch.cuda.device(0)
Out[3]: <torch.cuda.device at 0x7efce0b03be0>

In [4]: torch.cuda.device_count()
Out[4]: 1

In [5]: torch.cuda.get_device_name(0)
Out[5]: 'GeForce GTX 950M'

In [6]: torch.cuda.is_available()
Out[6]: True
```

Figure 12. An example of NVIDIA CUDA command to trigger its GPU.

3.2 Data Preparation

The sources of data came from the internet, including Australian Wagyu Association [1], American Wagyu Association [2], and The Wagyu Shop [3]. The dataset used for training were square fractions of the image files of the meat and had to be of high definition along with the fact that the fatty layers must be presented in a visible manner. The research goes by the rule of one data per one piece of Wagyu Beef, with that being said, due to the limited number of the original images of both Japanese and Australian Wagyu, DCGAN was applied for data augmentation to increase the number of training data in order to avoid the problems of bias and overfittings.

Table 1 & 2 summarize the number collected images, augmented images with DCGAN, and the total size of the dataset of each Wagyu source.

Table 1. Size of dataset containing images of Japanese and Australian Wagyu for binary classification.

	JPN	AUS	Total
Collected Images	350	350	**700**
DCGAN Images	250	250	**500**
Total	**600**	**600**	**1200**

Table 2. Size of dataset containing images of Japanese, Australian, and US Wagyu for multi-classification.

	JPN	AUS	US	Total
Collected Images	350	350	350	**1050**
DCGAN Images	250	250	250	**750**
Total	**600**	**600**	**600**	**1800**

The desired parts from the images of Wagyu beef from both Japan and Australia will be cropped showing a detectable marbled fatty layer of the Wagyu beef. Then all the images were resized to formal size and were arranged to the format of TensorFlow ImageGeneration folder as depicted in Figure 13. Please note that if the images were not exactly arranged by the format given, the neural network process would not be able to perform.

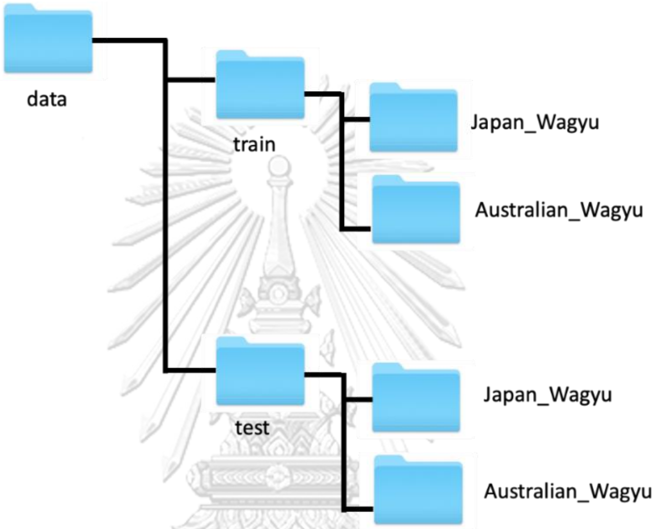


Figure 13. A format of constructing a data folder for training via Keras Tensorflow.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

In this research, YBAT program [11] would be used in order to perform the conversion of JPEG or PNG files to XML files. The reason why XML files are needed is because that it could be transformed into CSV files, which as a result, would contain the location (Xmin, Xmax, Ymin, Ymax) of the objects of each files we wish the machine to learn. The YBAT program could be easily download since it is an open source. The steps to proceed in order to create the XML files locating the objects correctly are demonstrated in Appendix.

3.3 Phase 1

The modelling of Convolutional Neural Networks contained the total number of ten layers, each of which was assigned independent parameters such as activation function, padding size, etc. Figure 14 illustrates the basic architectural nature of the network implemented. Note that the last dense layer was implemented with 'sigmoid' activation function due to its binary nature.

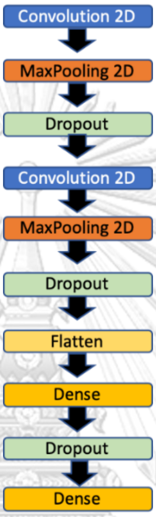


Figure 14. The architecture of trained CNNs.

The model was trained on a *balanced* dataset containing the images of Wagyu Beef from two distinct sources: Japanese and Australian. The images of various sizes were scaled down to 256x256 pixels (RGB color PNG files) via the Imagedatagenerator function attached with Tensorflow. The model was implemented using Keras Tensorflow 2.0 library. The computer's CPUs were 2 GHs Quad-Core Intel Core i5 with RAM of 16GB. The python version was 3.6 along with Keras Tensorflow version 2.0. The last layer, Fully Connected layer, was implemented with 'sigmoid' activation function, producing the output value ranging from 0.0 to 1.0. In this work, the threshold was set to 0.5; that is, the output value < 0.5 (closer to 0.0) would be classified as Australian Wagyu, otherwise it would be classified as Japanese where the output value no less than 0.5 (closer to 1.0).

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

For the multi-classification model, the same computer system was used along with the same CNN architecture. However, the last dense layer was inevitably changed to 'SoftMax' to support multi-classification task. Hence, the outcome or the predicted value was created in a form of a probability for each class. For example, if the outcome was stated as follows: [0.1 0.2 0.7]

The 3rd class would be predicted as the class of the inserted Wagyu Beef image, since it possessed the highest probability (0.7). Note that the total sum of all the probabilities of the classes will always add up to exactly 1.0, i.e. from the above predicted value the total sum = 0.1 + 0.2 + 0.7 = 1.0

3.4 Phase 2

For this latter stage of the research, Region-based CNN was implemented to perform object detection with multi-class labels. The architecture of the network constructed is shown in Figure 15. Note that there were 4 added layers at the end with the last layer activated as a tanh function to classify the data.

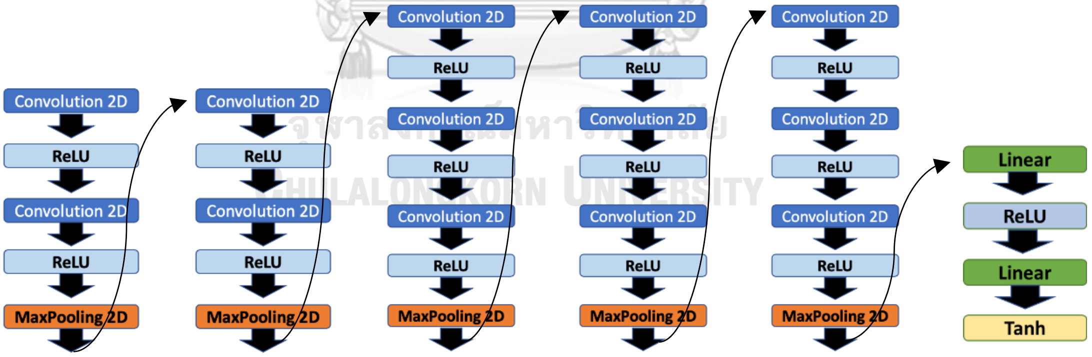


Figure 15. The architecture of Region-based CNN.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

CHAPTER 4

Experiments

4.1 Results of generating images via DCGAN

Five hundred images were successfully augmented via DCGAN for both Japanese and Australian Wagyu; Figure 16 illustrates the evolution of DCGAN images throughout the experiment. The generated images were then added to the original training dataset. Figure 17 and Figure 18 show and compare the original images and the augmented images of Japanese and Australian Wagyu, respectively.

```
Epoch:0, Step:0, D-Loss:1.231, D-Acc:20.000, G-Loss:0.671

Epoch:0, Step:50, D-Loss:0.986, D-Acc:80.000, G-Loss:10.366

………

Epoch:199, Step:150, D-Loss:0.035, D-Acc:100.000, G-Loss:3.451

Epoch:199, Step:200, D-Loss:0.021, D-Acc:100.000, G-Loss:3.558
```

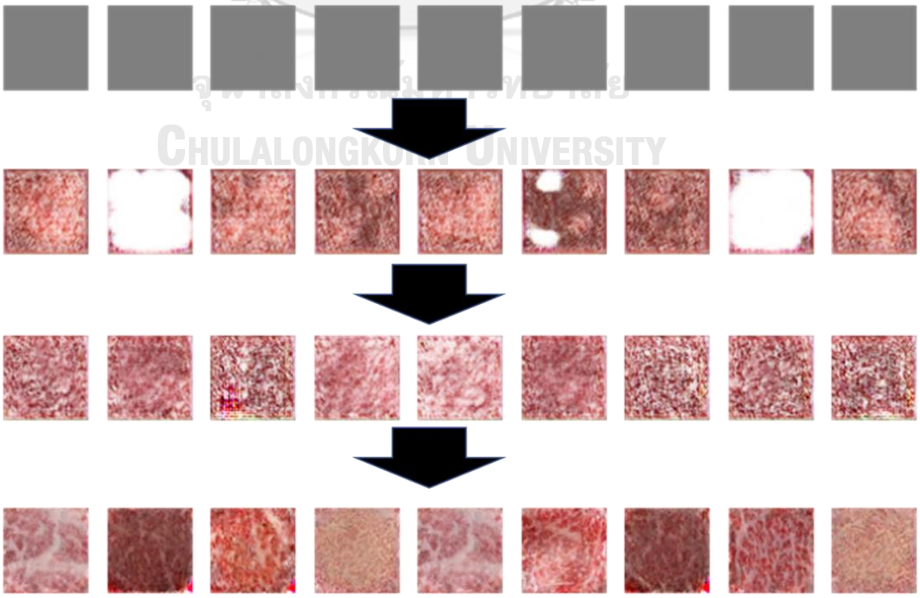


Figure 16. How DCGAN images evolved over the period trained.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

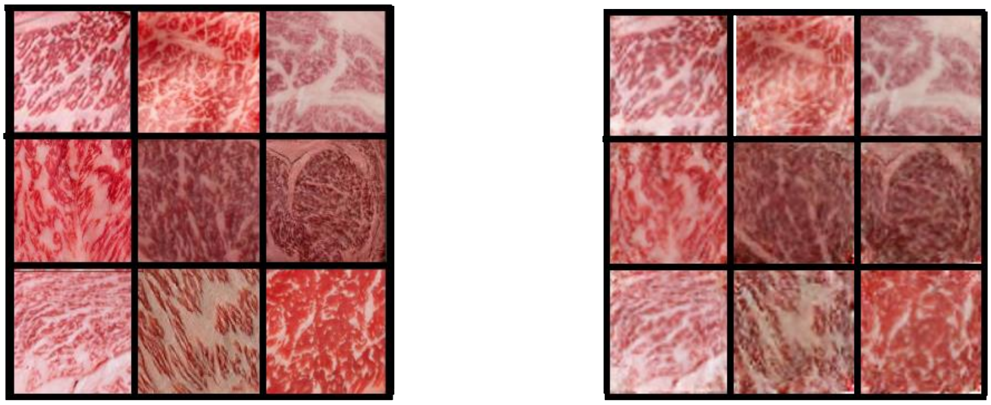


Figure 17. Original Japanese Wagyu images (Left) and Japanese Wagyu images augmented with DCGAN (Right).

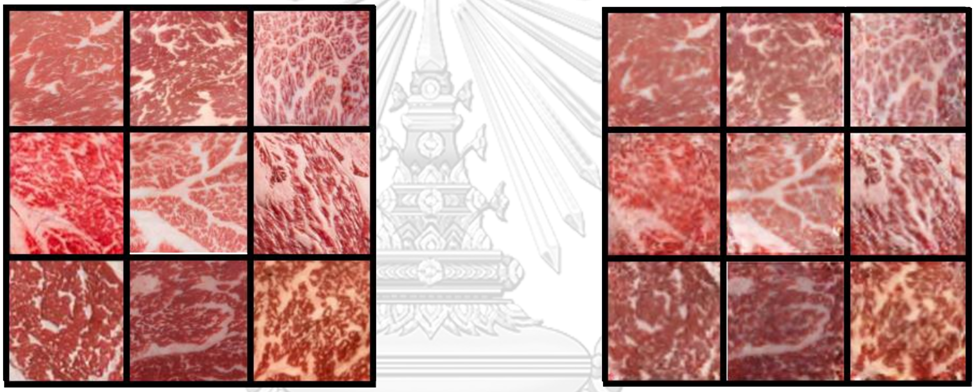


Figure 18. Original Australian Wagyu images (Left) and Australian Wagyu images augmented with DCGAN (Right).

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

4.2 Results of Phase 1

4.2.1 Results of Phase 1.1

The data were successfully trained through the created CNN, which the last layer was adjusted to 'sigmoid' since the experiment was based upon a binary classification. The first round of training was trained without the DCGAN images in the data set and achieved an accuracy via 5-fold cross validation at 91.16%

1st Fold

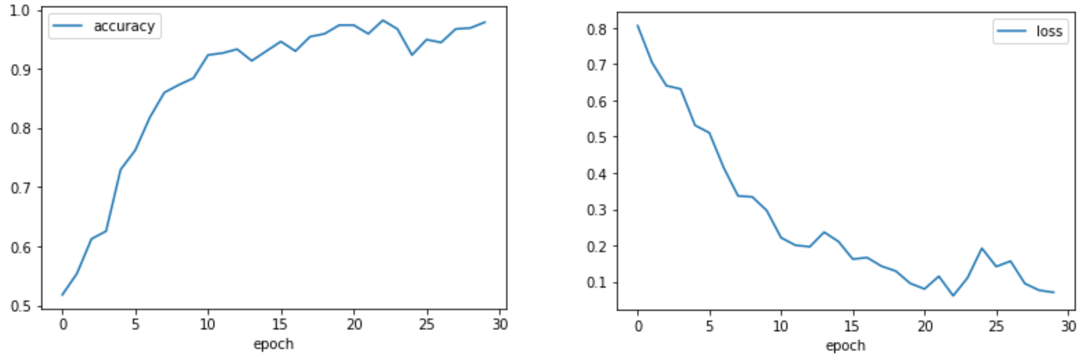


Figure 19. Accuracy (Left) and loss (Right) of the 1st fold of 5-fold cross validation of the CNN binary classification without DCGAN images.

2nd Fold

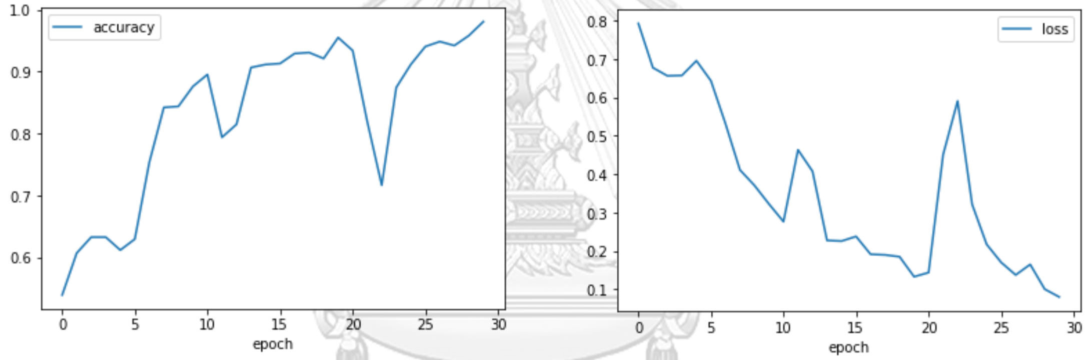


Figure 20. Accuracy (Left) and loss (Right) of the 2nd fold of 5-fold cross validation of the CNN binary classification without DCGAN images.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

3rd Fold

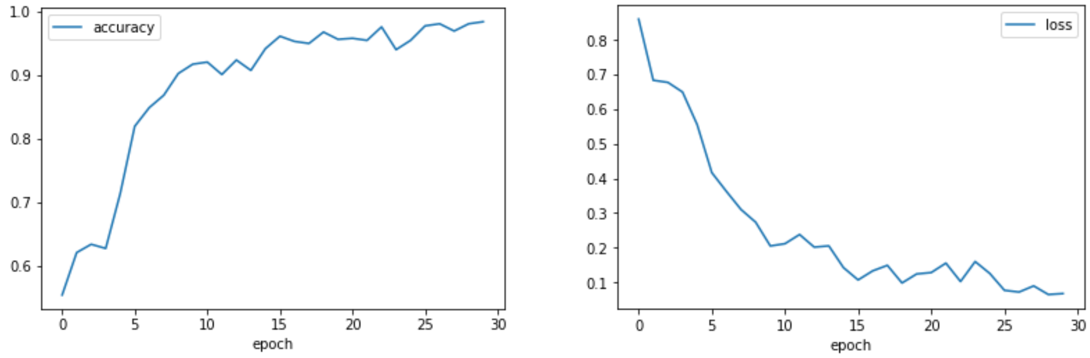


Figure 21. Accuracy (Left) and loss (Right) of the 3rd fold of 5-fold cross validation of the CNN binary classification without DCGAN images.

4th Fold

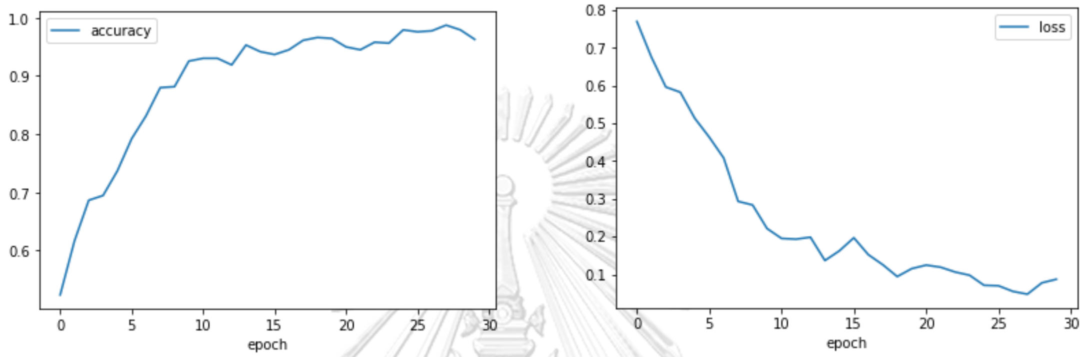


Figure 22. Accuracy (Left) and loss (Right) of the 4th fold of 5-fold cross validation of the CNN binary classification without DCGAN images.

5th Fold

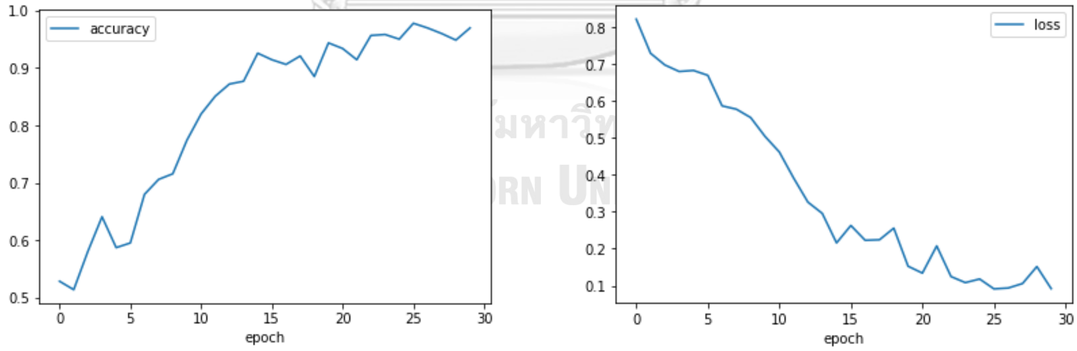


Figure 23. Accuracy (Left) and loss (Right) of the 5th fold of 5-fold cross validation of the CNN binary classification without DCGAN images.

Afterwards, the DCGAN created images were added to the dataset, and hence the second round of test was trained through the 5-fold cross validation method. The results were as expected as the CNN model with DCGAN dataset included outperformed the accuracy of the previous model (No DCGAN images) by about 1%.

The result of the training is as follows:

1st Fold

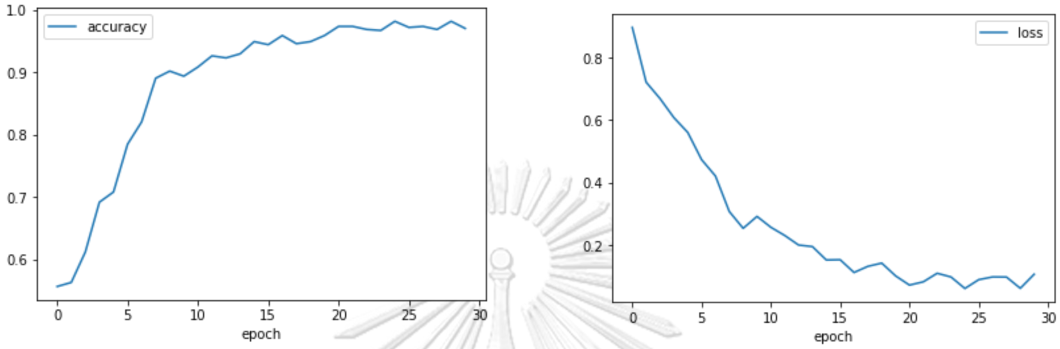


Figure 24. Accuracy (Left) and loss (Right) of the 1st fold of 5-fold cross validation of the CNN binary classification with DCGAN images.

2nd Fold

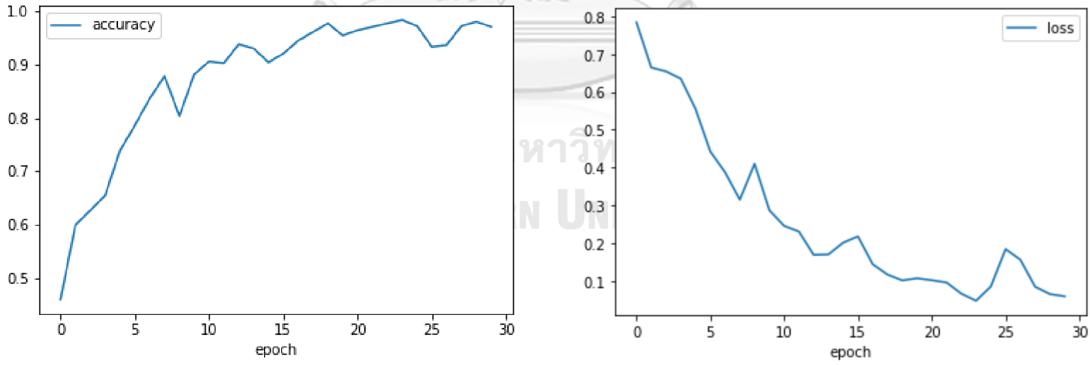


Figure 25. Accuracy (Left) and loss (Right) of the 2nd fold of 5-fold cross validation of the CNN binary classification with DCGAN images.

3rd Fold

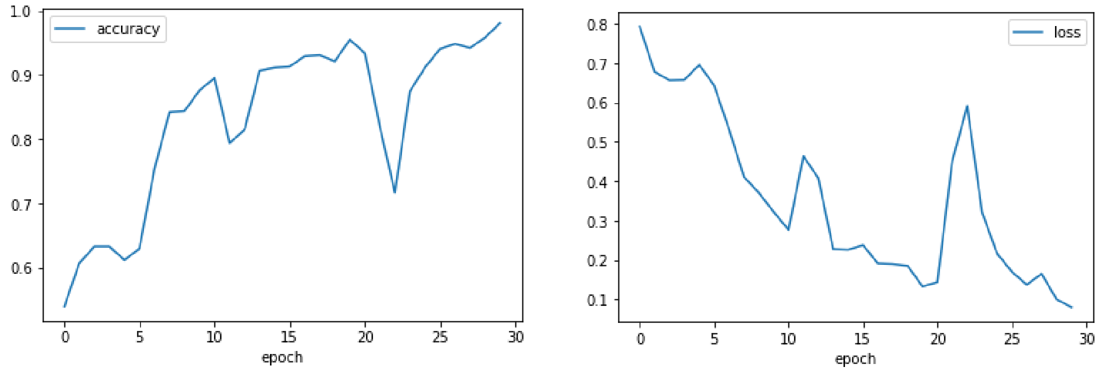


Figure 26. Accuracy (Left) and loss (Right) of the 3rd fold of 5-fold cross validation of the CNN binary classification with DCGAN images.

4th Fold

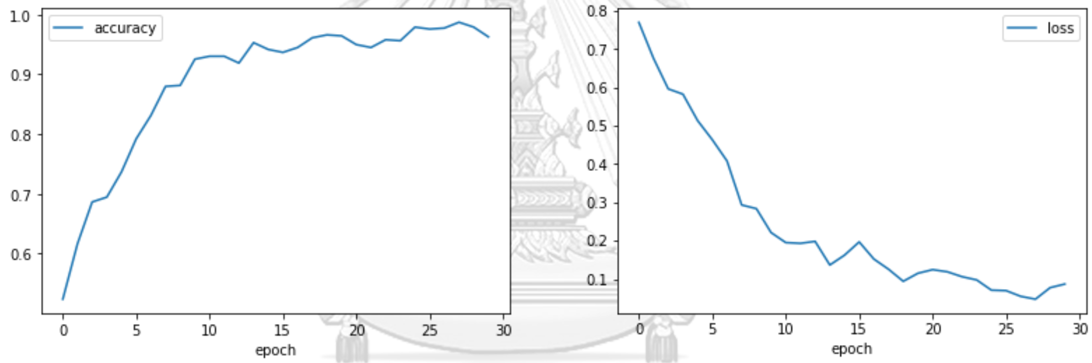


Figure 27. Accuracy (Left) and loss (Right) of the 4th fold of 5-fold cross validation of the CNN binary classification with DCGAN images

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

5th Fold

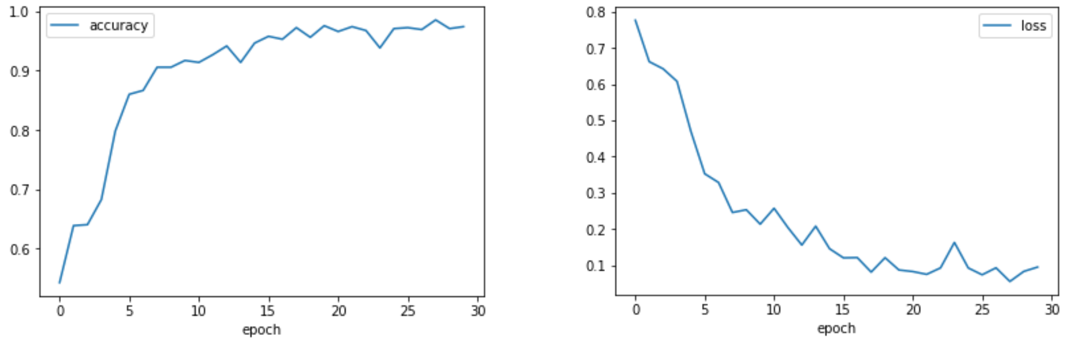


Figure 28. Accuracy (Left) and loss (Right) of the 5th fold of 5-fold cross validation of the CNN binary classification with DCGAN images.

4.2.2 Results of Phase 1.2

After the initial experiment was proved successful, US Wagyu images were inserted into the DCGAN model in order to repeat the process and turn the binary-classification into a multi one. The first time round the CNN model would be trained on the dataset without DCGAN images and achieved the accuracy of 73.2% though 5 -fold cross validation.

The result of this training is as follows:

1st Fold

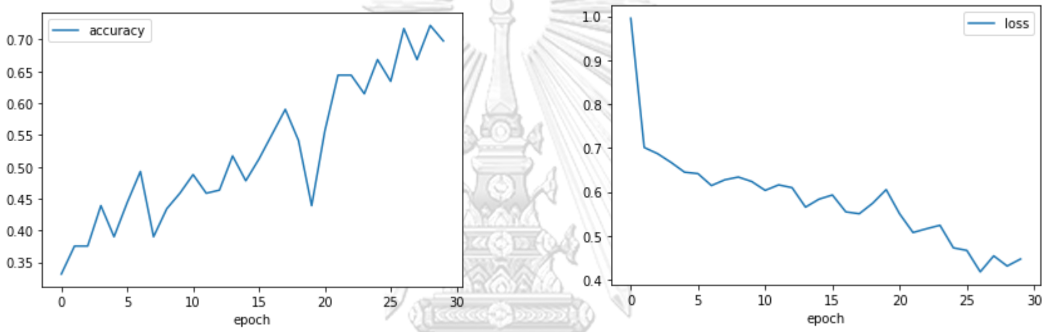


Figure 29. Accuracy (Left) and loss (Right) of the 1st fold of 5-fold cross validation of the CNN multi-classification without DCGAN images.

2nd Fold

จุฬาลงกรณ์มหาวิทยาลัย

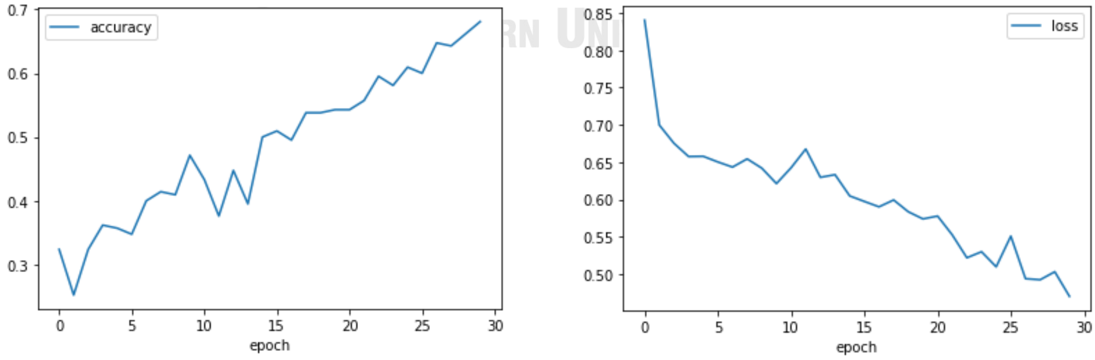


Figure 30. Accuracy (Left) and loss (Right) of the 2nd fold of 5-fold cross validation of the CNN multi-classification without DCGAN images.

3rd Fold

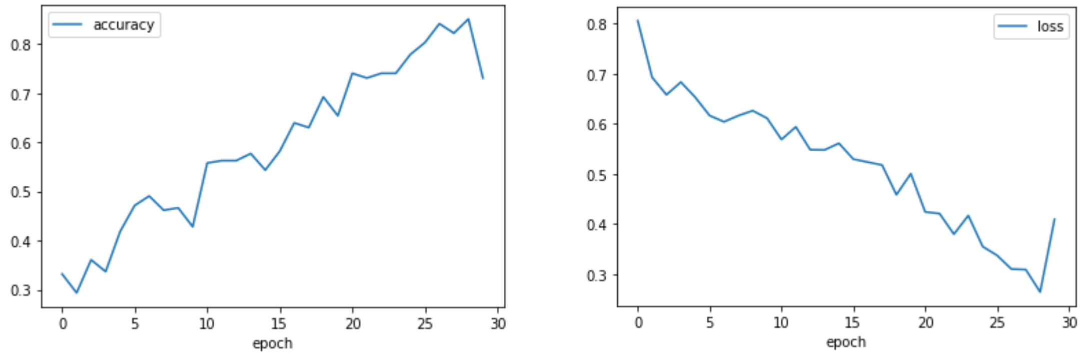


Figure 31. Accuracy (Left) and loss (Right) of the 3rd fold of 5-fold cross validation of the CNN multi-classification without DCGAN images.

4th Fold

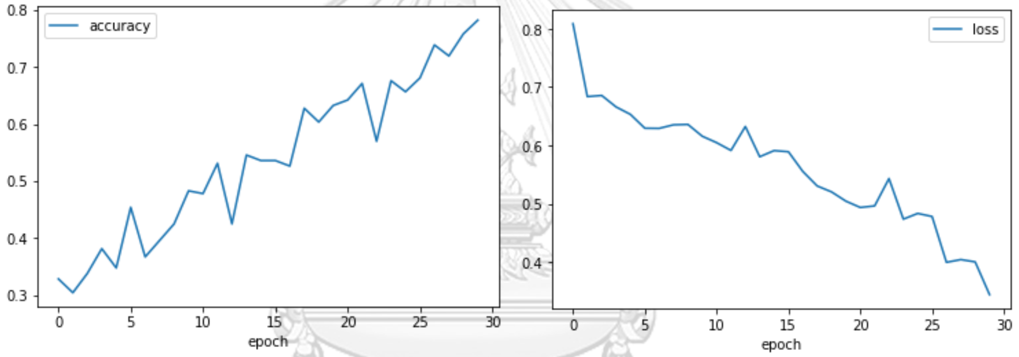


Figure 32. Accuracy (Left) and loss (Right) of the 4th fold of 5-fold cross validation of the CNN multi-classification without DCGAN images.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

5th Fold

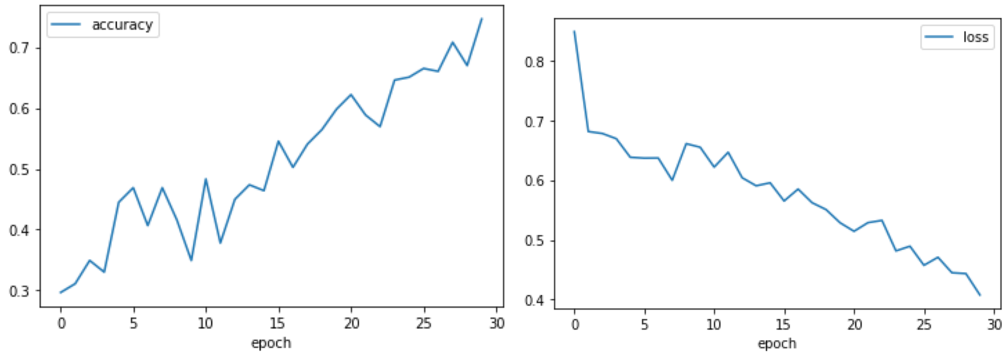


Figure 33. Accuracy (Left) and loss (Right) of the 5th fold of 5-fold cross validation of the CNN multi-classification without DCGAN images.

After this, the DCGAN images were added to the data set. As expected, the accuracy running through 5-fold cross validation method of the model was more than the accuracy of the previous one (No DCGAN images). The accuracy of this model was at 77.8%.

The result of the training is as follows:

1st Fold

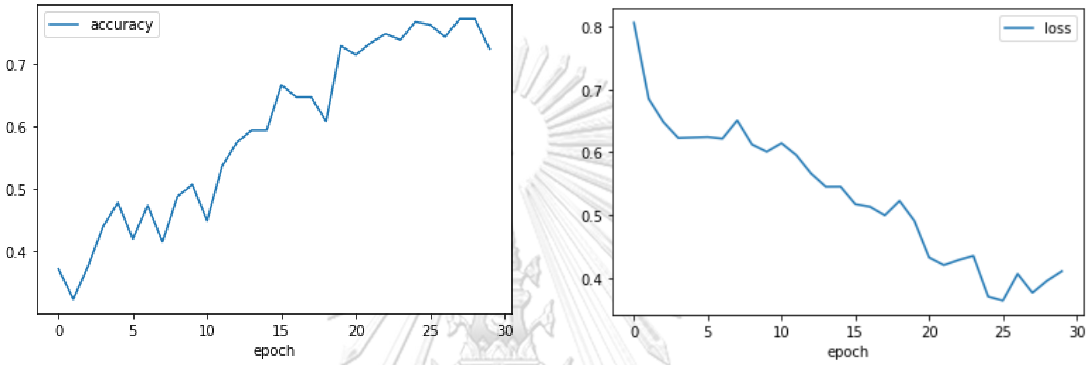


Figure 34. Accuracy (Left) and loss (Right) of the 1st fold of 5-fold cross validation of the CNN multi-classification with DCGAN images.

2nd Fold

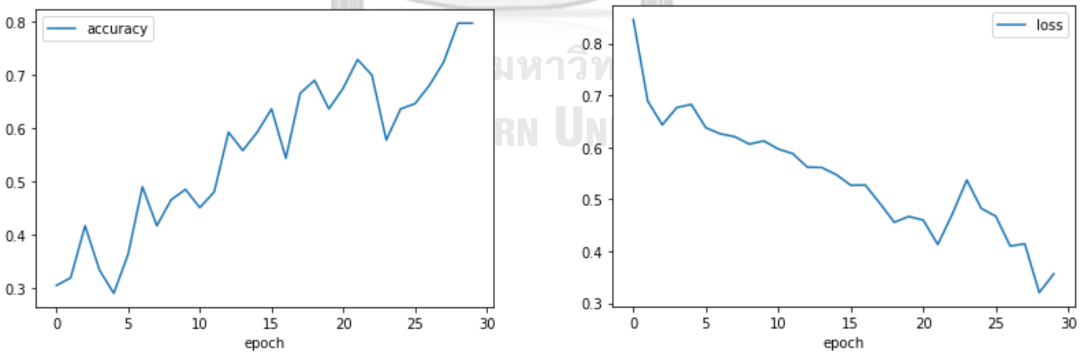


Figure 35. Accuracy (Left) and loss (Right) of the 2nd fold of 5-fold cross validation of the CNN multi-classification with DCGAN images.

3rd Fold

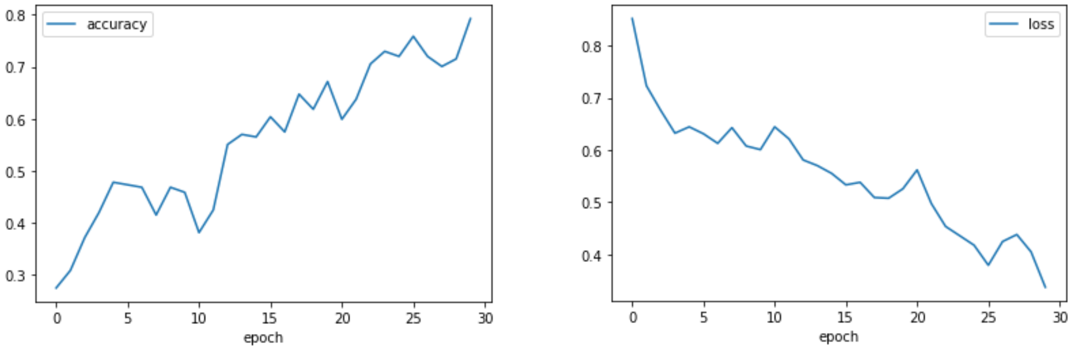


Figure 36. Accuracy (Left) and loss (Right) of the 3rd fold of 5-fold cross validation of the CNN multi-classification with DCGAN images.

4th Fold

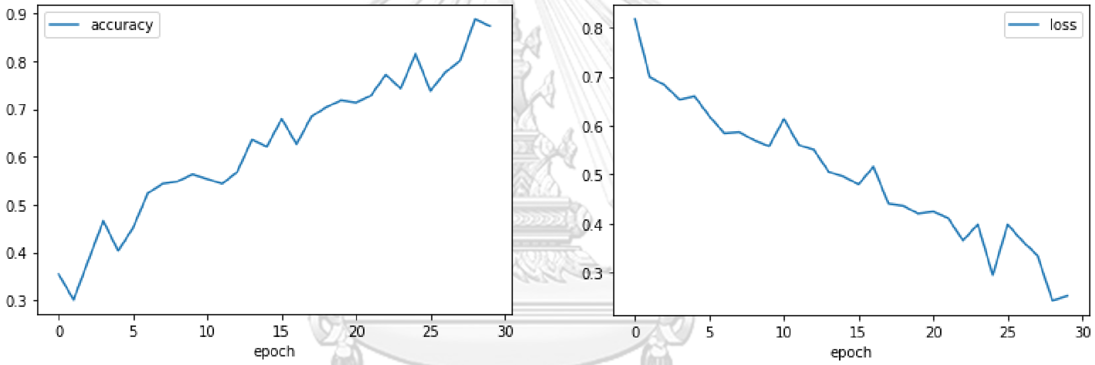


Figure 37. Accuracy (Left) and loss (Right) of the 4th fold of 5-fold cross validation of the CNN multi-classification with DCGAN images.

5th Fold

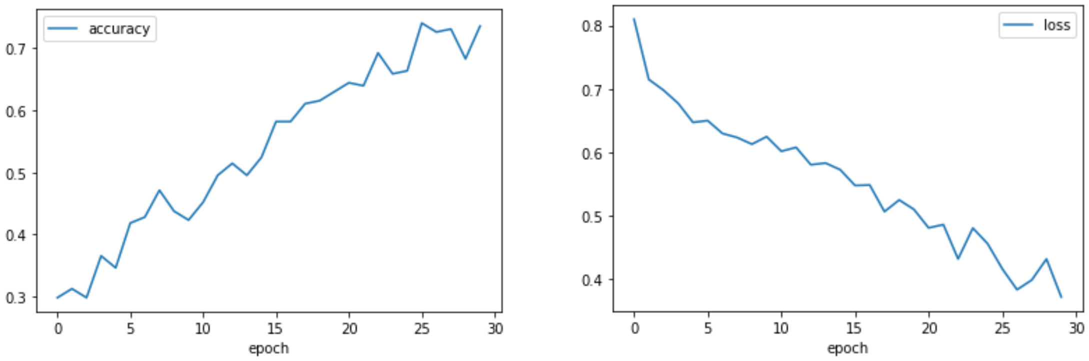


Figure 38. Accuracy (Left) and loss (Right) of the 5th fold of 5-fold cross validation of the CNN multi-classification with DCGAN images.

4.3 Results of Phase 2

4.3.1 Results of Phase 2.1

For this latter stage of the research, Region-based CNN was implemented in order to perform the object detection and also to tackle the problem of multi-classification and Out of Distribution detection (OOD). In order to perform such tasks successfully, NVIDIA CUDA was triggered and enabled to allow GPUs processing power.

The data were successfully trained through the Region Based-CNN model with its accuracy shown by K-Fold method (5-Folds). The first round of training in Phase 2.1 was trained by using the original data set “only”, by which, the images generated by DCGAN were left out.

The result of this training is as follows:

1st Fold

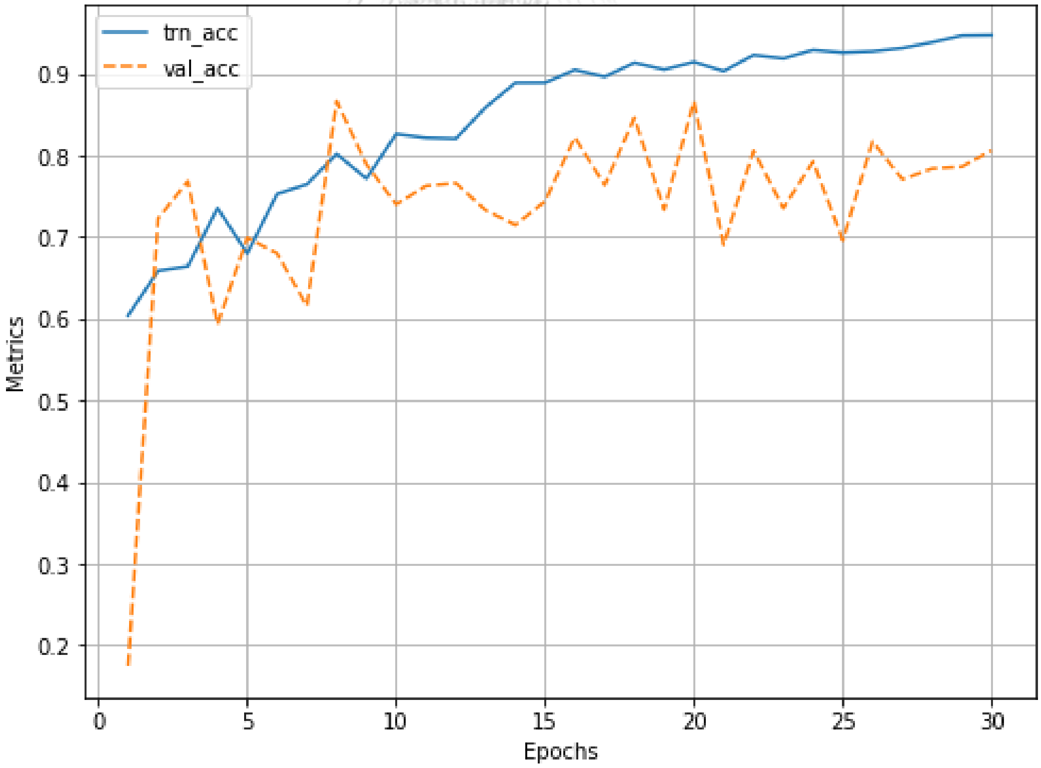


Figure 39. Accuracy of the 1st fold of 5-fold cross validation of the R-CNN binary classification without DCGAN images.

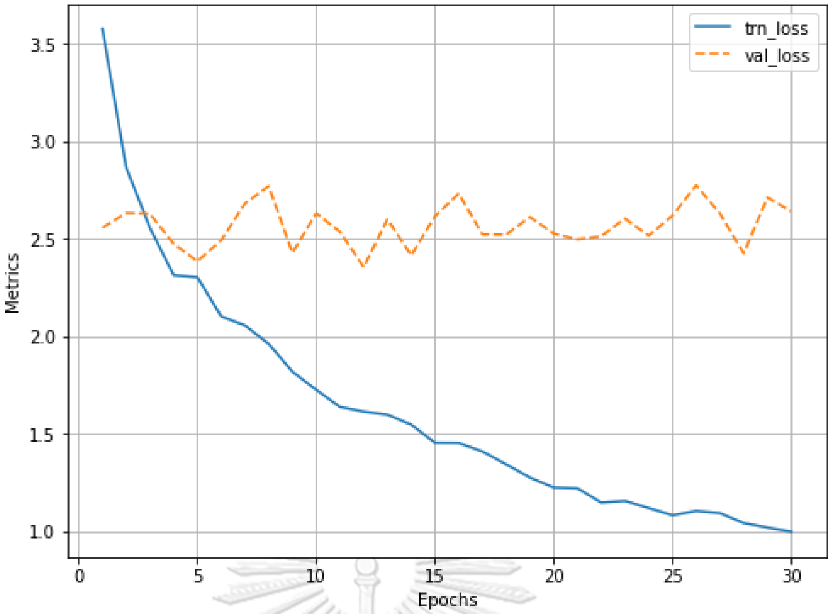


Figure 40. Loss of the 1st fold of 5-fold cross validation of the R-CNN binary classification without DCGAN images.

2nd Fold

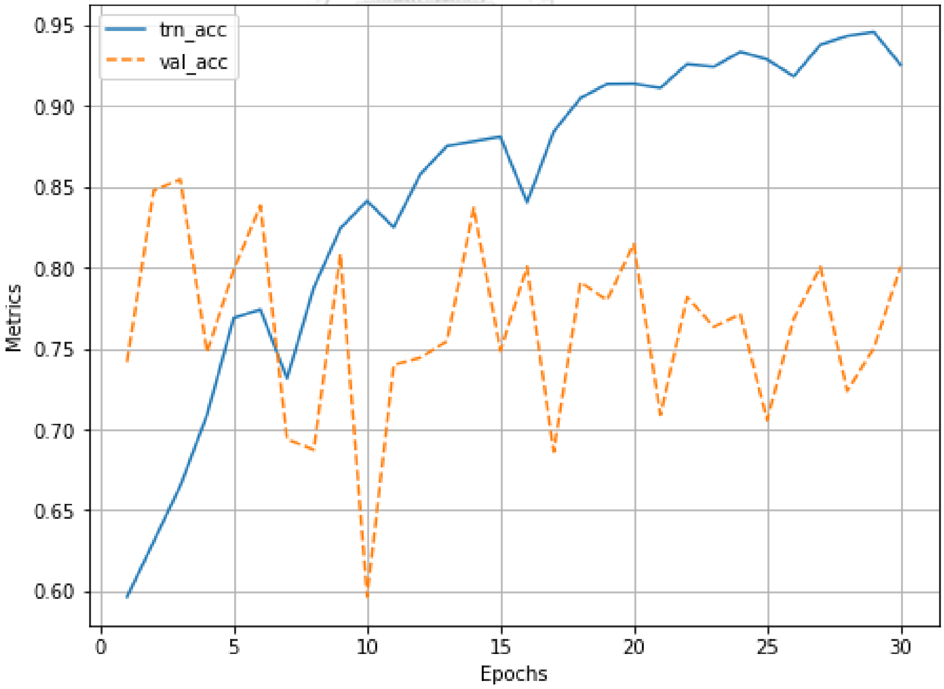


Figure 41. Accuracy of the 2nd fold of 5-fold cross validation of the R-CNN binary classification without DCGAN images.

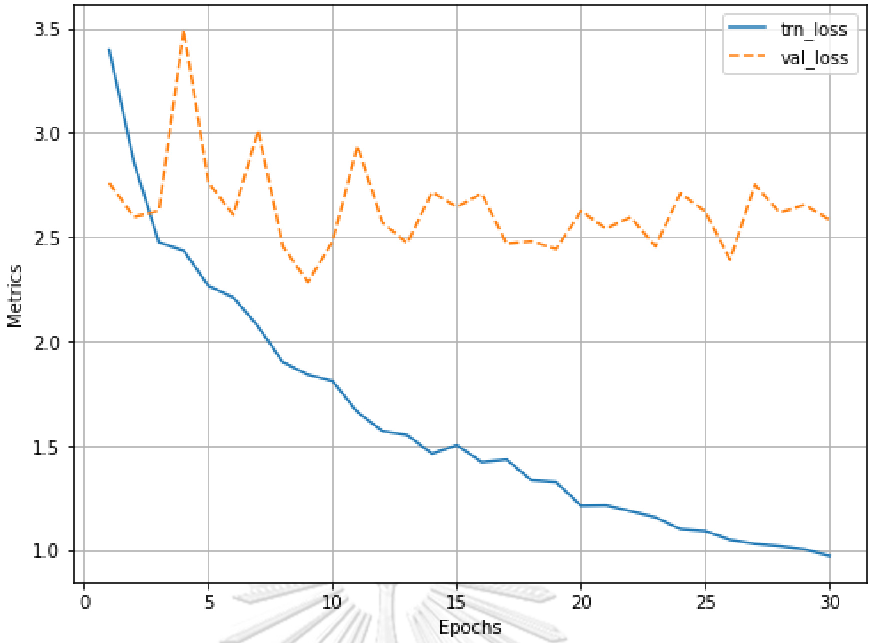


Figure 42. Loss of the 2nd fold of 5-fold cross validation of the R-CNN binary classification without DCGAN images.

3rd Fold

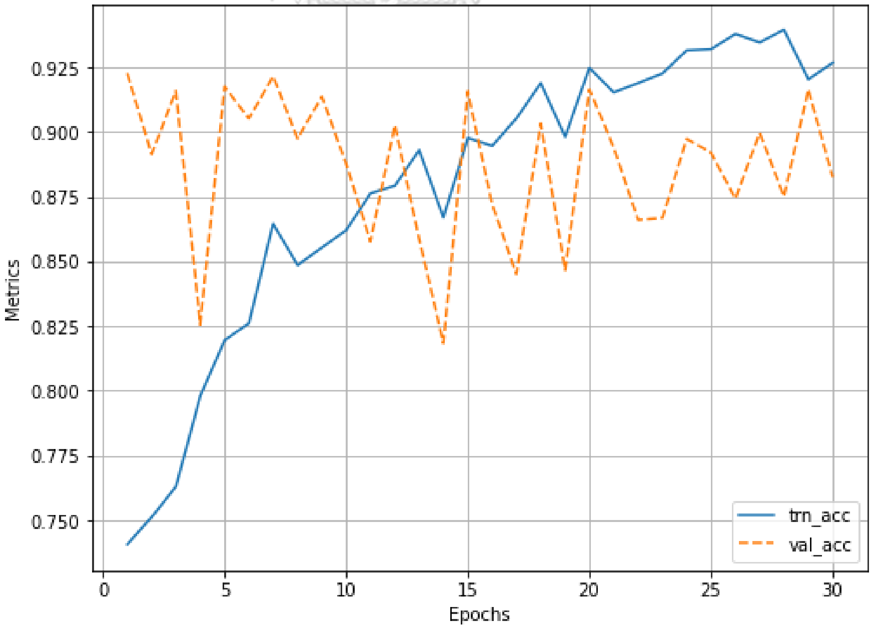


Figure 43. Accuracy of the 3rd fold of 5-fold cross validation of the R-CNN binary classification without DCGAN images.

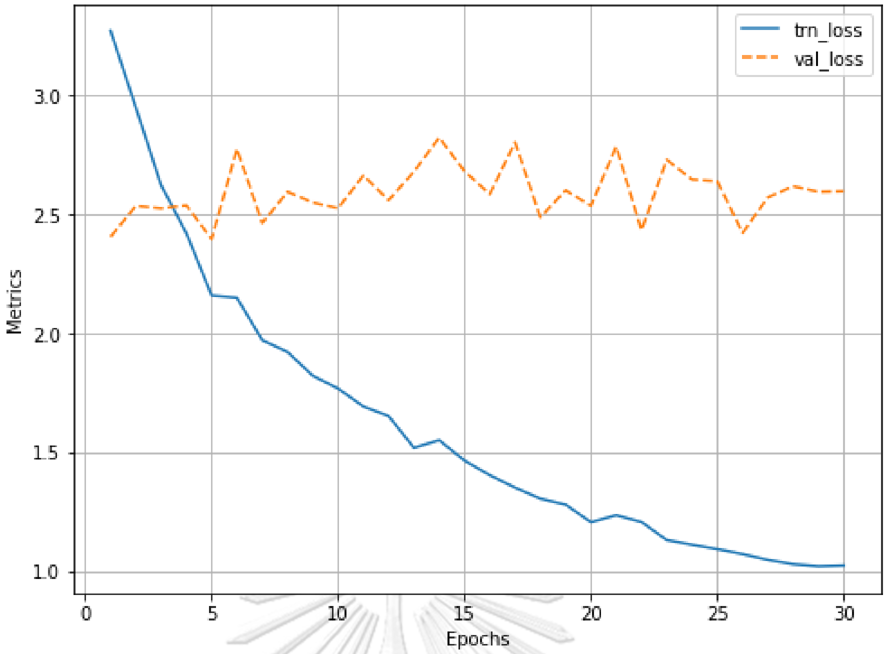


Figure 44. Loss of the 3rd fold of 5-fold cross validation of the R-CNN binary classification without DCGAN images.

4th Fold

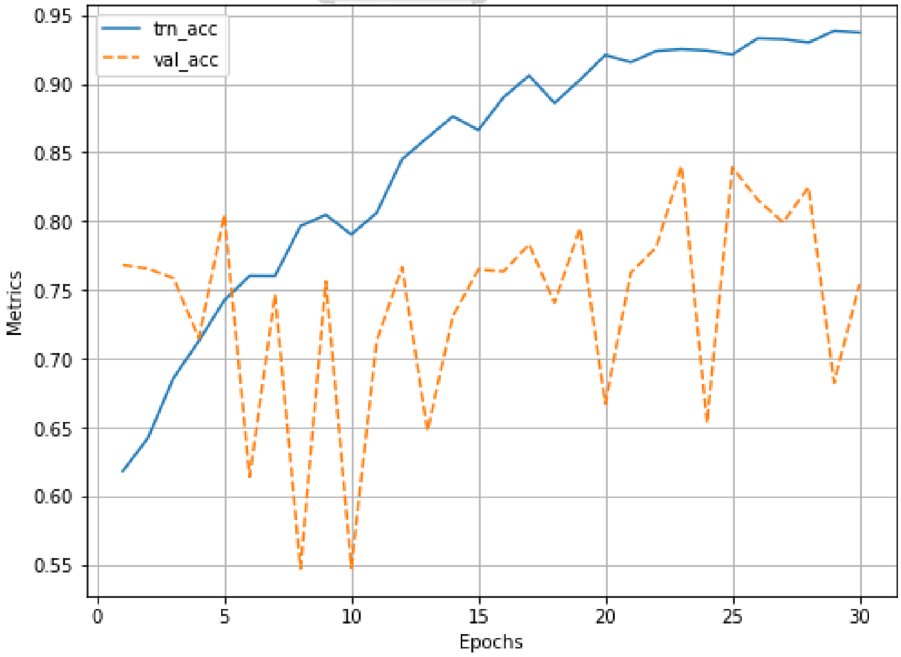


Figure 45. Accuracy of the 4th fold of 5-fold cross validation of the R-CNN binary classification without DCGAN images.

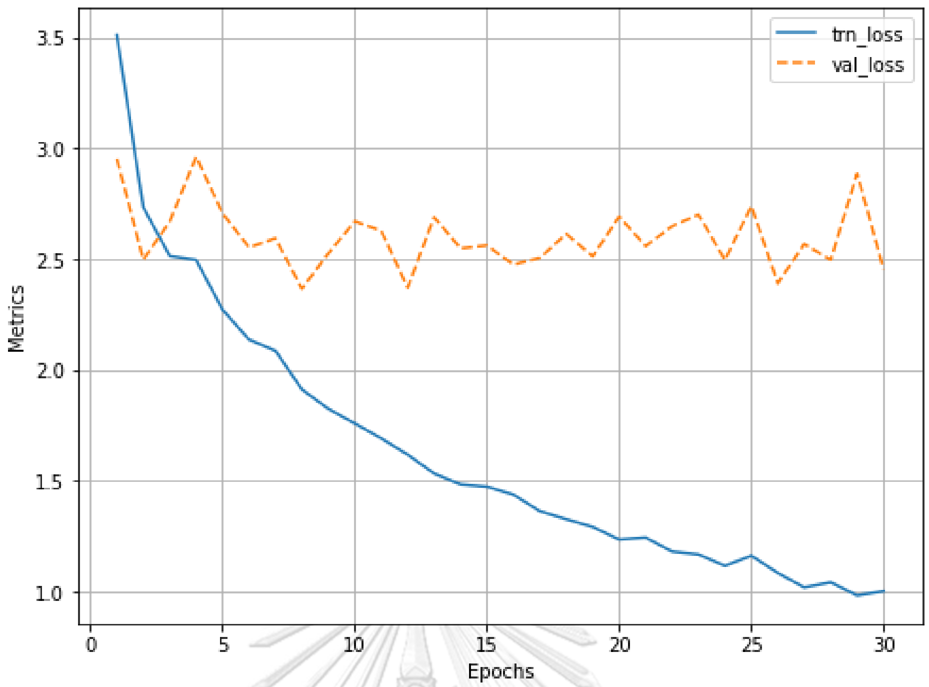


Figure 46. Loss of the 4th fold of 5-fold cross validation of the R-CNN binary classification without DCGAN images.

5th Fold

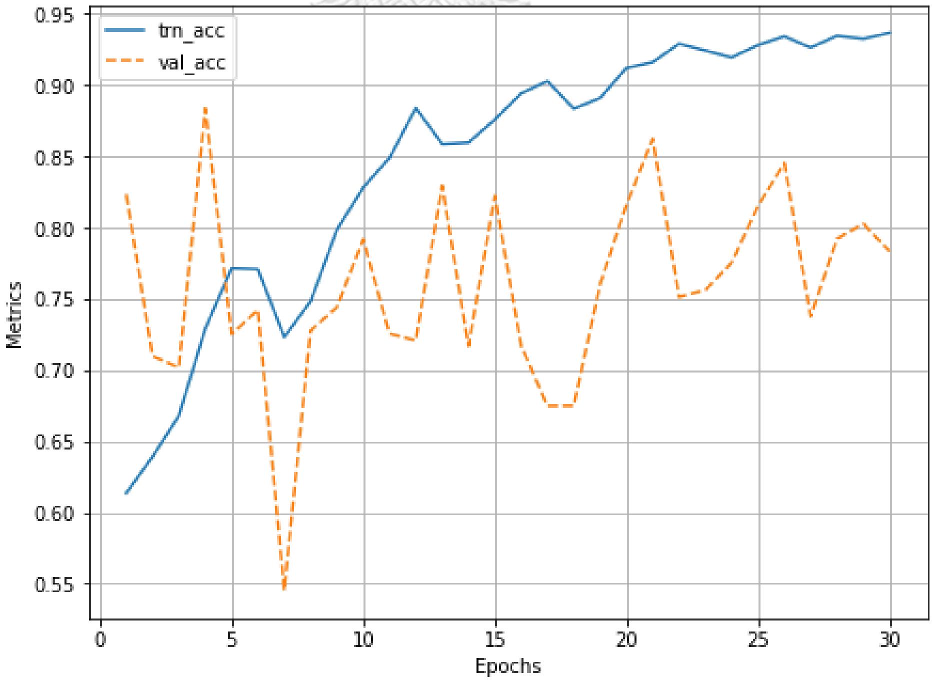


Figure 47. Accuracy of the 5th fold of 5-fold cross validation of the R-CNN binary classification without DCGAN images.

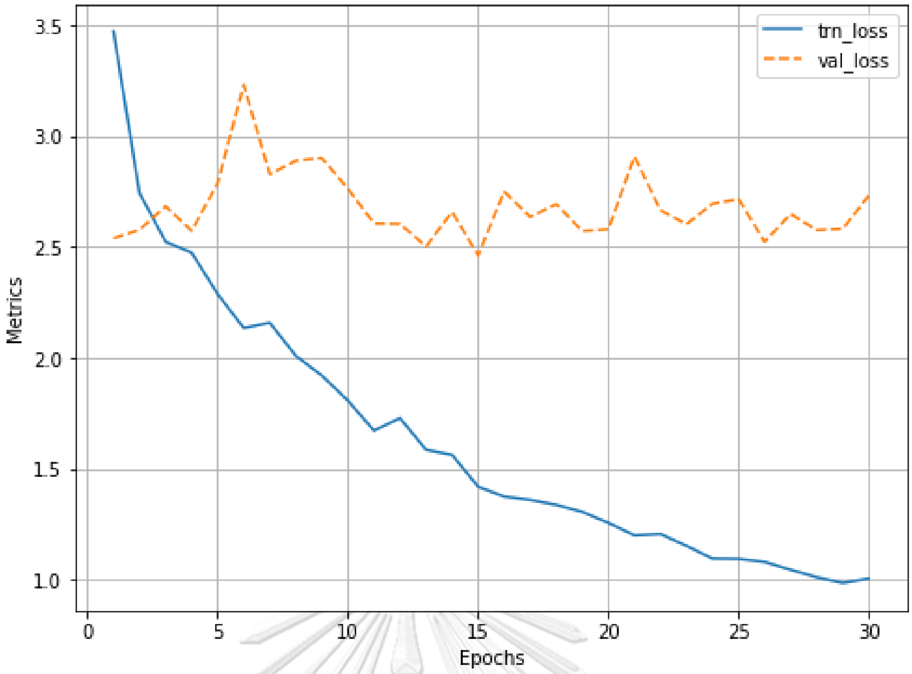


Figure 48. Loss of the 5th fold of 5-fold cross validation of the R-CNN binary classification without DCGAN images.

Afterwards, the images created by the DCGAN architecture were added into the model in order to compare the difference between having and 'not' having artificial images created by DCGAN.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

The result of this training is as follows:

1st Fold

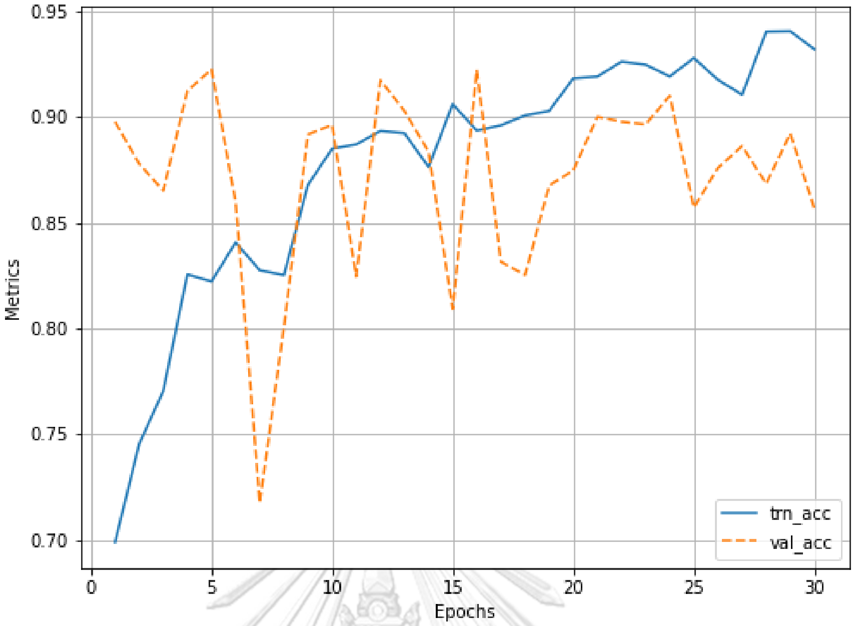


Figure 49. Accuracy of the 1st fold of 5-fold cross validation of the R-CNN binary classification with DCGAN images.

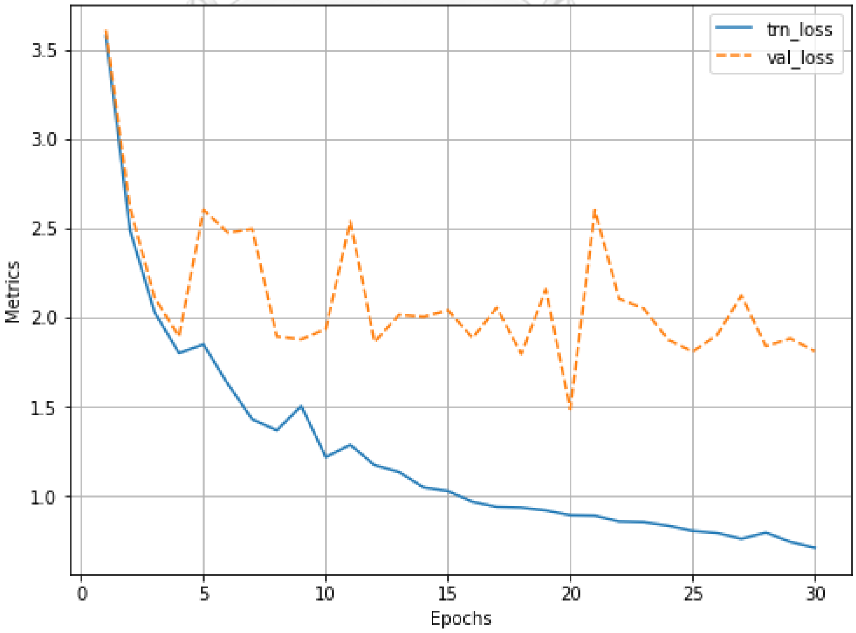


Figure 50. Loss of the 1st fold of 5-fold cross validation of the R-CNN binary classification with DCGAN images.

2nd Fold

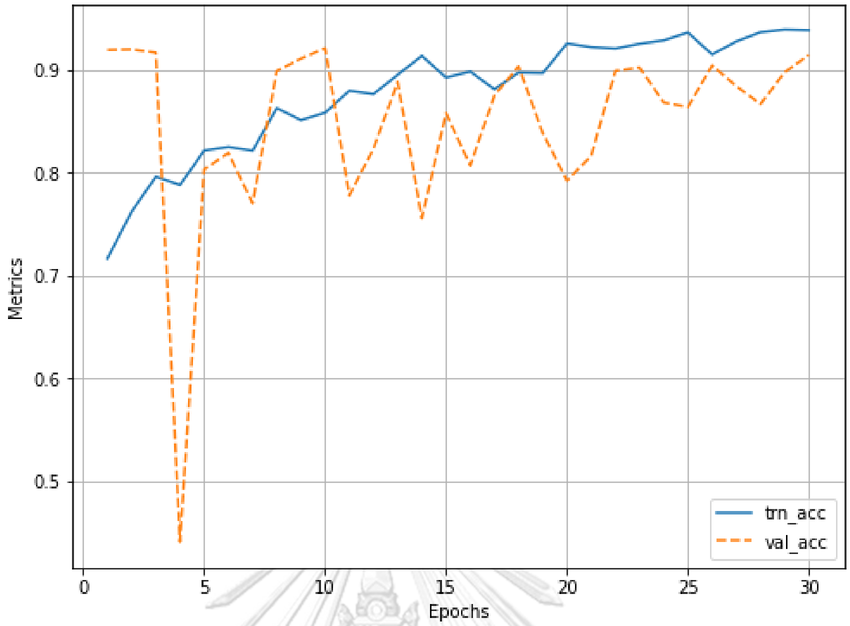


Figure 51. Accuracy of the 2nd fold of 5-fold cross validation of the R-CNN binary classification with DCGAN images.

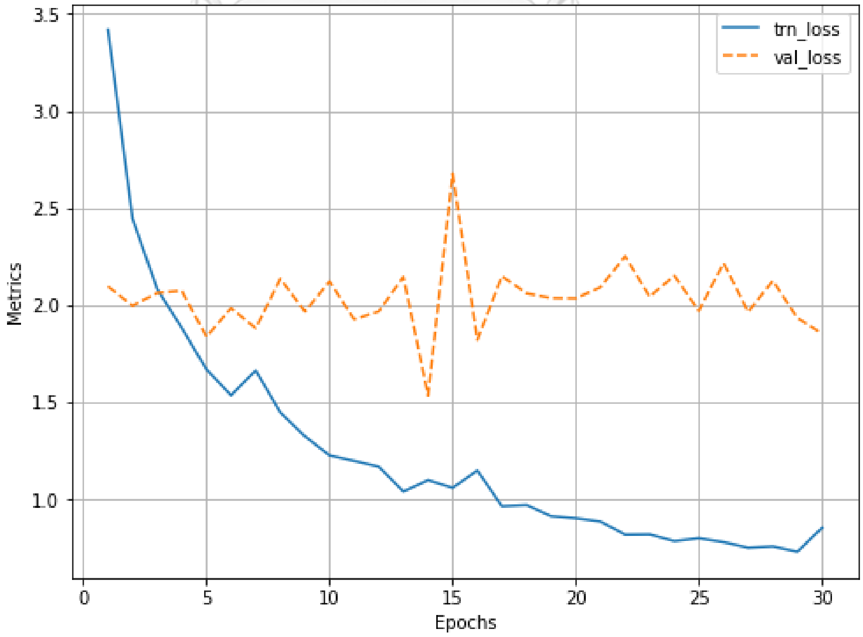


Figure 52. Loss of the 2nd fold of 5-fold cross validation of the R-CNN binary classification with DCGAN images.

3rd Fold

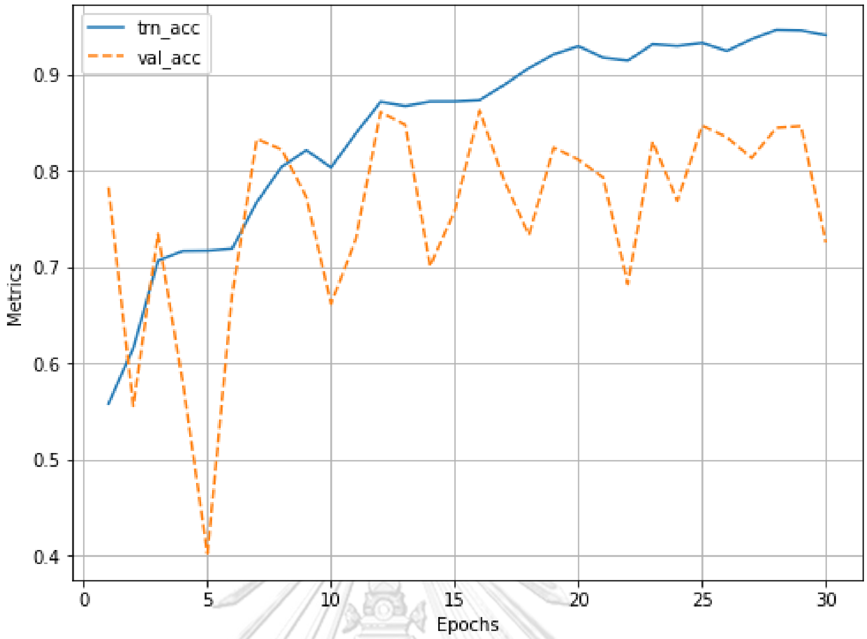


Figure 53. Accuracy of the 3rd fold of 5-fold cross validation of the R-CNN binary classification with DCGAN images.

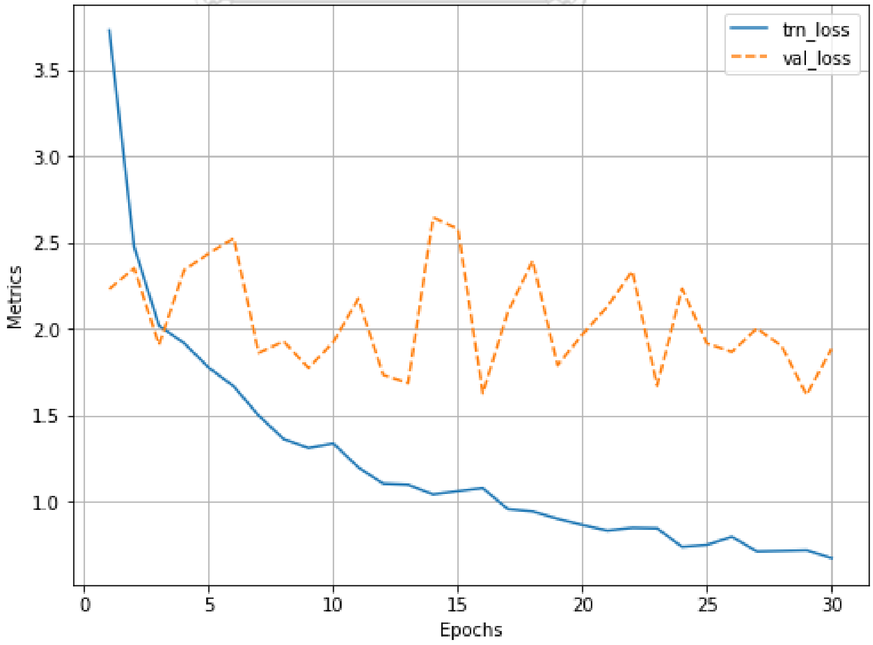


Figure 54. Loss of the 3rd fold of 5-fold cross validation of the R-CNN binary classification with DCGAN images.

4th Fold

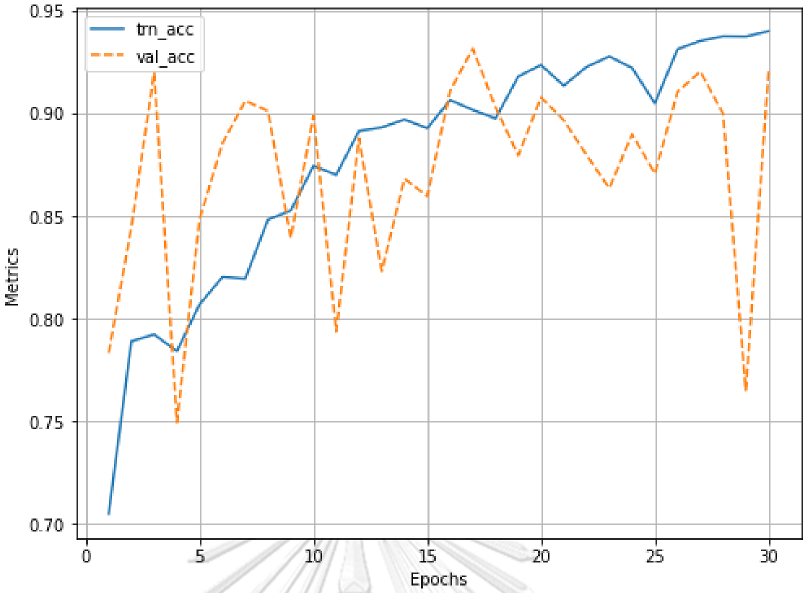


Figure 55. Accuracy of the 4th fold of 5-fold cross validation of the R-CNN binary classification with DCGAN images.

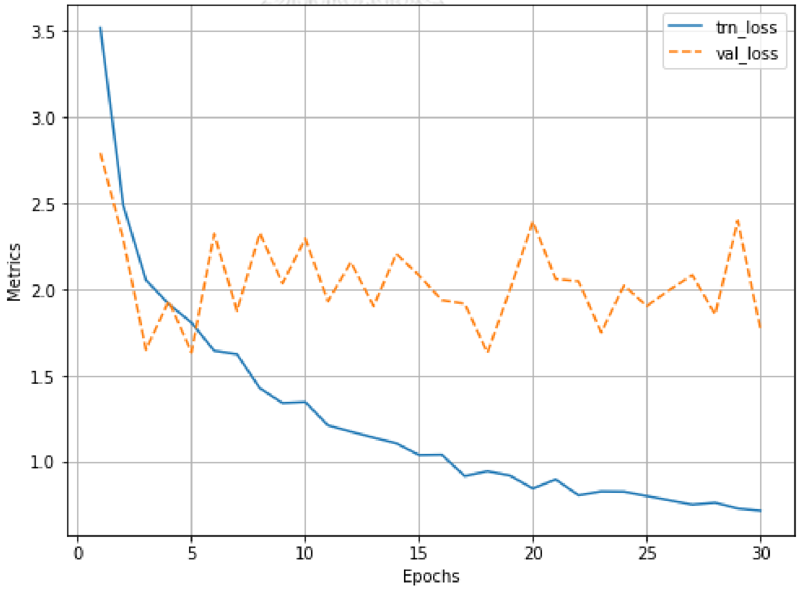


Figure 56. Loss of the 4th fold of 5-fold cross validation of the R-CNN binary classification with DCGAN images.

5th Fold

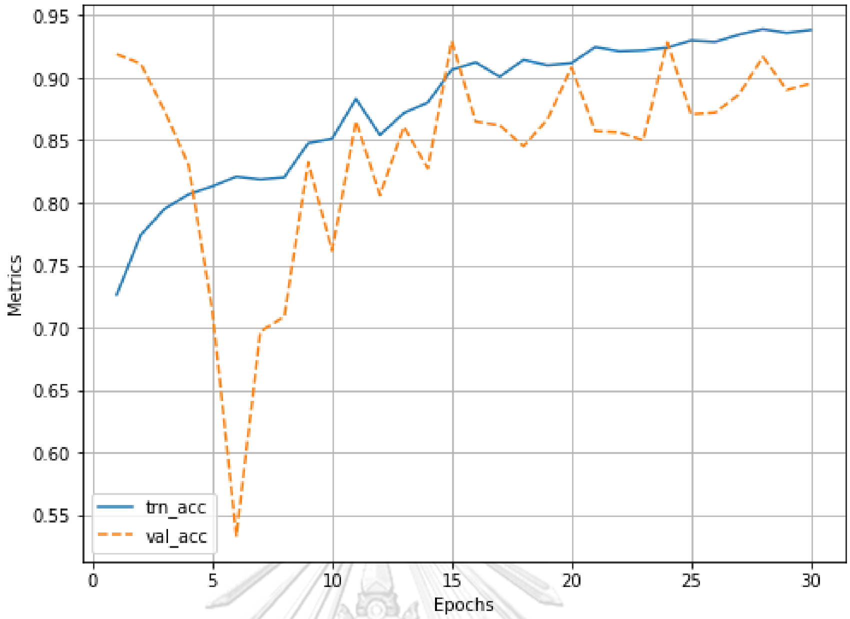


Figure 57. Accuracy of the 5th fold of 5-fold cross validation of the R-CNN binary classification with DCGAN images.

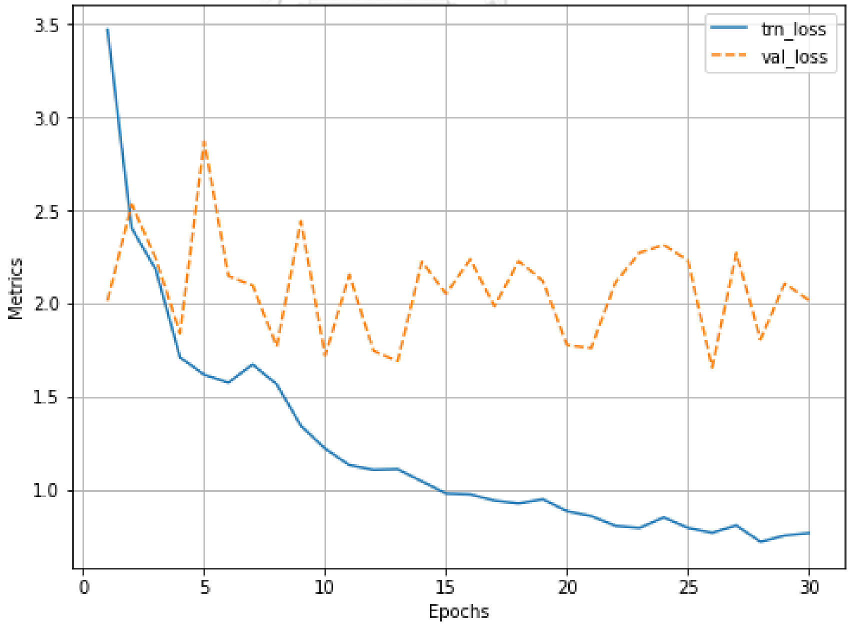


Figure 58. Loss of the 5th fold of 5-fold cross validation of the R-CNN binary classification with DCGAN images.

After this, the model performance of R-CNN with DCGAN images included was evaluated on test dataset containing Japanese and Australian Wagyu. Figure 59 and Figure 60 illustrate the results of successful detection of the patterns of marbled fat layers of Japanese Wagyu.

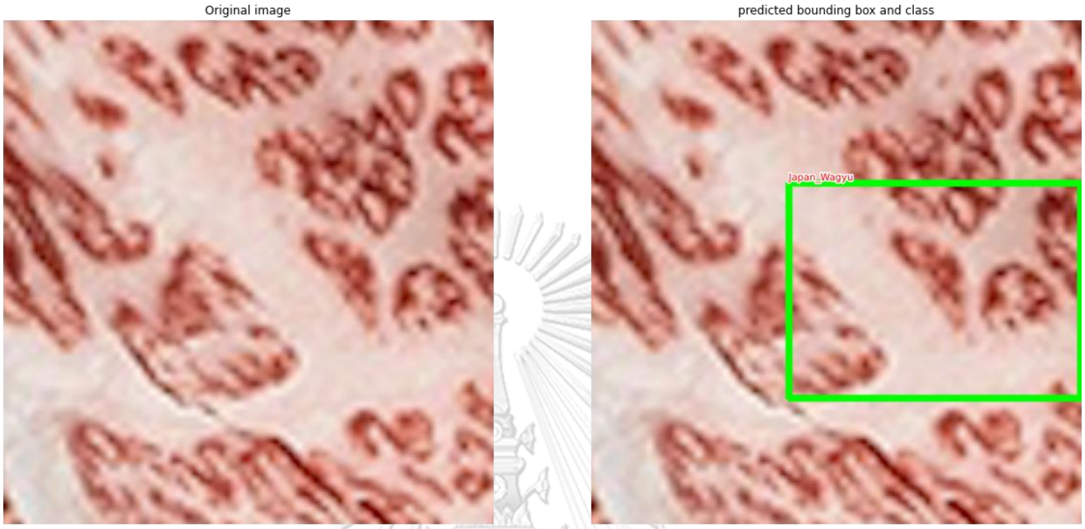


Figure 59 Example 1 illustrating how R-CNN could recognize the marbled fat layers of Japanese Wagyu.

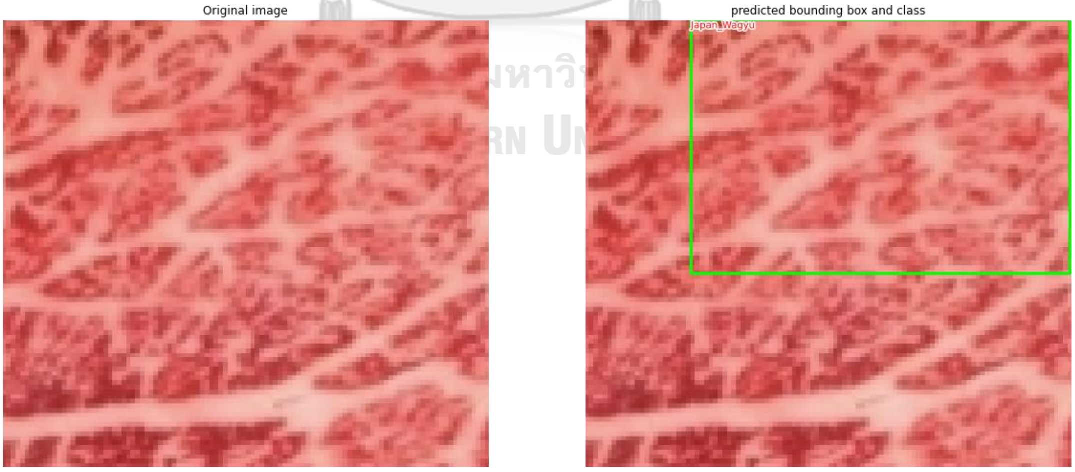


Figure 60 Example 2 illustrating how R-CNN could recognize the marbled fat layers of Japanese Wagyu.

Figure 61 and Figure 62 shows the results of successful detection of the patterns of marbled fat layers of Australian Wagyu beef.

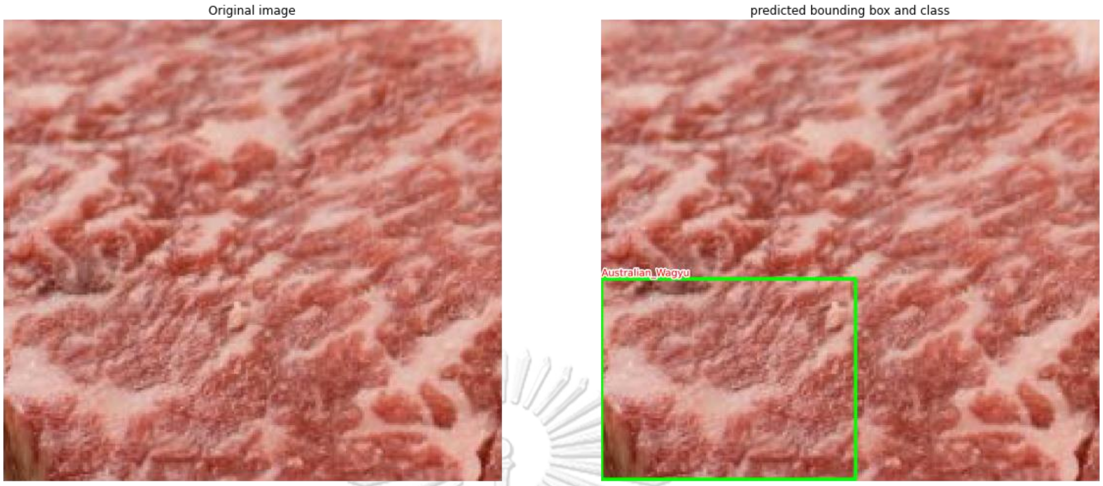


Figure 61. Example 1 illustrating how R-CNN could recognize the marbled fat layers of Australian Wagyu.

We also tested the model using the US Wagyu. As a result, the R-CNN could not recognize the patterns of Japanese Wagyu nor the Australian, therefore giving a result as “No object found” as shown in Figure 63.

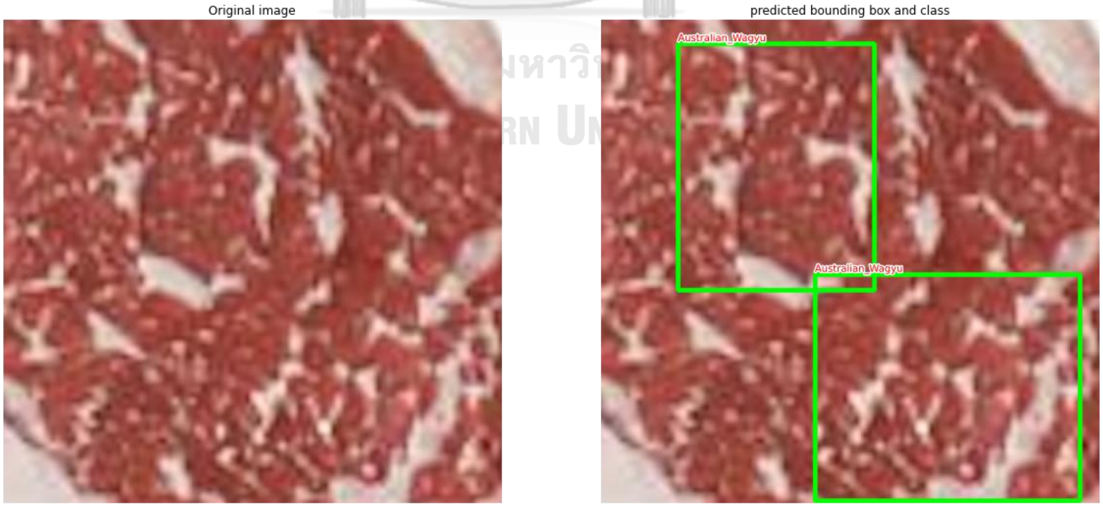


Figure 62. Example 2 illustrating how R-CNN could recognize the marbled fat layers of Australian Wagyu.

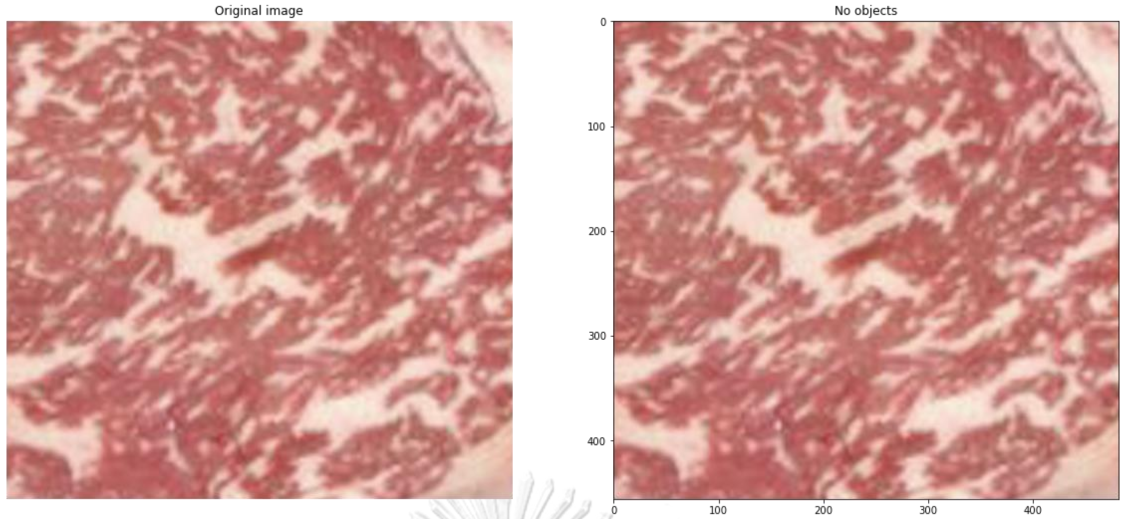


Figure 63. Illustrating how the machine could not recognize the pattern of the marbled fat layers of the US Wagyu beef.

4.3.2 Results of Phase 2.2

In this stage, the R-CNN was trained for multi-class object detection by adding the US Wagyu beef. The model was successfully trained through the same R-CNN architecture with enabling NVIDIA CUDA. For the first round of training, the model achieved an accuracy of 74.3% through 5-fold cross validation, the dataset of this round was limited to original images only (No DCGAN images).

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

1st Fold

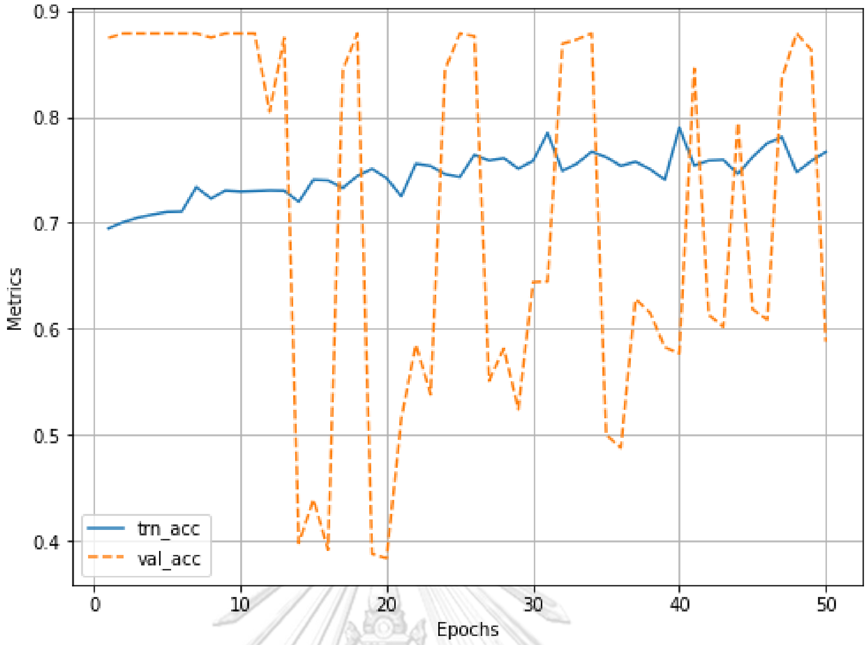


Figure 64. Accuracy of the 1st fold of 5-fold cross validation of the R-CNN multi-classification without DCGAN images.

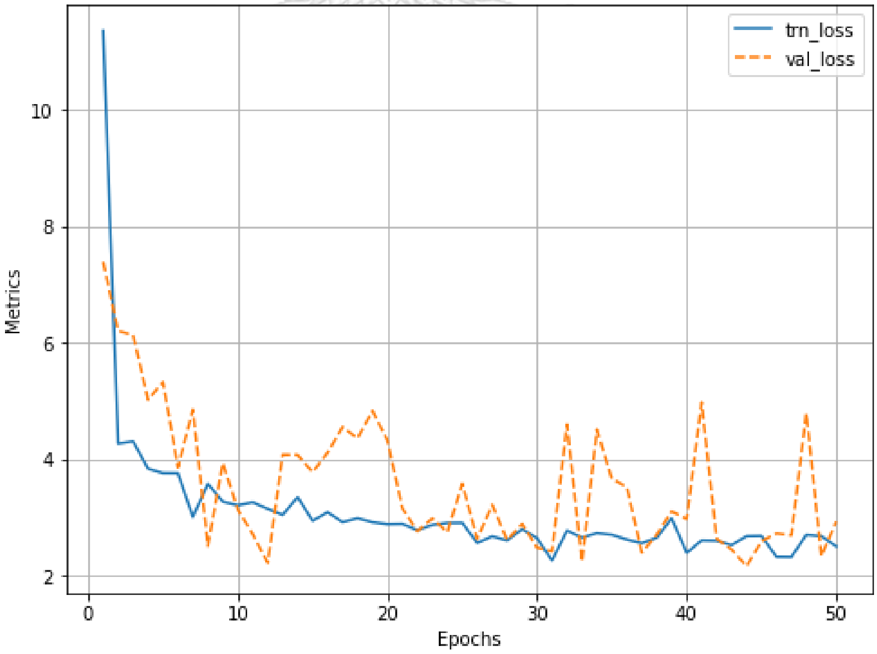


Figure 65. Loss of the 1st fold of 5-fold cross validation of the R-CNN multi-classification without DCGAN images.

2nd Fold

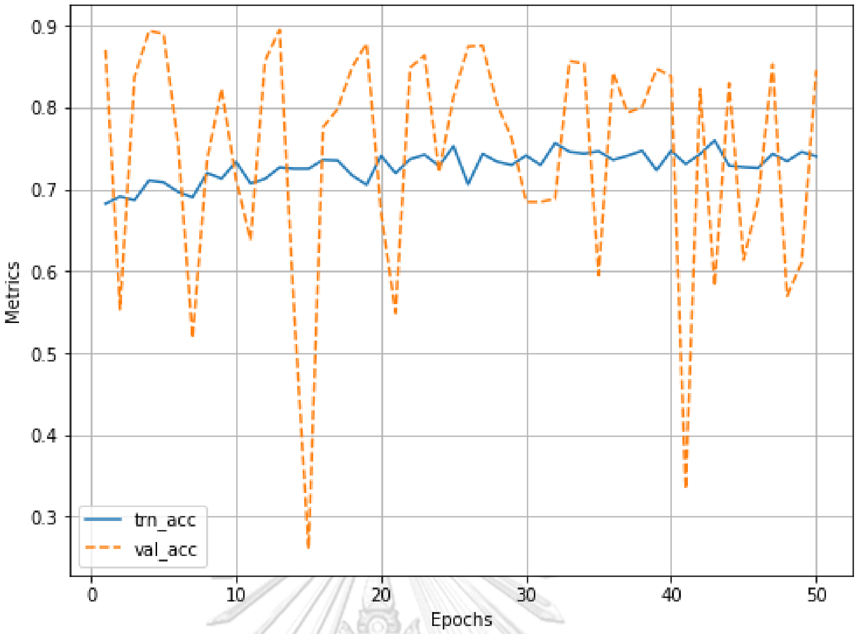


Figure 66. Accuracy of the 2nd fold of 5-fold cross validation of the R-CNN multi-classification without DCGAN images.

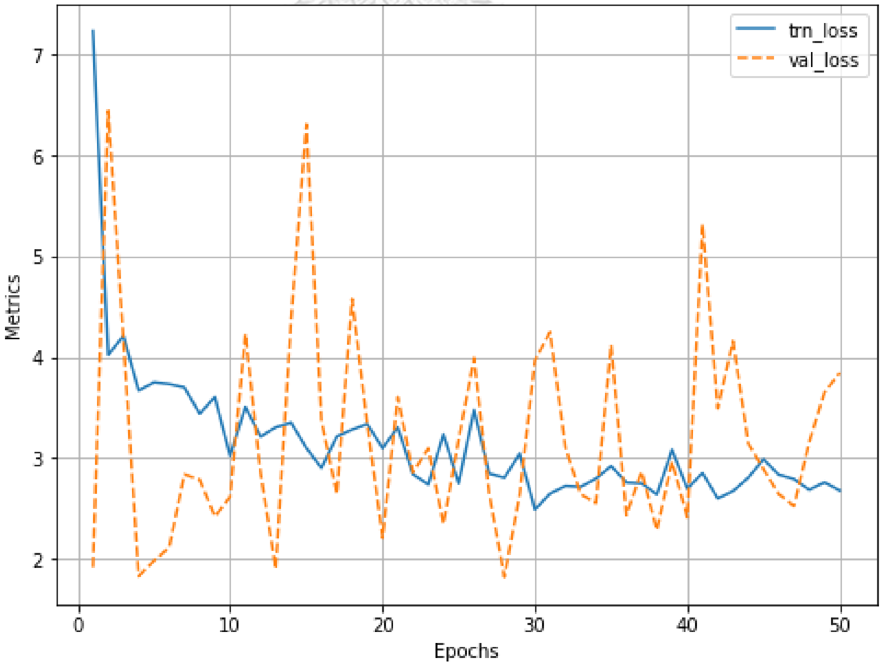


Figure 67. Loss of the 2nd fold of 5-fold cross validation of the R-CNN multi-classification without DCGAN images.

3rd Fold

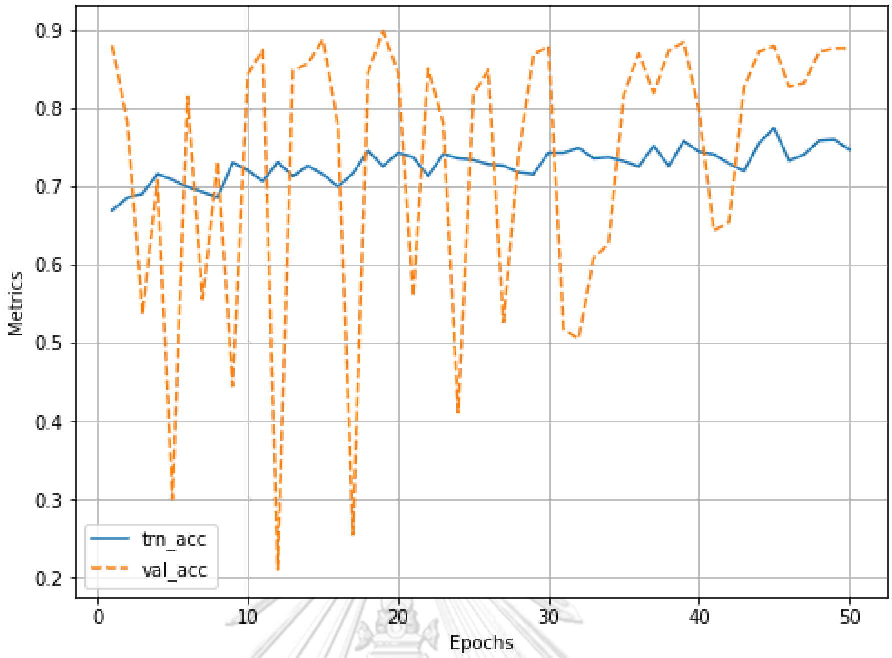


Figure 68. Accuracy of the 3rd fold of 5-fold cross validation of the R-CNN multi-classification without DCGAN images.

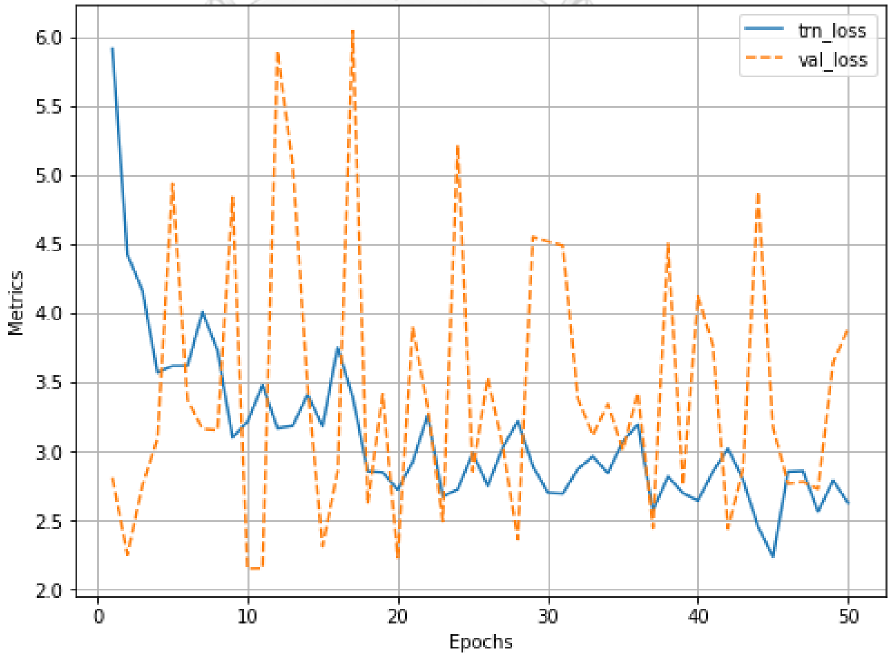


Figure 69. Loss of the 3rd fold of 5-fold cross validation of the R-CNN multi-classification without DCGAN images.

4th Fold

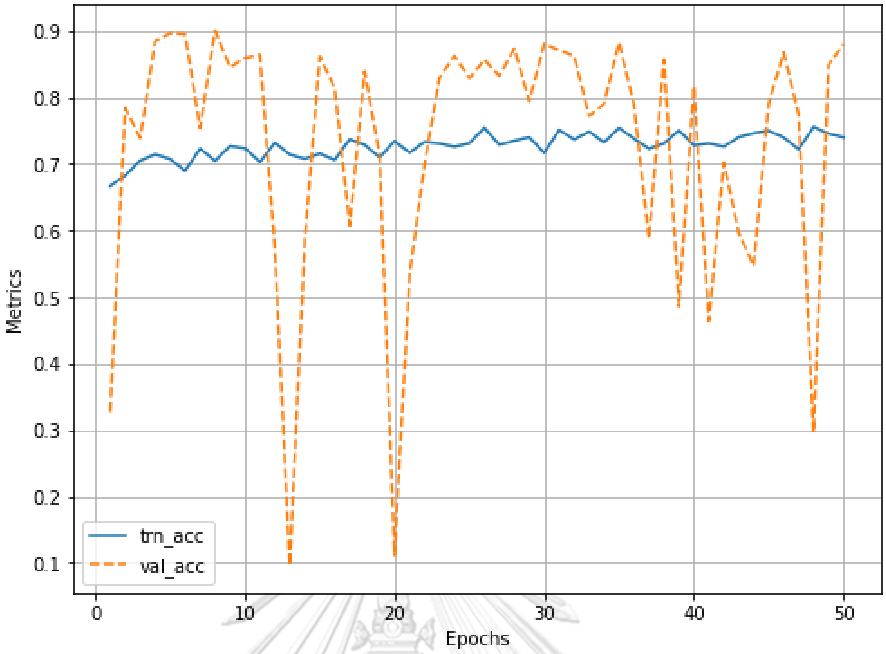


Figure 70. Accuracy of the 4th fold of 5-fold cross validation of the R-CNN multi-classification without DCGAN images.

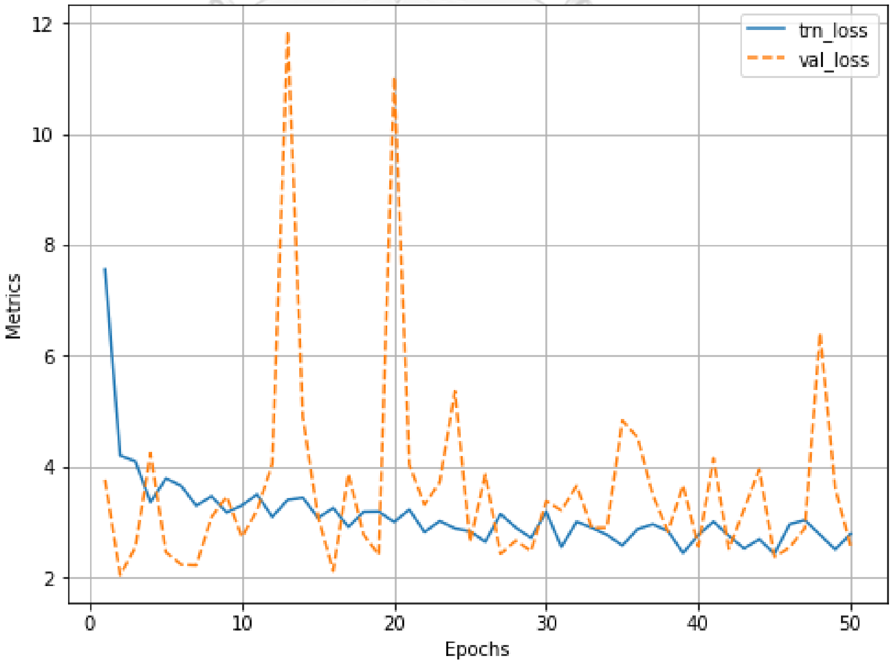


Figure 71. Loss of the 4th fold of 5-fold cross validation of the R-CNN multi-classification without DCGAN images.

5th Fold

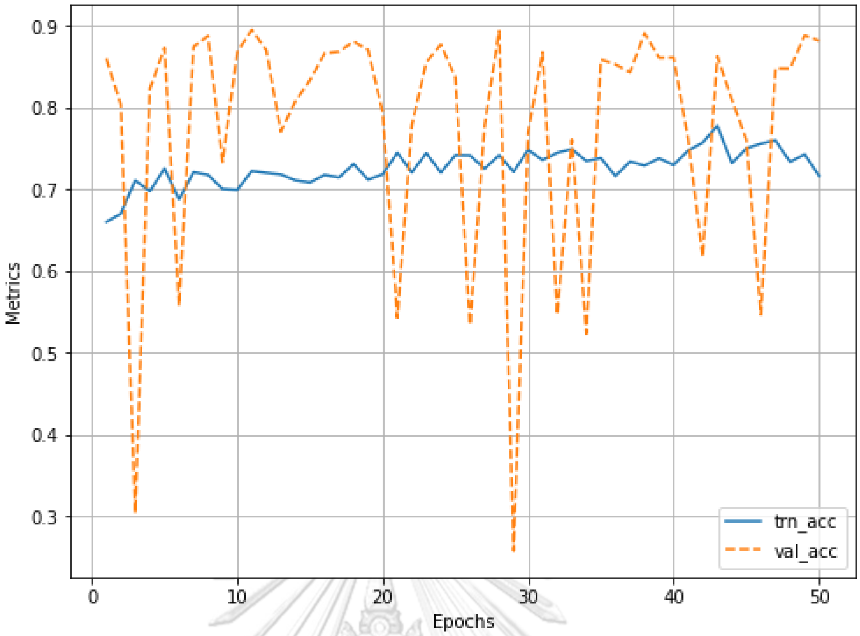


Figure 72. Accuracy of the 5th fold of 5-fold cross validation of the R-CNN multi-classification without DCGAN images.

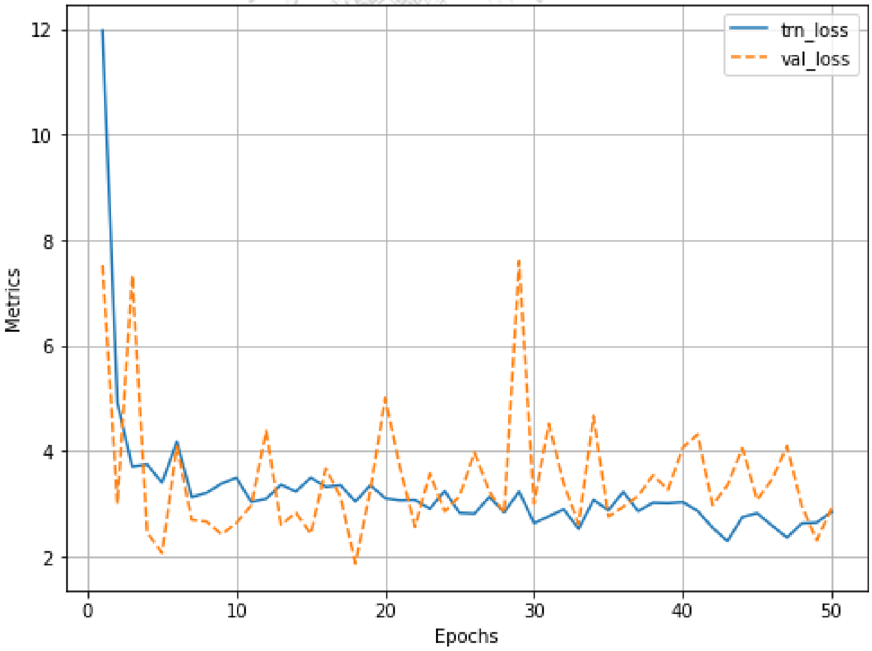


Figure 73. Loss of the 5th fold of 5-fold cross validation of the R-CNN multi-classification without DCGAN images.

Afterwards, the R-CNN for multi-classification was trained on the dataset with DCGAN images included. The model's accuracy and performance were better than the previous one through 5-K-Fold method at 79.8%.

The result of the training is as follows:

1st Fold

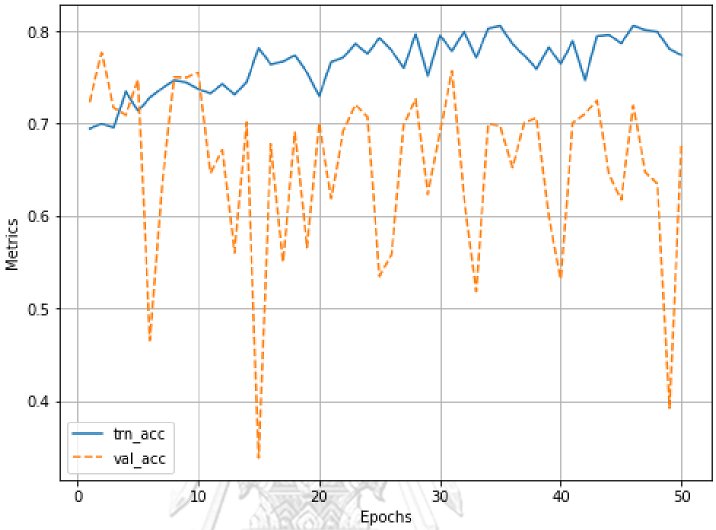


Figure 74. Accuracy of the 1st fold of 5-fold cross validation of the R-CNN multi-classification with DCGAN images.

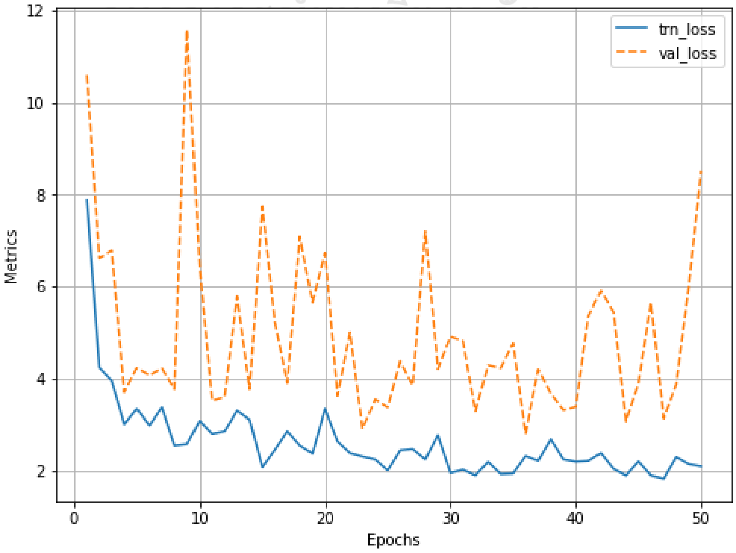


Figure 75. Loss of the 1st fold of 5-fold cross validation of the R-CNN multi-classification with DCGAN images.

2nd Fold

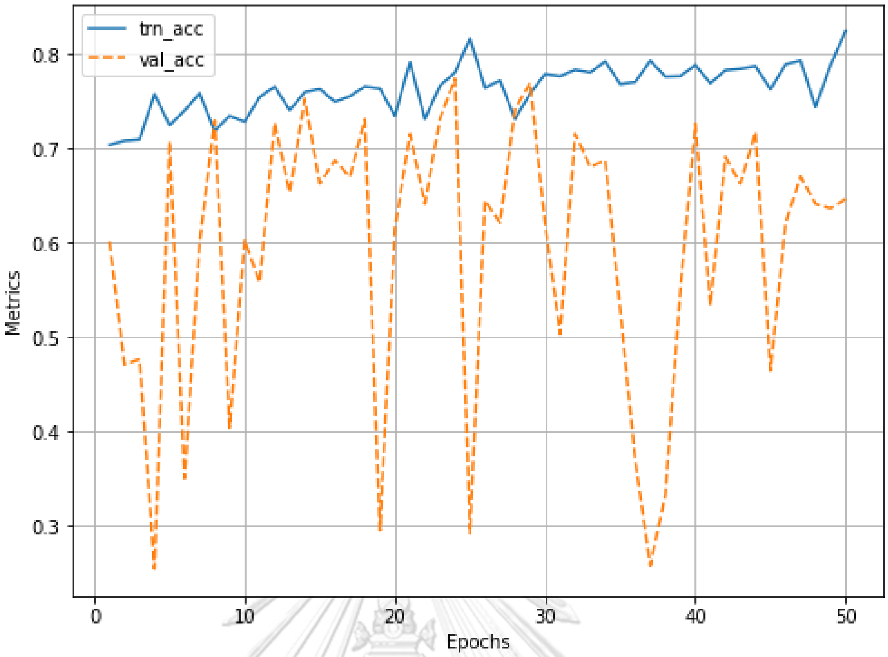


Figure 76. Accuracy of the 2nd fold of 5-fold cross validation of the R-CNN multi-classification with DCGAN images.

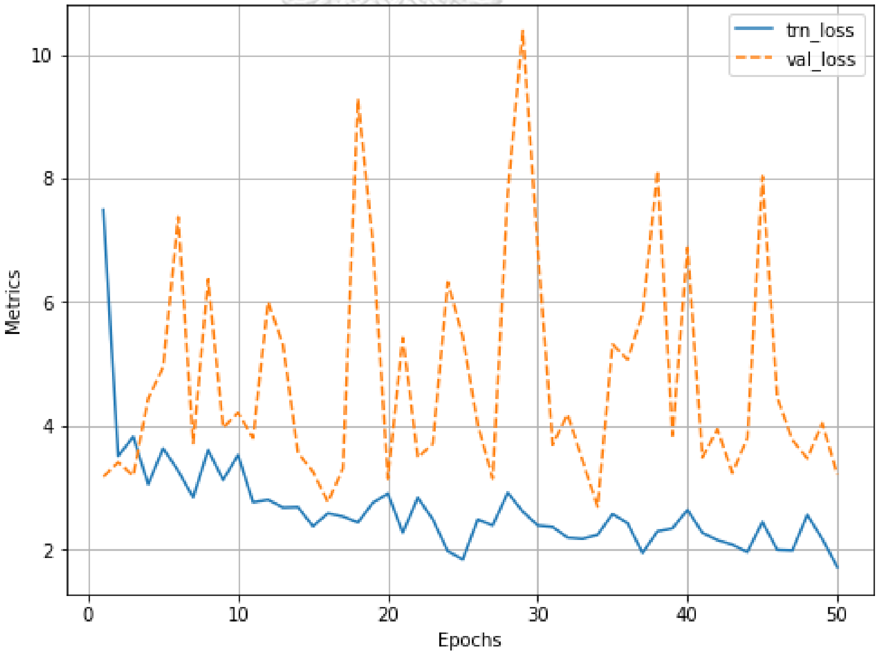


Figure 77. Loss of the 2nd fold of 5-fold cross validation of the R-CNN multi-classification with DCGAN images.

3rd Fold

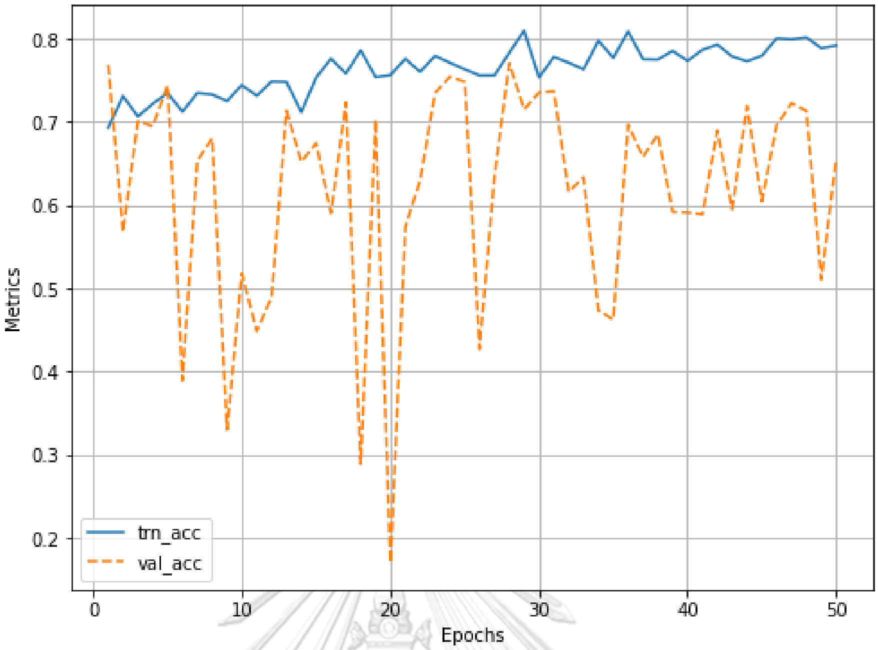


Figure 78. Accuracy of the 3rd fold of 5-fold cross validation of the R-CNN multi-classification with DCGAN images.

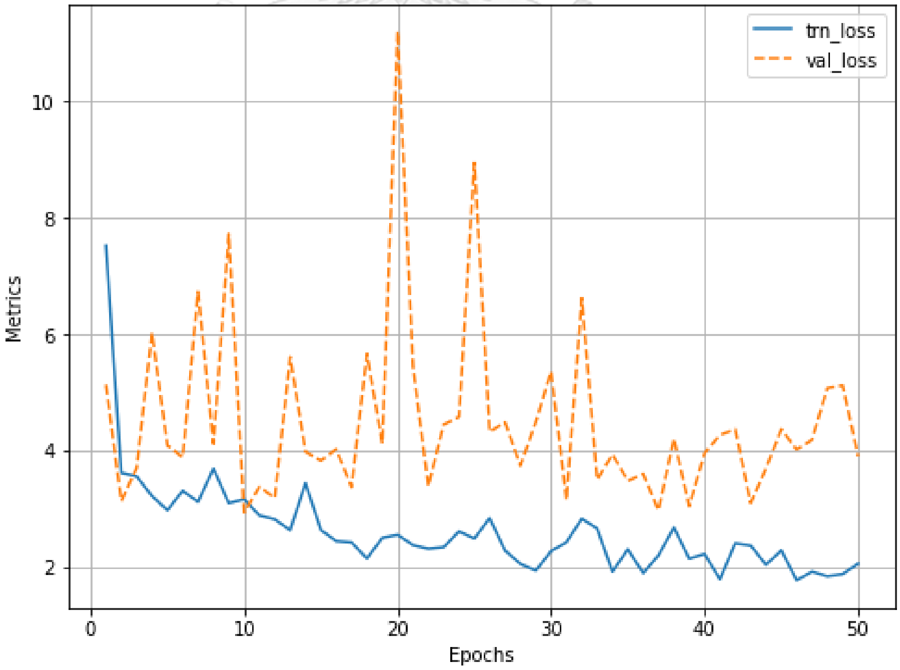


Figure 79. Loss of the 3rd fold of 5-fold cross validation of the R-CNN multi-classification with DCGAN images.

4th Fold

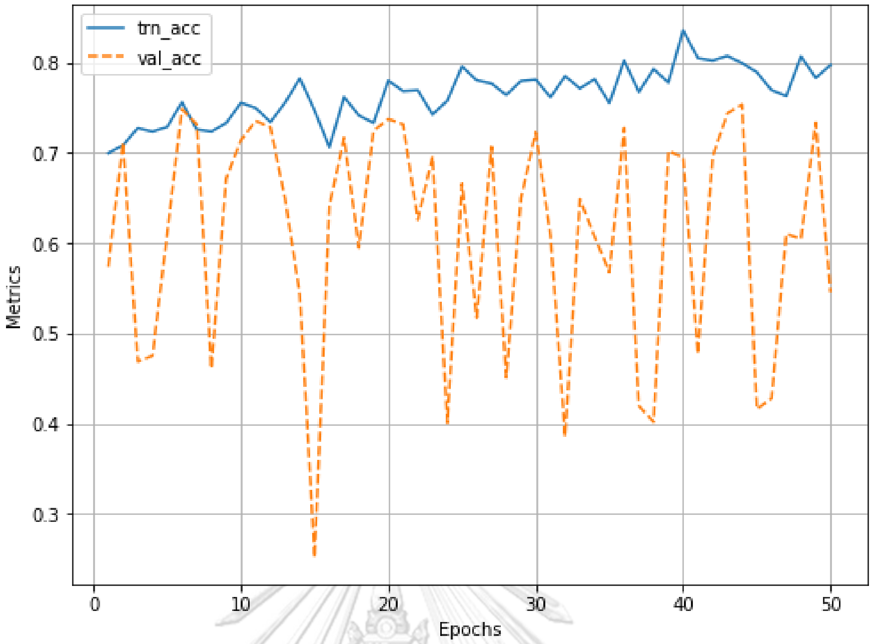


Figure 80. Accuracy of the 4th fold of 5-fold cross validation of the R-CNN multi-classification with DCGAN images.

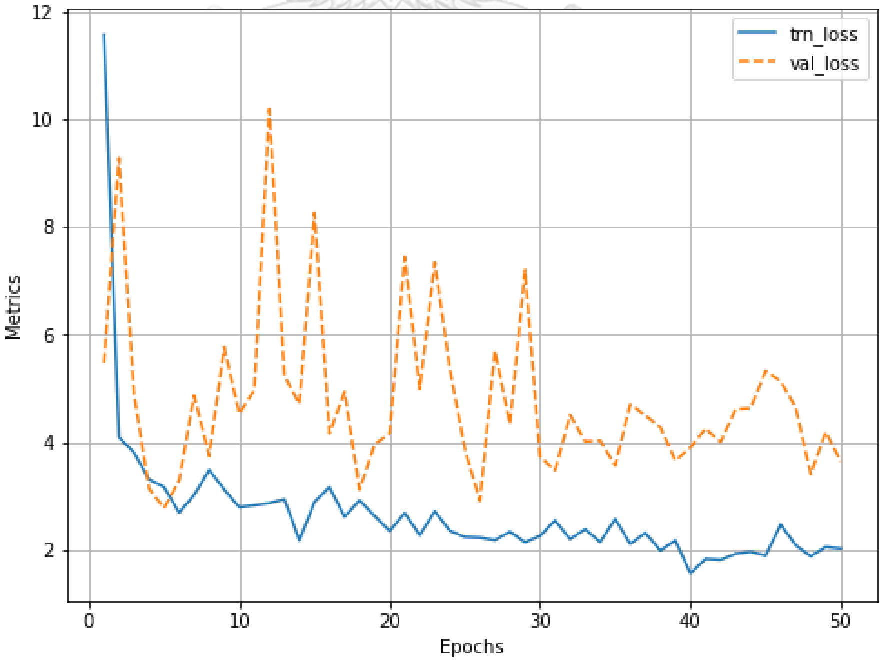


Figure 81. Loss of the 4th fold of 5-fold cross validation of the R-CNN multi-classification with DCGAN images.

5th Fold

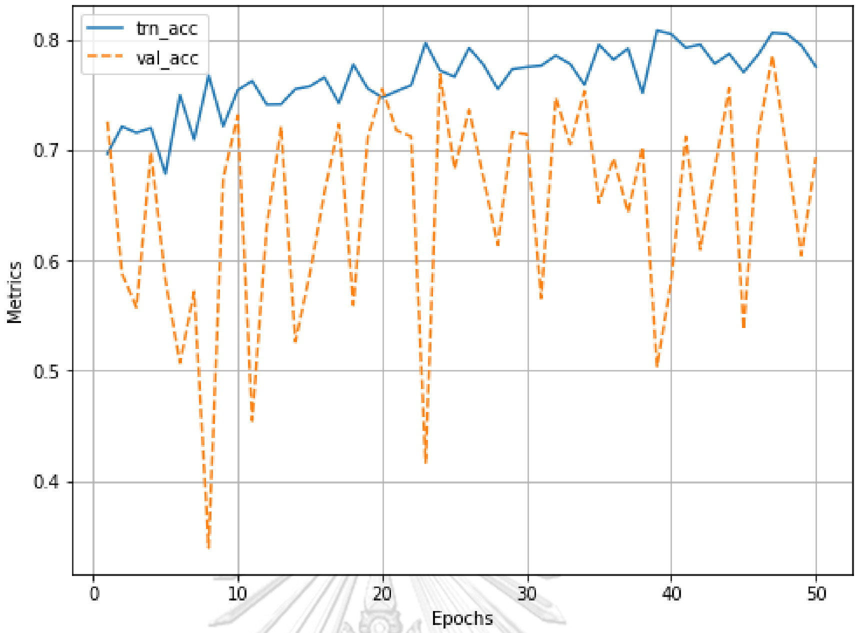


Figure 82. Accuracy of the 5th fold of 5-fold cross validation of the R-CNN multi-classification with DCGAN images.

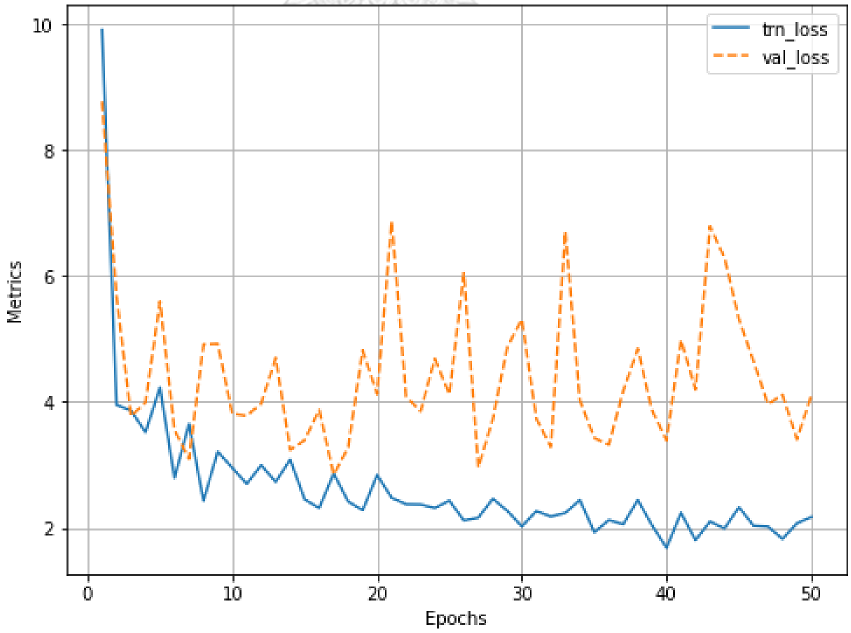


Figure 83. Loss of the 5th fold of 5-fold cross validation of the R-CNN multi-classification with DCGAN images.

Later on, the multi-classifier R-CNN model (with DCGAN images) was tested on the test set which includes US Wagyu, Australian Wagyu, and Japanese Wagyu. Figure 84 shows the results of successful detection of the patterns of marbled fat layers of the US Wagyu beef.

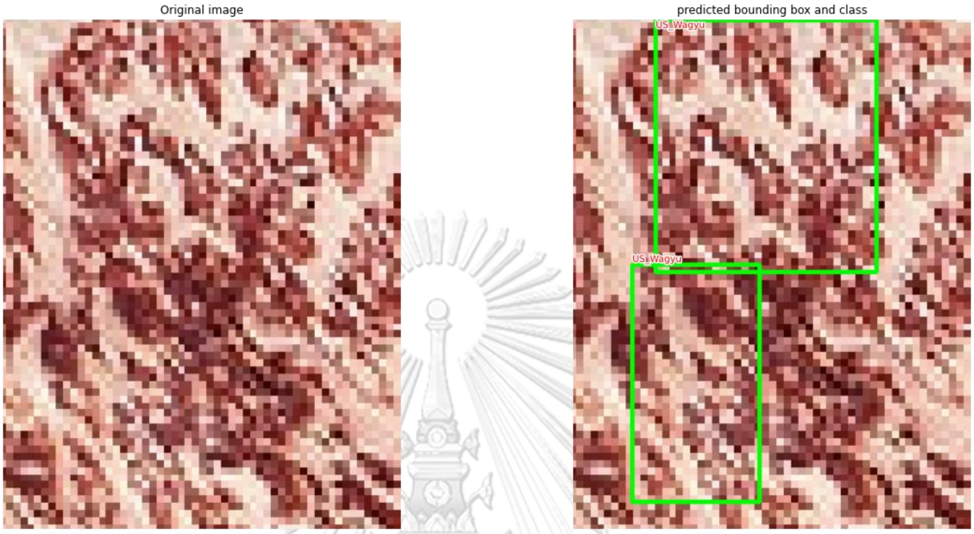


Figure 84. Illustrating how the machine could recognize the pattern of marbled fatty layers of the US Wagyu Beef

Further, we tested the model using images of artificial marbling beef. As illustrated in Figure 85, the R-CNN returned the output of "No object found" since the model could not match the marbled fat layers with either patterns of Japanese, Australian, or the US Wagyu beef.

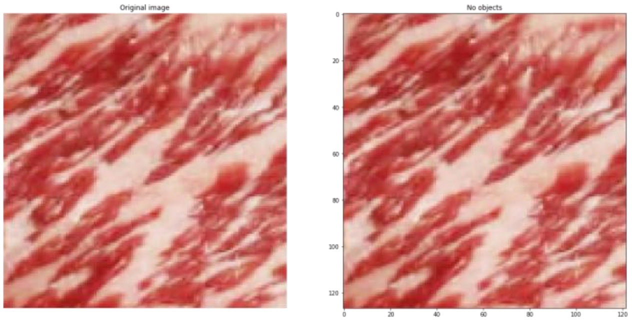


Figure 85 Illustrating how the R-CNN could not recognize the pattern of marbled fatty layers of artificial marbling beef.

4.4 Overall Test Results

Table 3 summarizes the accuracy rates of phase 1.1 using 5-fold cross-validation.

Table 3 Comparison of accuracy of CNN binary classification

DCGAN AUGMENTATION	ACC FOLD1	ACC FOLD2	ACC FOLD3	ACC FOLD4	ACC FOLD5	AVERAGE ACCURACY
WITH	93.5	92.7	91.0	94.3	89.5	**92.2**
WITHOUT	87.0	90.0	91.0	93.5	94.3	91.16

Table 4 summarize the accuracy rates of phase 12 using 5-fold cross-validation.

Table 4 Comparisons of accuracy of CNN multi-classification

DCGAN AUGMENTATION	ACC FOLD1	ACC FOLD2	ACC FOLD3	ACC FOLD4	ACC FOLD5	AVERAGE ACCURACY
WITH	71.5	80.0	79.5	87.0	71.0	**77.8**
WITHOUT	70.0	70.0	72.0	80.0	74.0	73.2

Table 5 summarizes the accuracy rates of phase 2.1 using 5-fold cross-validation.

Table 5 Comparisons of accuracy of R-CNN binary classification.

จุฬาลงกรณ์มหาวิทยาลัย

DCGAN AUGMENTATION	ACC FOLD1	ACC FOLD2	ACC FOLD3	ACC FOLD4	ACC FOLD5	AVERAGE ACCURACY
WITH	93.5	95.0	94.0	94.0	94.5	**94.2**
WITHOUT	93.0	92.5	92.5	94.0	93.0	93.0

Table 6 summarizes the accuracy rates of phase 2.2 using 5-fold cross-validation.

Table 6 Comparison of accuracy of R-CNN multi-classification

DCGAN AUGMENTATION	ACC FOLD1	ACC FOLD2	ACC FOLD3	ACC FOLD4	ACC FOLD5	AVERAGE ACCURACY
WITH	78.5	82.0	80.0	80.0	78.5	**79.8**
WITHOUT	76.5	74.5	75.0	74.0	71.5	74.3

As illustrates by Table 3, the CNN model with DCGAN images included, outperformed the one without, by gaining more accuracy by around 1%. This shows that by adding the augmented images, in this case DCGAN images, the model’s accuracy would increase consequently, due to that there are more sources of data for the model to train on. As same as the result from previous Table, the results from Table 4, strongly suggest that by adding the augmented images created via DCGAN algorithms, would increase the performance and accuracy of the model used for training the multi-classification tasks. In this case, the model’s accuracy improves by 4.6% after adding the DCGAN images.

As it could be seen from Table 5, the accuracy of R-CNN for binary-classification with DCGAN images included after 5-fold cross validation is more than the accuracy of the one without by 1.2 %. This, again, proves that by adding the augmented image created by DCGAN, the model was able to perform and gain more accuracy than the one limited to just only the original dataset. As similar as the results from Table 6, which the R-CNN model with the DCGAN images outperformed the model of the one without by 5.5%. This clearly shows that when a task gets more demanding such as a multi-classifying task the more dataset available, the more accuracy the model would get. And by adding the DCGAN images to the dataset, it proved as another successful way to increase the accuracy for the model when the number of data sources were limited.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

To add on, both the binary classification and multi-classification results of R-CNN were proven to be better than the ones of CNN architecture. This stands as a proof that dividing the image of Wagyu Marbled fat layer into small pieces first (warped process of R-CNN), would enable the model to train more effectively and accurately, and consequently, enable it to detect the distinguished pattern of the source of the beef.

Furthermore, while the CNN model always returns the output belongs to the set of class labels, the R-CNN will output “Not found” when detecting anomalous inputs out of the training data distribution. The preliminary result of phase 2 reported that the R-

CNN was promising for correctly detecting the testing images of which the sources not included in the training set.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

CHAPTER 5

CONCLUSION

For this research, we proposed the deep neural network modelling for image classification of the Wagyu sources: Japanese and Australian. Due to the small datasets, data augmentation was carried out using DCGAN to mainly reduce bias and overfitting. The constructed CNNs achieved the accuracy of 95% (DCGAN images included). When the model was modified to have an activation function as Softmax as multi-classification was required, the adjusted model with DCGAN images included reached an accuracy at 77.8%. However, when GPUs was triggered via NVIDIA CUDA, the model become much more robust and achieved a higher accuracy of classification. For the binary-classification, the accuracy arises as high as 94.2% (DCGAN images included). And for the multi-classification, the model achieved a satisfying level of accuracy of 79.8% (DCGAN images included). Furthermore, when DCGAN images were not included as part of the source for training, the performance and accuracy of every category decreased, this includes both binary classification and multi-classification.

None of the less, the model could be exposed to more classes of Wagyu such as the UK Wagyu and Canada Wagyu etc. Further direction would be adjusting the model to be able to serve and multi-classify more sources of Wagyu beef along with OOD at the same time.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

REFERENCES

1. *Australian Wagyu Association, .* [cited March 2021; Available from: https://www.wagyu.org.au/.
2. *American Wagyu Association*. March 2021]; Available from: https://wagyu.org/.
3. *The Wagyu Shop*. [cited March 2021; Available from: https://wagyushop.com/.
4. LeCun, Y., *LeNet-5, convolutional neural networks.* URL: http://yann. lecun. com/exdb/lenet, 2015. **20**(5): p. 14.
5. Radford, A., L. Metz, and S. Chintala, *Unsupervised representation learning with deep convolutional generative adversarial networks.* arXiv preprint arXiv:1511.06434, 2015.
6. *Tensorflow, "Deep Convolutional Generative Adversarial Networks"*. Available from: https://www.tensorflow.org/tutorials/generative/dcgan.
7. Dosovitskiy, A., et al., *Discriminative unsupervised feature learning with exemplar convolutional neural networks.* IEEE transactions on pattern analysis and machine intelligence, 2015. **38**(9): p. 1734-1747.
8. R. Jagtap. *Implementing Deep Convolutional Generative Adversarial Networks(DCGANs)*. 2020 Available from: https://towardsdatascience.com/implementing-deep-convolutional-generative-adversarial-networks-dcgan-573df2b63c0d.
9. Ayyadevara, K. and Y. Reddy, *Modern Computer Vision with Pytorch*. 2020: Packt Publishing Ltd.
10. *Towards Data Science*. [cited July 2020; Available from: https://towardsdatascience.com/.
11. drainingsun. *ybat*. [cited July 2020; Available from: https://github.com/drainingsun/ybat.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

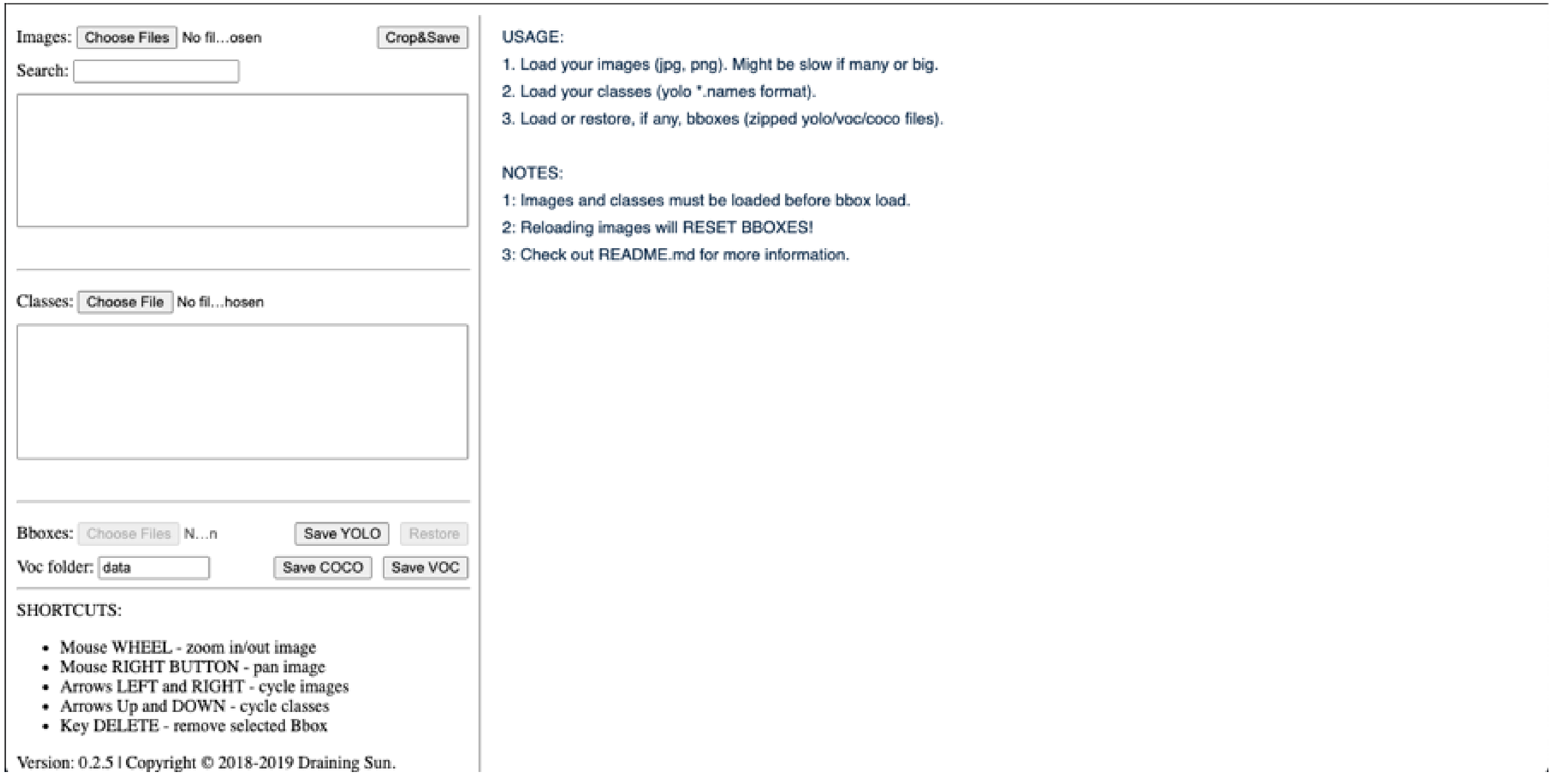
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

APPENDIX

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Demonstration of applying YBAT for converting JPEG or PNG files to XML files. Steps are as follow:

1. Open the ybat.html using Chrome Browser as default since other browsers may not be supported by the software



Voc folder: data

SHORTCUTS:

- Mouse WHEEL - zoom in/out image
- Mouse RIGHT BUTTON - pan image
- Arrows LEFT and RIGHT - cycle images
- Arrows Up and DOWN - cycle classes
- Key DELETE - remove selected Bbox

Version: 0.2.5 | Copyright © 2018-2019 Draining Sun.

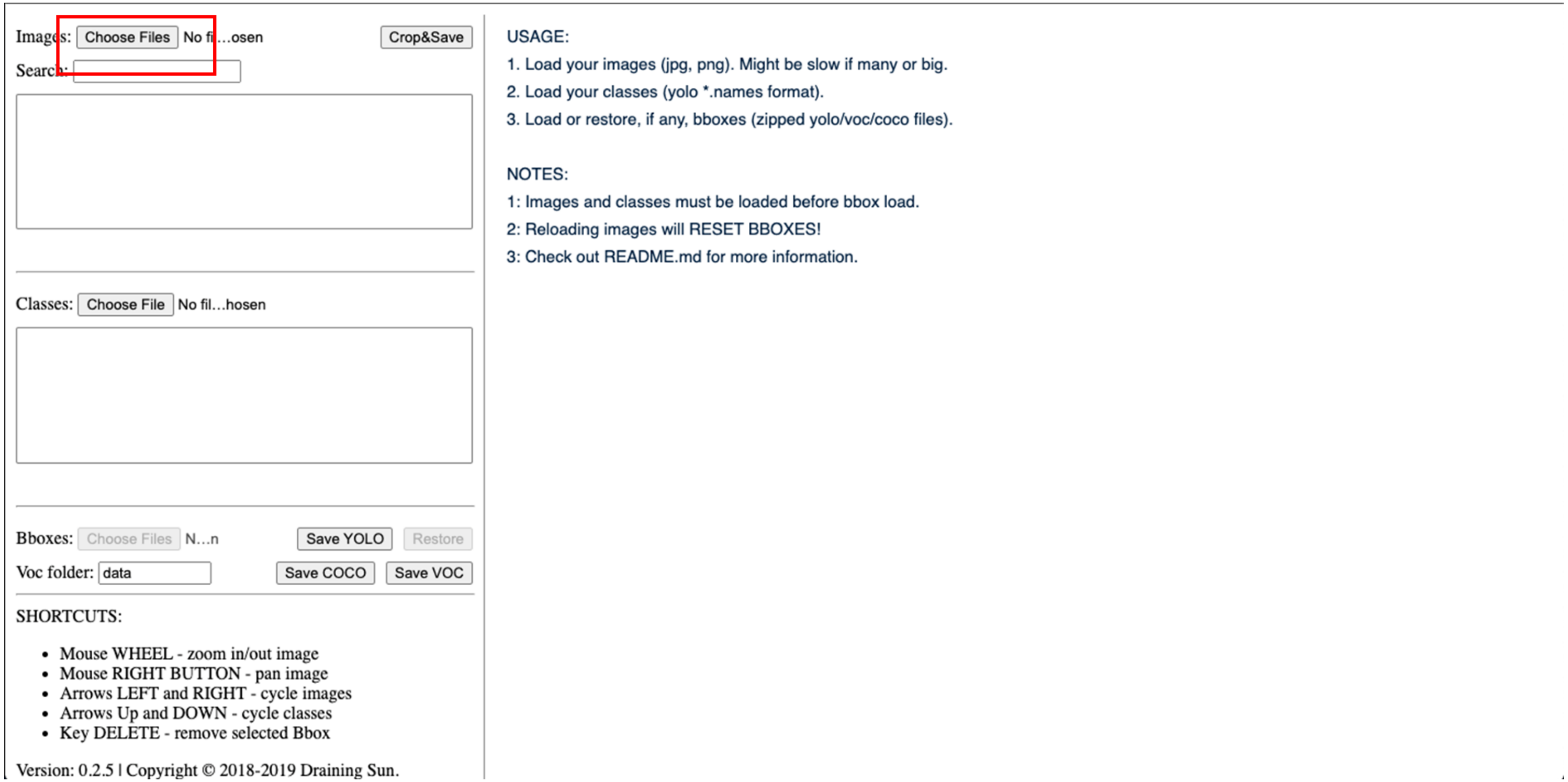
USAGE:

1. Load your images (jpg, png). Might be slow if many or big.
2. Load your classes (yolo *.names format).
3. Load or restore, if any, bboxes (zipped yolo/voc/coco files).

NOTES:

1: Images and classes must be loaded before bbox load.
2: Reloading images will RESET BBOXES!
3: Check out README.md for more information.

2. Locate the ‘Choose Files’(Images) button



USAGE:

1. Load your images (jpg, png). Might be slow if many or big.
2. Load your classes (yolo *.names format).
3. Load or restore, if any, bboxes (zipped yolo/voc/coco files).

NOTES:

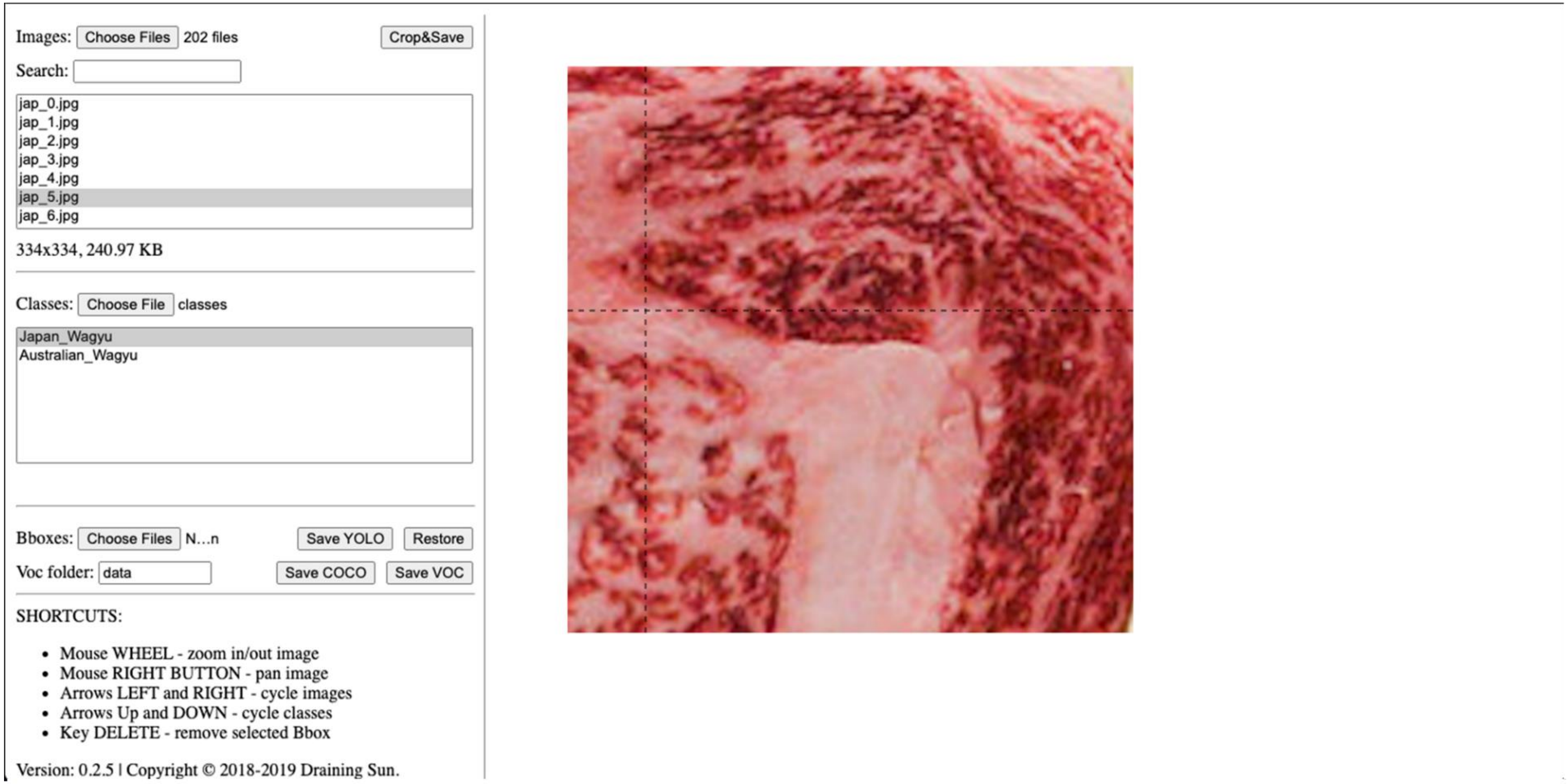
1: Images and classes must be loaded before bbox load.
2: Reloading images will RESET BBOXES!
3: Check out README.md for more information.

SHORTCUTS:

- Mouse WHEEL - zoom in/out image
- Mouse RIGHT BUTTON - pan image
- Arrows LEFT and RIGHT - cycle images
- Arrows Up and DOWN - cycle classes
- Key DELETE - remove selected Bbox

Version: 0.2.5 | Copyright © 2018-2019 Draining Sun.

3. Choose the file to be trained and adjust its size as needed



Images: 202 files

Search:

jap_0.jpg
jap_1.jpg
jap_2.jpg
jap_3.jpg
jap_4.jpg
jap_5.jpg
jap_6.jpg

334x334, 240.97 KB

Classes: classes

Japan_Wagyu
Australian_Wagyu

Bboxes: N…n

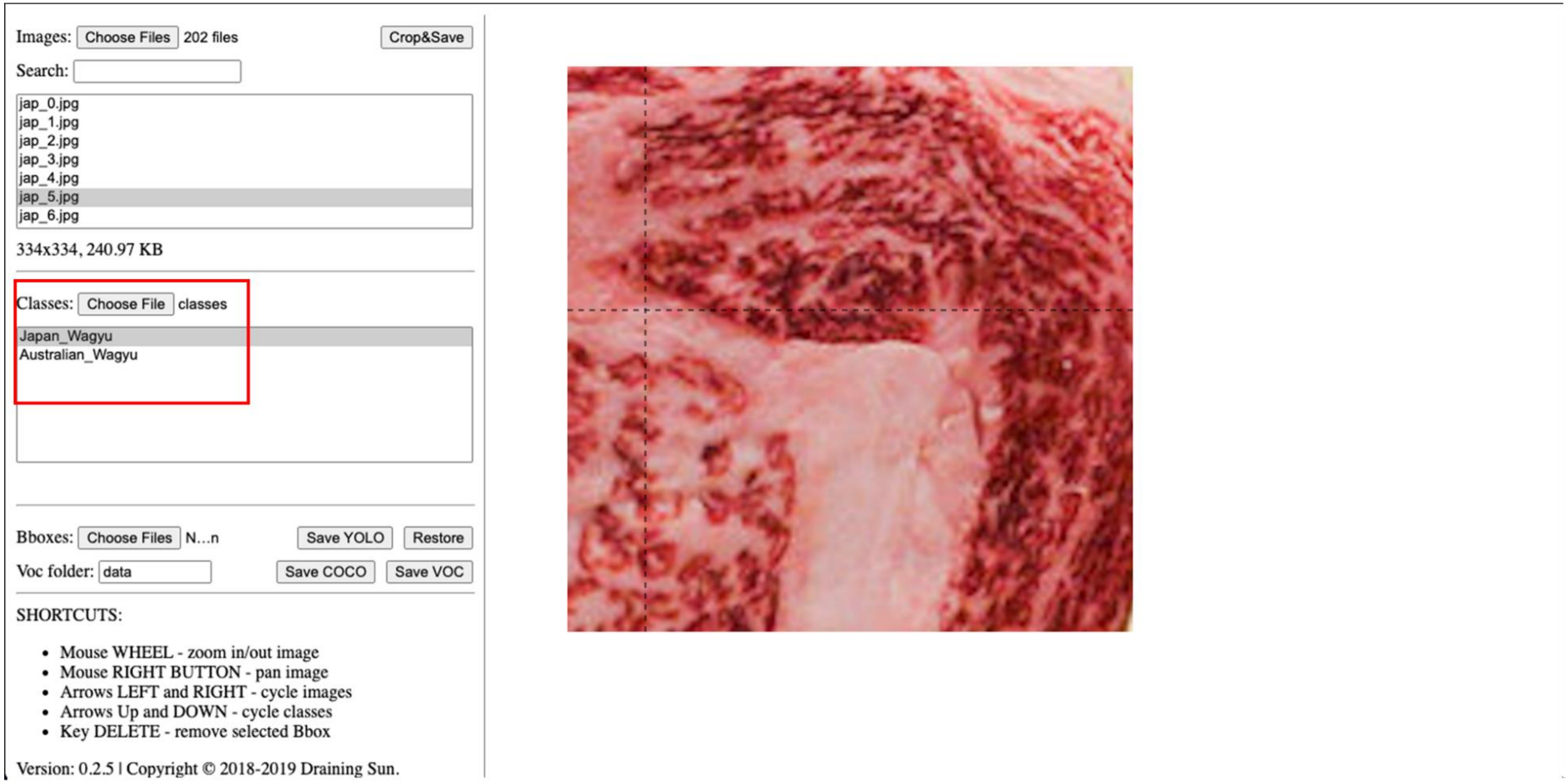
Voc folder: data

SHORTCUTS:

- Mouse WHEEL - zoom in/out image
- Mouse RIGHT BUTTON - pan image
- Arrows LEFT and RIGHT - cycle images
- Arrows Up and DOWN - cycle classes
- Key DELETE - remove selected Bbox

Version: 0.2.5 | Copyright © 2018-2019 Draining Sun.

4. Locate the ‘Choose files’ (classes) button, and upload the classes of the dataset, note that the file type uploaded via this section must be .txt files.



Images: 202 files

Search:

jap_0.jpg
jap_1.jpg
jap_2.jpg
jap_3.jpg
jap_4.jpg
jap_5.jpg
jap_6.jpg

334x334, 240.97 KB

Classes: classes

Japan_Wagyu
Australian_Wagyu

Bboxes: N...n

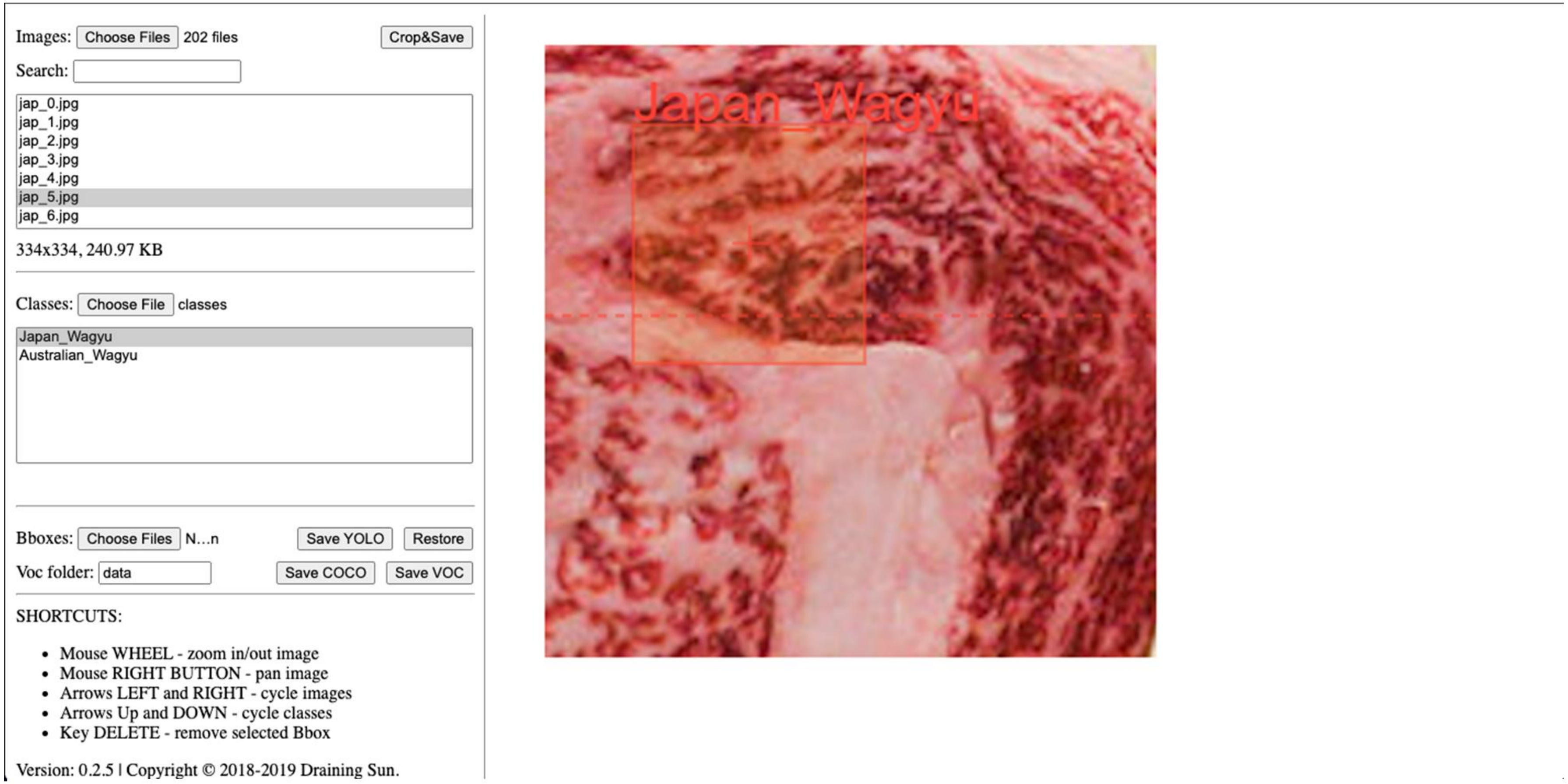
Voc folder: data

SHORTCUTS:

- Mouse WHEEL - zoom in/out image
- Mouse RIGHT BUTTON - pan image
- Arrows LEFT and RIGHT - cycle images
- Arrows Up and DOWN - cycle classes
- Key DELETE - remove selected Bbox

Version: 0.2.5 | Copyright © 2018-2019 Draining Sun.

5. Draw the box on top of the area to be trained for object detection



Images: 202 files

Search:

jap_0.jpg
jap_1.jpg
jap_2.jpg
jap_3.jpg
jap_4.jpg
jap_5.jpg
jap_6.jpg

334x334, 240.97 KB

Classes: classes

Japan_Wagyu
Australian_Wagyu

Bboxes: N...n

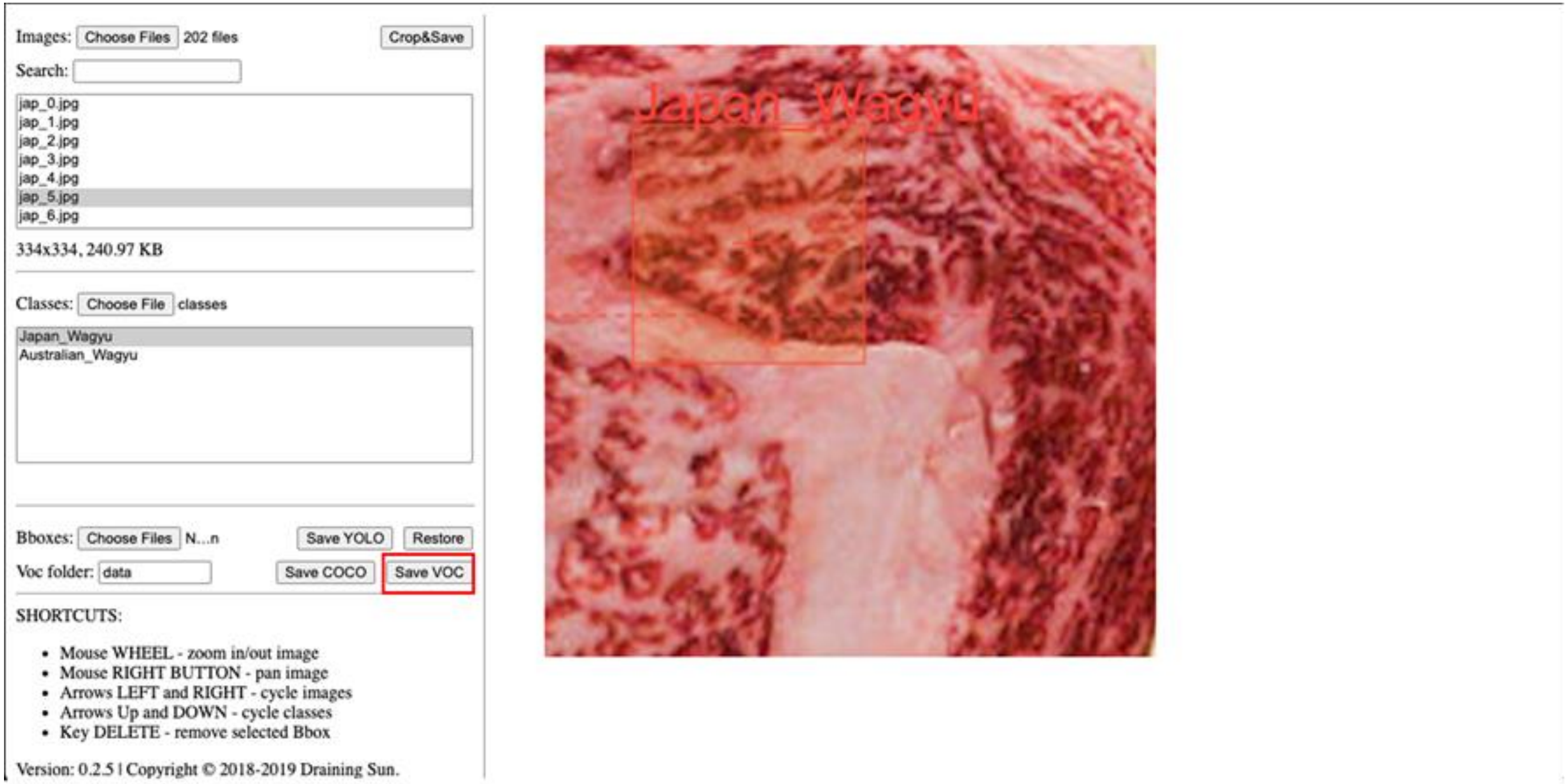
Voc folder: data

SHORTCUTS:

- Mouse WHEEL - zoom in/out image
- Mouse RIGHT BUTTON - pan image
- Arrows LEFT and RIGHT - cycle images
- Arrows Up and DOWN - cycle classes
- Key DELETE - remove selected Bbox

Version: 0.2.5 | Copyright © 2018-2019 Draining Sun.

6. Locate the Save as VOC button and save. The file would be saved as XML file automatically.



Images: 202 files

Search:

jap_0.jpg
jap_1.jpg
jap_2.jpg
jap_3.jpg
jap_4.jpg
jap_5.jpg
jap_6.jpg

334x334, 240.97 KB

Classes: classes

Japan_Wagyu
Australian_Wagyu

Bboxes: N...n

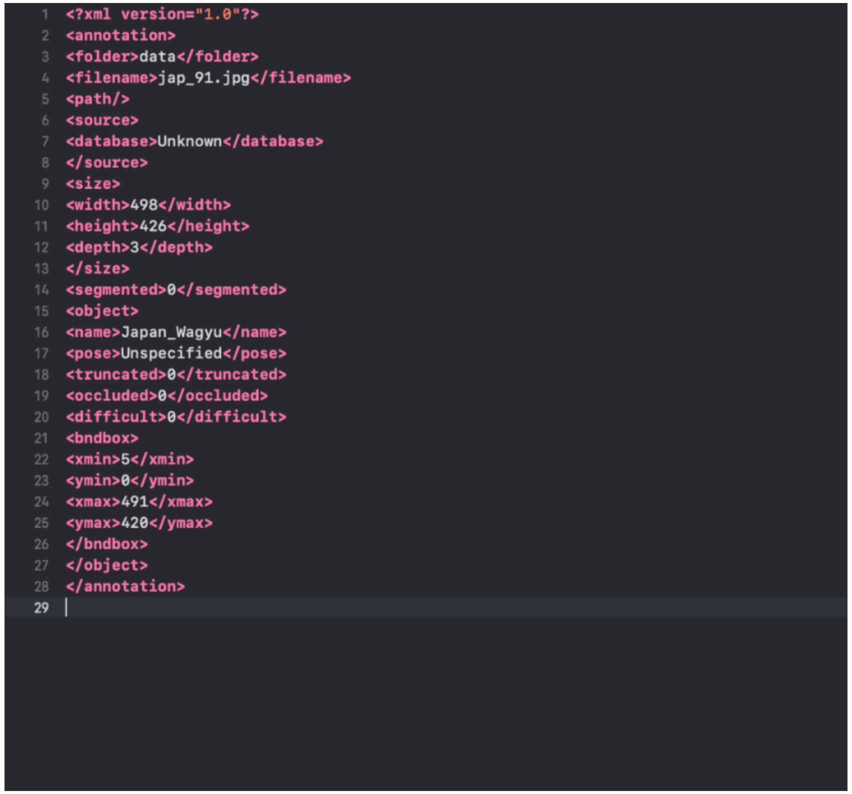
Voc folder: data

SHORTCUTS:

- Mouse WHEEL - zoom in/out image
- Mouse RIGHT BUTTON - pan image
- Arrows LEFT and RIGHT - cycle images
- Arrows Up and DOWN - cycle classes
- Key DELETE - remove selected Bbox

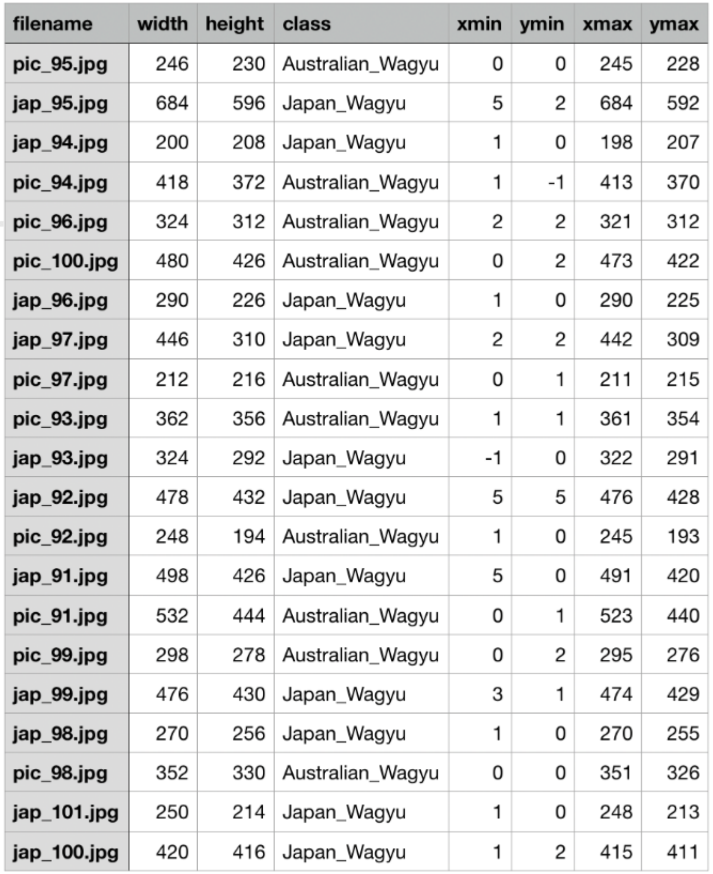
Version: 0.2.5 | Copyright © 2018-2019 Draining Sun.

7. Next, write a code to convert the XML file (as shown below) to a csv file. The example of the xml file is illustrated below



```
<?xml version="1.0"?>
<annotation>
<folder>data</folder>
<filename>jap_91.jpg</filename>
<path/>
<source>
<database>Unknown</database>
</source>
<size>
<width>498</width>
<height>426</height>
<depth>3</depth>
</size>
<segmented>0</segmented>
<object>
<name>Japan_Wagyu</name>
<pose>Unspecified</pose>
<truncated>0</truncated>
<occluded>0</occluded>
<difficult>0</difficult>
<bndbox>
<xmin>5</xmin>
<ymin>0</ymin>
<xmax>491</xmax>
<ymax>420</ymax>
</bndbox>
</object>
</annotation>
```

8. Convert the XML files to CSV files possessing the form provided below in



filename	width	height	class	xmin	ymin	xmax	ymax
pic_95.jpg	246	230	Australian_Wagyu	0	0	245	228
jap_95.jpg	684	596	Japan_Wagyu	5	2	684	592
jap_94.jpg	200	208	Japan_Wagyu	1	0	198	207
pic_94.jpg	418	372	Australian_Wagyu	1	-1	413	370
pic_96.jpg	324	312	Australian_Wagyu	2	2	321	312
pic_100.jpg	480	426	Australian_Wagyu	0	2	473	422
jap_96.jpg	290	226	Japan_Wagyu	1	0	290	225
jap_97.jpg	446	310	Japan_Wagyu	2	2	442	309
pic_97.jpg	212	216	Australian_Wagyu	0	1	211	215
pic_93.jpg	362	356	Australian_Wagyu	1	1	361	354
jap_93.jpg	324	292	Japan_Wagyu	-1	0	322	291
jap_92.jpg	478	432	Japan_Wagyu	5	5	476	428
pic_92.jpg	248	194	Australian_Wagyu	1	0	245	193
jap_91.jpg	498	426	Japan_Wagyu	5	0	491	420
pic_91.jpg	532	444	Australian_Wagyu	0	1	523	440
pic_99.jpg	298	278	Australian_Wagyu	0	2	295	276
jap_99.jpg	476	430	Japan_Wagyu	3	1	474	429
jap_98.jpg	270	256	Japan_Wagyu	1	0	270	255
pic_98.jpg	352	330	Australian_Wagyu	0	0	351	326
jap_101.jpg	250	214	Japan_Wagyu	1	0	248	213
jap_100.jpg	420	416	Japan_Wagyu	1	2	415	411

CHULALONGKORN UNIVERSITY

VITA

NAME	Nattakorn Kointarangkul
DATE OF BIRTH	9 June 1986
PLACE OF BIRTH	Bangkok
INSTITUTIONS ATTENDED	Berklee College of Music, Bachelor Degree Imperial College London, Master Degree Chulalongkorn University, Master Degree
HOME ADDRESS	143-5 Phaholyotin Rd., Samsennai, Phayatai, Bangkok 10400

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY