

Virus Identification From Digital Images Using Deep Learning

Luxin Zhang

A thesis submitted to the Auckland University of Technology
in partial fulfilment of the requirements for the degree of
Master of Computer and Information Sciences (MCIS)

2020

School of Engineering, Computer & Mathematical Sciences

Abstract

Given the electron microscopy images, virus recognition using deep learning approaches from digital images is critical at present, because virus identification by human virology experts is slow and time-consuming, this research project aims to develop a deep learning-based method for automatic virus detection. There are four virus species in this thesis, they are SARS, MERS, HIV, and COVID-19. This study is based on classification and bounding box regression.

In this thesis, we firstly examine virus morphological characteristics and propose a novel loss function which targets to reflect the viruses on the given electron micrograph. In this project, we take into account the attention mechanism, virus images are processed in advance to be trained for classification and localization. In order to make the best estimation of bounding boxes and classification for a virus, we test five deep learning networks: R-CNN, Fast R-CNN, Faster R-CNN, YOLO, and SSD, based on the prior knowledge of virus electron microscopy. Additionally, in this project, we discuss the deep learning training problems and illustrate the evaluation approaches. The conclusion reveals SSD and Faster R-CNN outperform in the virus detection from digital images.

Keywords: Classification, localization, CNN, virus, electron microscopy images

Table of Contents

Abstract.....	II
Table of Contents.....	III
List of Figures.....	VI
List of Tables.....	VIII
Acknowledgment.....	X
Chapter 1 Introduction.....	1
1.1 Background and Motivation.....	2
1.2 Research Question.....	3
1.3 Contributions.....	4
1.4 Objectives of This Thesis.....	5
1.5 The Structure of This Thesis.....	5
Chapter 2 Literature Review.....	7
2.1 Introduction.....	8
2.2 Target Morphological & Electron Micrograph Characteristics.....	8
2.2.1 HIV Morphological Characteristics on Electron Micrograph.....	10
2.2.2 SARS-CoV-1 Morphological Characteristics on Electron Micrograph.....	11
2.2.3 MERS-CoV Morphological Characteristics on Electron Micrograph.....	12
2.2.4 SARS-COV-2 Morphological Characteristics on Electron Micrograph.....	13
2.3 Deep Learning.....	14
2.4 R-CNN.....	17
2.5 Fast R-CNN.....	19
2.6 Faster R-CNN.....	24
2.7 You Only Look Once (YOLO).....	26
2.8 SSD.....	31
Chapter 3 Methodology.....	35
3.1 Research Design.....	36
3.2 Dataset Preprocessing.....	36
3.2.1 Grey-Scaling Transformation.....	36
3.2.2 Image Resizing.....	37
3.2.3 Image Brightness Adjustment.....	37
3.2.4 Image Contrast Adjustment.....	40
3.2.5 Image Sharpening.....	42

3.2.6	Image Rotation	43
3.2.7	Image Noise Removal	44
3.2.8	Image Random Region Removal	46
3.3	Jaccard Index Crop	47
3.4	An Image Preprocessing Pipeline	48
3.5	Prior Virus Morphological Knowledge	50
3.6	Attention Mechanism	55
Chapter 4 Training		58
4.1	Training Toolboxes	59
4.2	Supervised Training.....	59
4.3	Activation Functions	59
4.4	Hyperparameters.....	60
4.4.1	Learning Rate	60
4.4.2	Mini-Batch Size	61
4.5	Regularization.....	62
4.6	Training Parameters.....	63
4.7	Attention Mechanism Training.....	63
4.8	Base Networks for Feature Extraction.....	64
4.9	Training for Virus Detection (Classification and Localization)	66
Chapter 5 Results		68
5.1	Evaluation Methods.....	69
5.2	Performance Metrics.....	72
5.2.1	R-CNN	74
5.2.2	Fast R-CNN.....	75
5.2.3	Faster R-CNN.....	76
5.2.4	YOLO.....	77
5.2.5	SSD	78
Chapter 6 Analysis and Discussions		82
6.1	Analysis	83
6.1.1	Protein Projections and Morphological Features	83
6.1.2	Morphological Feature for Each Type of Virus	83
6.1.3	Attention Mechanism	84
6.2	Discussions	84
6.2.1	R-CNN	84
6.2.2	Performance Discussions	85

6.2.3	The Base Networks	85
6.2.4	The Bounding Box	85
6.2.5	Failed Prediction Discussions	86
6.2.6	Learning Rates	86
6.2.7	The Contribution of Prior Knowledge	86
Chapter 7 Conclusion and Future Work		87
7.1	Conclusion.....	88
7.2	Limitations.....	89
7.3	Future Work.....	89
References.....		91

List of Figures

Figure 2.1: Electron micrograph of HIV.....	11
Figure 2.2: Electron micrograph of SARS coronavirus	12
Figure 2.3: An example of MERS particles under the electron microscope (in 100 nm)	13
Figure 2.4: Electron micrograph of COVID-19.....	14
Figure 2.5: A typical YOLO process on an image	28
Figure 2.6: SSD convolutional layers on which predictions are made (parameters shown in this figure are indicative only, may or may not be the same in practice)	33
Figure 3.1: A COVID-19 image (perceptually dark)	39
Figure 3.2: An example of a COVID-19 virus particle image before and after applied enhanced brightness level.....	39
Figure 3.3: Image before and after contrast adjustment.....	41
Figure 3.4: An example of a sharpened image.....	43
Figure 3.5: An image with a rotation angle of 45 degrees.....	44
Figure 3.6: An example of noise removal.....	46
Figure 3.7: An example of random image region removal.....	47
Figure 3.8: An example of a random image cropping by the defined Jaccard Index condition.....	48
Figure 3.9: An example of four data enhancement pre-processing techniques, displayed as before and after.....	50
Figure 3.10: A coronavirus micrograph after thresholding.....	51
Figure 3.11: A region of interest from a denoised coronavirus electron micrograph.....	53
Figure 3.12: A region of interest from a coronavirus electron micrograph with spherical enclosing.....	54

Figure 3.13: An attention mechanism in computer vision	56
Figure 3.14: An electron microscopy image of four coronavirus particles and its masks.....	57
Figure 3.15: A soft masked source image by using attention mechanism (this exemplary image is presented as the ground truth instead of a real output)	57
Figure 5.1: The trends of R-CNN training loss.....	75
Figure 5.2: The trends of Fast R-CNN training loss.....	76
Figure 5.3: The loss trends of Faster R-CNN training.....	77
Figure 5.4: The trends of YOLO training loss.....	78
Figure 5.5: SSD training loss trends.....	79
Figure 5.6: The example of virus detection 1 (COVID-19)	79
Figure 5.7: The example of virus detection 2 (MERS)	80
Figure 5.8: The example of virus detection 3 (SARS)	80
Figure 5.9: The example of virus detection 4 (HIV)	81

List of Tables

Table 3.1: For Image brightness enhancement, four regions of interest are computed for their mean and median.....	39
Table 5.1: A summary of classification results (mAP) per classifier per class.....	72
Table 5.2: A summary of classification results (mAP) for the base networks.....	73
Table 5.3: The means of IOUs for different predictive models.....	73
Table 5.4: The loss (x100) against the prior knowledge.....	74
Table 5.5: The results by using different update schemes, measured by using total loss for each predictive model	74
Table 5.6: R-CNN classification performance metrics by using the validation set.....	75
Table 5.7: Fast R-CNN performance metrics for virus detection by using the validation set.....	76
Table 5.8: Faster R-CNN metrics by using the validation set.....	76
Table 5.9: The metrics for evaluating YOLO classification by using the validation set.....	77
Table 5.10: The metrics for evaluating SSD classification by using the validation set.....	78

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgments), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Signature:

Date: 03 August 2020

Acknowledgment

First of all, I would like to appreciate my supervisor Wei Qi Yan. Dr Yan provided me with extremely professional academic guidance. I also have benefited from the theoretical lectures delivered by my supervisor regularly. He offered not only timely academic answers but also continuously spiritual encouragement through the Internet during lockdown time, which was the greatest motivation for me to complete this research thesis.

Secondly, I must express my gratitude to my friend, Ms Lu Luo, who gave extremely professional guidance on virology and pathology from the academic perspectives. This filled a gap of my knowledge on virology.

Moreover, I would like to say thanks for the learning facilities provided by the Auckland University of Technology, such as libraries and laboratories, also facilitate my study and research.

Finally, I would like to thank my family for their encouragement of my studies and provide financial support for my research and living in New Zealand.

Luxin Zhang

Auckland, New Zealand

August 2020

Chapter 1

Introduction

This chapter mainly includes five parts. The first part primarily explicates background and motivation of automatic recognition of viruses from digital images by using deep learning methods. In Sections 1.2 and 1.3, we list the research problems to be probed in this thesis and make meaningful contributions to the field of deep learning. In Section 1.4, we expound the significance of this research and its implementation. Finally, the detailed content of this thesis and the context of each chapter will be outlined in Section 1.5.

1.1 Background and Motivation

Viruses are the most active lives on our earth, with millions of years of history, more than the sum of all others. The viruses have a variety of shapes, like ball, silk thread, bullet, brick, tadpole, etc. Some of these viruses are highly contagious and have posed a significant threat to public health.

SARS-CoV-2 has seriously threatened the safety of our human lives. By May of 2020, the new coronavirus (SARS-CoV-2) has infected about 5.1 million people and causes at least 333,000 deaths, and the numbers are predicted to continue growing in the future. It has not only caused severe economic panic but also raised panic for all humans. The CT imaging and nucleic acid detection are two effective testing methods. However, they have their own advantages and disadvantages, especially the nucleic acid test is not accurate inaccuracy. Thus multiple tests are needed before the confirmation of diagnosis (Jing, et al., 2020). Accordingly, how to quickly and accurately classify viruses, especially highly infectious viruses, is a must.

Deep learning is a branch of machine learning, which applies artificial neural networks as an architecture to conduct representation learning on well-collected data. Its advantage is to use unsupervised or semi-supervised feature learning as well as hierarchical feature extraction algorithm to replace manual feature acquisition. The application fields of deep learning (e.g., computer vision, bioinformatics, medical image analysis, material inspection) have produced significant contributions. The results obtained by using this technology are comparable to the performance of human experts, even exceed the performance of the experts. Convolutional neural networks, as one of the deep learning architectures, can provide more reliable results in image and speech recognition than other methods. CNNs provided the most advanced accuracy in various image recognition, including object recognition, segmentation, image super-resolution, object detection, etc.

In this thesis, virus detection using deep learning is proposed to identify the viruses with electron microscope images. Although the image recognition technology has become more and more experienced, there is still a lack of practice in automatic recognition of virus images under the microscope. Consequently, this research project aims to develop an automatic detection method based on deep learning. In this thesis, the images of four types of viruses will be taken into consideration: SARS, MERS, HIV, and COVID-19. Additionally, for the betterment of detecting viruses, in this research project, we will compare the performance of five deep learning models, namely, R-CNN, Fast R-CNN, Faster R-CNN, YOLO, and SSD, for the purpose of identifying viruses from microscope images.

1.2 Research Question

The main research question of this thesis is related to classification and bounding box regression for multiple viruses appeared on electron microscopy images. This is a typical mission in computer vision. In this research project, we have identified the following issues that need to be resolved.

Different from image classification problems as illustrated in various computer vision competitions such as ImageNet, the provided images for virus detection were derived from the electron microscope and have distinguished visual displays. The derived images are often grey scaled and noisy due to hardware limitations. Prior to training, a number of operations of data augmentations should be conducted to improve the data quality.

In order to achieve better prediction results in this research project, we take advantage of known virus morphological characteristics. A detailed examination of the knowledge is required to quantify the morphological characteristics, so that deep learning algorithms can utilize the knowledge to process image data.

There are five predictive models in this research project: R-CNN, Fast R-CNN, Faster R-CNN, YOLO, and SSD. The networks are pretrained on different datasets such as

ImageNet. There exists a compatibility issue between the predictive models and input data, a number of network modifications are made to permit the models to be more fit to the virus classification and bounding box regression problem.

1.3 Contributions

We would like to list the three contributions of this thesis as:

- A comprehensive summary of the visual features of virus images from electron microscopy

The electron microscopy images and virus morphological information are reviewed in this thesis; accordingly, image augmentations with multiple operations are implemented to make better utilization of the source images.

- A term of novel virus morphological loss function

The objects in this thesis are not agnostic, which is possible to include the prior knowledge for prediction. The approach is a comparison between predicted objects confined in bounding boxes with visual features.

- Implemented the state-of-the-art technology of deep learning

There have been numerous novel image processing proposals in recent years. In this research project, we take into account the latest developments and implement a few were available, such as recently proposed activation functions and attention mechanism.

- Multiple virus recognition models

R-CNN, Fast R-CNN, Faster R-CNN, YOLO, and SSD are tested in this research. A number of modifications are made to achieve better object prediction results within the virus bounding box for resolving the object anchoring and object classification problem.

1.4 Objectives of This Thesis

Automatic recognition of virus particles by using deep learning methods is studied in this research project, this thesis aims to illustrate the motivations, review the trending technology pertaining to this research topic, and what this experiment has achieved in the given context.

The topics covered in this thesis are background introduction, literature review, virus morphological characteristics that are beneficial to object detection, visual features of electron microscopy images, deep learning implementations for the virus detection as well as the betterment for the deep learning networks.

Based on the practical methods for analysing the electron microscopy images, the performances of proposed predictive models are evaluated. In this thesis, we launch a number of discussions on the effectiveness of the proposed solutions. Additionally, in this thesis, we share knowledge of experimental limitations and our future work.

1.5 The Structure of This Thesis

This thesis is arranged in seven chapters: Introduction, literature review, methodology, model training, experimental results, resultant analysis and discussions, conclusion and future work.

The first two chapters (i.e., introduction and literature review) present the motivations of this research work and an introduction of the latest technology developed based on the relevant subjects.

The next chapter is the methodology. This chapter provides an elaboration of a novel loss term that reflects prediction errors against well-known virus morphological information and selected predictive models as well as the modifications to the models for compatibility, where the predictive models can better reflect the information that the source images contain. Data preprocessing is discussed in this chapter.

The training section is arranged in a chronological sequence. Not only does this chapter discuss the proposed hyper parameters, but also it illustrates modifications to the networks in practice when training.

Our experimental results follow the training chapter, which introduces the selected evaluation methods and lists performance metrics reflecting different research interests in the virus particle classification and bounding box regression problem.

The next chapter (i.e., analysis and discussions) is detailed based on the results that shed light on possible reasons why virus detection is failed the predictive models. Our significant statistical results, e.g., superior performance of a particular predictive model than others, are explained.

The last chapter (Conclusion and Future Work) briefs the experiment limitations and future work that aims to address the listed limitations for expected better networks and broad implementations.

Chapter 2

Literature Review

With a comprehensive examination of the research questions and reasonable reviews of the previous studies, the focus of this chapter is on the performance of five deep learning networks, namely, R-CNN, fast-CNN, Faster R-CNN, YOLO, and SSD.

2.1 Introduction

With the spread of COVID-19 pandemic, reagents are lack of supply, and the testing is slow and widespread infection among the public is becoming increasingly severe. At present, the most popularly utilized detection methods are to find target DNA, RNA, and unique proteins through various detection methods. However, there is insufficient to meet the requirements of people around the world. Thus, how to identify the virus quickly and accurately is becoming increasingly urgent.

In recent years, deep learning, especially deep convolutional neural network, has rapidly developed into a research hotspot in the field of medical image analysis (Shen, et al., 2017). At the same time, with the continuous development and promotion of the technology in the fields of computer vision and medical imaging, the computers science for medical image analysis have become an indispensable tool and technical means for clinical disease diagnosis, medical research and treatment (Miller and Brown, 2018). Accordingly, it resembles that deep learning techniques could be tried to identify viruses.

The two events inspired the research experiment, which is based on deep learning for the detection of virus particles in microscopic images. Furthermore, due to the lack of deep learning technology to achieve automatic virus identification, this thesis will conduct an experimental comparison of several popular image detection models in terms of R-CNN, Fast R-CNN, Faster R-CNN, SSD, and YOLO.

2.2 Target Morphological & Electron Micrograph Characteristics

Virus detection from digital images is one of the common subjects of object detection and image recognition. Nevertheless, learning and detecting the virus have invariably been one of the challenging and complex tasks because it is challenging to learn and detect via computer systems with different noise levels (Shakri, et al., 2017). In this project, we

will leverage virus morphological characteristics (on electron microscopy images) for virus recognition by using deep learning. Useful features are angular second moment, contrast, entropy (Haralick, Dinstein, & Shanmugam, 1973) selected features by filtering (Wu, Chen, & Hsieh, 1992), texture information (Kylberg & Sintorn, 2011)(Tuceryan & Jain, 1998)(Nanni, Paci, Brahnam, Ghidoni, & Menegatti, 2013), including the assessment of intensity level of pixel information (Ojala, Pietikäinen, & Mäenpää, 2002).

Major virus morphological characteristics are shown as spherical shape and scattered “white dots” for spike protein projections. Coronavirus family shares substantial similarity in morphology while HIV is relatively different. However, they all have protein projections on the viral envelope. These features can be utilized, at least, for distinguishing viruses from irrelevant objects, due to disturbances from the background.

The electron micrography images are generated by using the electron microscope that radiates a beam of electron electrons as a source of the illumination, with the wavelength of an electron up to 100,000 times shorter than that of human visible light photons, and the resulted image resolution can be 50 pm (Erni, Rossell, Kisielowski, & Dahmen, 2009), enabling magnifications above 50 million times. The produced images are usually greyscaled (brightness levels) (Hortolà, 2010), colourization of electron microscopy images adds no new information to the greyscale specimen other than aesthetics.

Multi-detector can be used to combine various specimen properties into one single pixel (Antonovsky, 1984). Different attributes representing the different aspect of information and colourization can be conducted accordingly for each primary colour representing one information channel (Danilatos, 1986). Nevertheless, the images from the electron microscope are brightness level representation as the received energy feedbacks from an electron beam on a specimen do not include wavelengths that are semantic in human perception to visible light.

Given the nature of electron micrography image generation, all produced images are in greyscale. However, when encoded in a digital format out of distribution concerns, images are of RGB colours. In this research project, we consider this issue and one-

channel greyscale images are derived in data preprocessing, as multicolour information in general serves for aesthetic purposes rather than reflecting actual virus existence. Since all images are obtained via the electron microscope, it is unlikely that visual features are highly non-linear, low-level features extracted from CNN should be scale invariant across all virus particles on electron microscopy images (Denton, Zaremba, Bruna, LeCun, & Fergus, 2014) (Xue, Li, & Gong, 2013). In other words, image distortion during the pre-processing data phase should be exerted with caution, as in reality highly scale variant visual features in electron microscopy images do not exist (Oho, Ichise, Martin, & Peters, 1995).

2.2.1 HIV Morphological Characteristics on Electron Micrograph

HIV (Human Immunodeficiency Virus) is a member of the genus *Lentivirus* and share similar morphological properties. It has two species: HIV-1 and HIV-2 (Gilbert, et al., 2003). Inside the viruses, envelopes are two copies of positive-sense single stranded RNA responsible for encoding nine viral genes that are enclosed by 2,000 copies of conical capsid virus protein P24 (Compendium, 2008). Viral protein P17 surrounds the capsid aiming to preserve the integrity of virion particle (Compendium, 2008), viral protein P17 is enclosed by viral envelope comprised of lipid bilayer derived from host cell membrane when virus particle buds erupt from host cells, taking with glycoprotein120 (Chan, Fass, Berger, & Kim, 1997) (Klein, Bjorkman, & Rall, 2010).

The protein projections on the viral envelope are *N*-linked glycans, a type of chemical compound consisted of a substantial number of monosaccharides connected glycosidically, and are of high density in distribution on the viral surface. The molecular structure of the protein projections can be revealed by electron microscopy (Lyumkis, et al., 2003). The projections present as spikes on electron microscopy images.

HIV morphological appearance is roughly spherical usually about 120 nm of diameter (McGovern, Caselli, Grigorieff, & Shoichet, 2002). An electron microscopy image of the HIV virus particle is shown in Figure 2.1.

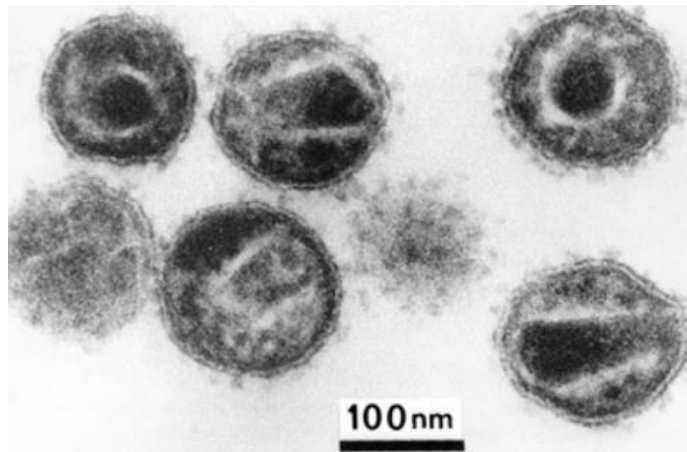


Figure 2.1: Electron micrograph of HIV

2.2.2 SARS-CoV-1 Morphological Characteristics on Electron Micrograph

SARS (Severe Acute Respiratory Syndrome) is a member (species) of coronavirus family and shares morphological similarities (Wong, et al. 2019), appeared in electron micrograph as pleomorphic spherical particles with bulbous surface projections (Goldsmith, et al., 2004). Physical size of one virus ranges from 50 to 150 nm, excluding spikes (Neuman, et al., 2006). Typical viral envelope is comprised of a lipid bilayer in which membrane, envelope and spike proteins are anchored, forming bulbous surface projections (Michael, Cavanagh, & Lai, 1997). Nucleocapsid inside viral envelope protects 30-kb (kilo base pair) RNA genome, constructed in a continuous beads-on-a-string conformation manner (Fehr & Perlman, 2015) (Chang, et al., 2014). The spike protein of SARS-Cov-2 shows a sequence similarity of 76%~78% with that of SARS-Cov-1 (Rabaan, et al., 2020).

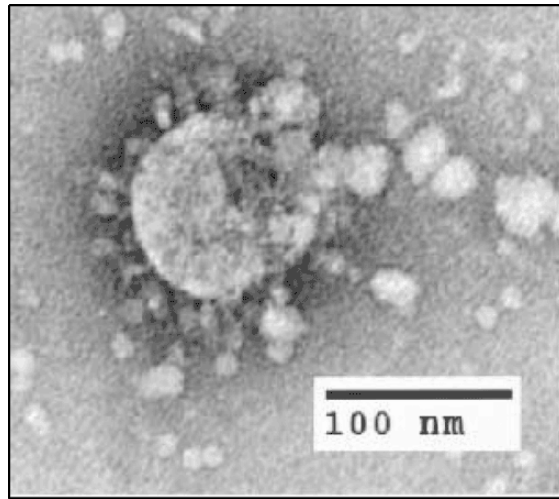


Figure 2.2: Electron micrograph of SARS coronavirus

2.2.3 MERS-CoV Morphological Characteristics on Electron Micrograph

MERS-CoV (Middle-East Respiratory Syndrome-Related Coronavirus) (Groot, et al., 2013) causes MERS (Middle-East Respiratory Syndrome), also known as camel flu. It is a coronavirus species (Saey, 2013). Similar to other coronavirus structure, MERS appears crown-like structures on electron microscopy images, with positive-stranded RNA as genomic material and an outer envelope (McIntosh, Dees, & Becker, 1967)(Masters, 2019). The genome is enclosed inside the nucleocapsid, helical in shape. Glycoprotein spikes scatter on the virus's surface.

In comparison with COVID-19 and SARS, MERS is the most distinct in terms of genetic composition (Rabaan, et al., 2020). The length of Spike proteins on SARS and MERS is shorter than that of COVID-19, which is a noticeable visual feature in electron microscopy images (Rabaan, et al., 2020).

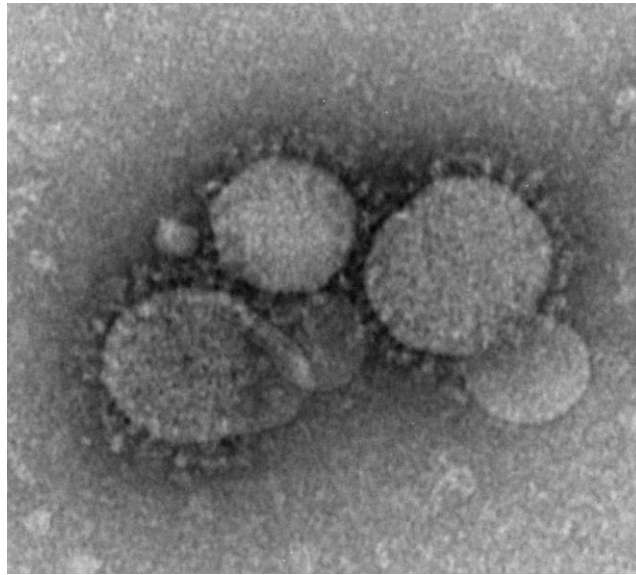


Figure 2.3: An example of MERS particles under the electron microscope (in 100 nm)

2.2.4 SARS-COV-2 Morphological Characteristics on Electron Micrograph

Severe Acute Respiratory Syndrome Corona Virus 2 (SARS-COV-2), or COVID-19 given the year of discovery, is a novel severe acute respiratory syndrome coronavirus (Johnson, et al., 2020). COVID-19 is similar to SARS in terms of morphological appearance on electron microscopy images. Four typical proteins are identified Spikes (S), Envelope (E), Membrane (M) and Nucleocapsid (N) (Zhang, et al., 2020). One recognized electron microscopy feature is spiked imageable at an atomic level (Wrapp, et al., 2020).

SARS-COV-2 morphological appearance is roughly spherical about 50 - 200 nm of diameter (Chen, et al., 2020). An electron microscopy image of COVID-19 virus is shown in Figure 2.4.

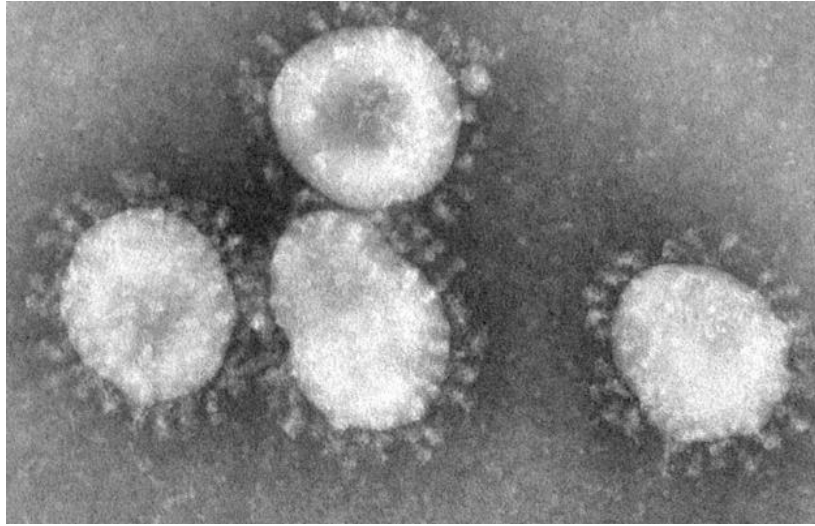


Figure 2.4: Electron micrograph of COVID-19

2.3 Deep Learning

Deep learning, also known as deep neural learning or deep neural network, is an artificial intelligence method that mimics how the human brain works when processing data and creating patterns for decision making (Samek, Wiegand, & Müller, 2017). Furthermore, deep learning is a subset of machine learning that trains a computer to perform human-like tasks, such as recognizing speech, identifying images or making predictions (Ishtiaq, et al., 2020). It utilizes hierarchical artificial neural networks to complete the process of machine learning. The physical structure of the artificial neural network is like the brain of the human, and neuron nodes are connected like a network (Nie, Gao, Wang, & Shen, 2018), its network can learn from unstructured or unlabelled data without supervision.

- **Artificial Neural Network**

Artificial neural network (ANN) is inspired by biological neural networks where neurons release energy signals to inform other neurons (Chen, Lin, Kung, Chung, & Yen, 2019), and collectively they represent the flow of information (Guimarães & McGreavy, 1995). The concept was first proposed (McCulloch & Walter, 1943) by following Hebbian learning and invention of perception (Rosenblatt, 1958). The modern ANN (Schmidhuber,

2015) is proposed with the philosophy (Willis, Montague, Di Massimo, Tham, & Morris, 1992) that the algorithm should be capable of “learning” from given observations (Kleene, 1956). From a graph perspective, an ANN is a directed weighted graph that input neurons are weighted and connected with next layer neurons (Zell, et al., 1994). There are a host of extensions of ANN such as CNN and RNN, and they will be discussed in the following paragraphs.

Neuron is the basic unit of ANN. The internal structure includes weighted links that take into account source inputs and a sum operator (sum is the most selected operator) that adds all weighted input information. The thresholding is optional. An activation is applied to map the input to an output. Neurons in the first layer take input data such as documents and images, while neurons in the last layer output semantic probability for a given task.

A typical network consists of connections, and each connection provides the output of one neuron as an input to another neuron. In addition, each connection is assigned a weight (connections are weighted links) that represents its relative importance (Zell, et al., 1994). A neuron has multiple input and output connections (Abbod, 2007).

A propagation function treats the input to a neuron from the outputs of its predecessor neurons. The connections in between are weighted. A bias term can be linearly added to the result of a propagation function (Dawson & Wilby, 1998). This is used to simulate signal energy released by biological neurons, e.g., sigmoid and ReLU (Nair & Hinton, 2010).

- **Convolutional Neural Network**

Convolutional Neural Network (CNN) is widely used in image and video analysis (Valueva, Nagornov, Lyakhov, Valuev, & Chervyakov, 2020), which is the backbone technology for various network proposals in this research. It employs convolution for matrix multiplication to process multi-dimensional information (Goodfellow, Bengio, & Courville, 2016). The invention of CNN draws inspiration from biological processes

(Hubel & Wiesel, 1968), where artificial neuron connectivity resembles that of animal visual cortex (Fukushima, 1980), in which cortical neurons react to stimuli on restricted vision-functional regions known as receptive fields (Matusugu, Mori, Mitari, & Kaneda, 2003).

CNN utilizes high dimensional filters as receptive fields to an artificial neuron to specific stimuli and filter sizes vary in different layers as defined in hyper parameters. A convolutional layer is efficient in processing clustered semantic information than by a fully connected layer (Aghdam & Heravi, 2017).

Due to considerable computation cost given high dimensionality, pooling significantly reduces input dimensions by combining the outputs of a neuron cluster into single value output. Pooling layers usually follows convolutional layers (Ciresan, Meier, Masci, Gambardella, & Schmidhuber, 2011). Typical pooling methods are maxed operator by selecting max elements (Wang et al., 2018) and average operator by computing the mean values over inputs (Mittal, 2020). Some famous novel CNN structure proposals are GoogleNet with inception modules and ResNet that takes into account cross-layer information transmission.

- **Recurrent Neural Network**

Recurrent Neural Network (RNN) (Williams, Hinton, & Rumelhart, 1986) includes latent information between samples with positional relationships by forming a directed graph (recent RNN proposals see the complex flow of information instead of a naïve directed graph) along with a temporal sequence of inputs. Thus, RNN is capable of using internal memory to process sequential inputs, and this property is useful in image processing tasks (Graves, et al., 2009).

The significant contribution of this network is the inclusion of a time-varying activation mechanism (Miljanovic, 2012) that considers input with an additional dimension: time, and accordingly additional weight matrices are introduced for each neuron to process the extra dimension information (Elman, 1990). Neurons of RNN are

arranged in a successive manner (Jordan, 1997) that the neurons in the layer h_t given the input x_t at the t -th input are connected to the layer at the previous state h_{t-1} , and will influence the state of the layer h_{t+1} given the next input x_{t+1} (Zhang *et al.*, 2018).

The variant of RNN are Long Short Term Memory (LSTM) that adds the complexity of different “gates” to control information memory in respects of time (Hochreiter & Schmidhuber, 1997), and Gated Recurrent Unit (GRU) that attempts to simplify computation with fewer gates but with comparable performance (Cho, et al., 2014).

2.4 R-CNN

R-CNN (‘R’ for region proposal) is selected in this experiment for virus particle recognition and localization. As discussed in the previous sections, a number of modifications are made to the loss function. In this research project, we test different activation functions on R-CNN, but the structure is unaltered.

R-CNN proposes image regions of interest in an agnostic manner by using selective search (Girshick, Donahue, Darrell, & Malik, 2017) (Uijlings, Sande, Gevers, & Smeulders, 2013) (Wang, Yang, Zhu, & Lin, 2013). Regions of interest appear in semantic colour, shading, texture, morphological characteristics, scale, etc., selective search approaches solve this problem by using hierarchical grouping (Uijlings, Sande, Gevers, & Smeulders, 2013). It takes into account bottom-up grouping as a hierarchical approach (Comaniciu & Meer, 2002) (Chen, Liu, Tuzel, & Xiao, 2016) that firstly proposes a number of initial object location hypotheses by using a fast segmentation method (Felzenszwalb & Huttenlocher, 2004), then merges neighbour regions with high levels of similarities until only one region is presented for an image.

In the experiment, the region proposal mechanism in R-CNN remains unchanged. The proposed image regions are warped before being fed into a base CNN consisting of five convolutional neural networks and two fully connected networks (Krizhevsky, Sutskever,

& Hinton, 2012), which are scored by SVM applied with a greedy non-maximum suppression that rejects a region for a high IoU (Intersection Over Union).

The size of the input to the base CNN is $227 \times 227 \times 3$ for RGB encoded images, which is consistent for all R-CNN implementations. To demonstrate how this proposal is applied to virus recognition and localization with prior knowledge in comparison to the base CNN structure, in this research, we take into consideration of the construct of the same CNN as stated in the early study (Girshick, Donahue, Darrell, & Malik, 2017) (Krizhevsky, Sutskever, & Hinton, 2012), but it is different in terms of input sizes since the source images contain less information (in grey scale with limited numbers of objects in question). There are two image sizes tested: 64×64 and 128×128 . Different sizes of images have distinct levels of granularities by mandating different sizes of input, it is expected that if granularities are in any way conducive to high accuracy prediction, these tests can confirm this hypothesis.

For the implementation of SVM (support vector machine) with R-CNN, which is used for scoring features for each class, a modification is made to the SVM loss function to reflect misclassifications on the virus. A typical SVM loss function is given as

$$Loss = \frac{1}{2} ||w||^2 + C \sum_i \xi_i$$

subject to (s. t.)

$$y_i(w^T x_i + b) \geq 1 - \xi_i \quad (2.1)$$

where C represents the cost to control sensitivity to misclassified data sample x_i corresponding to label y_i , w is the weight for data samples, $||w||$ should be minimized to maximize hyperplane margin, ξ_i is a slack variable for error tolerance. Given the prior knowledge, the loss function is expressed as equation (2.2).

$$Loss = \frac{1}{2} ||w||^2 + C \sum_i \xi_i + L_{coronaV}(y'_i, y_i, t'_i)$$

s. t.

$$y_i(w^T x_i + b) \geq 1 - \xi_i \quad (2.2)$$

where $L_{coronaV}(y'_i, y_i)$ is the loss function to impose penalty on wrong classifications by SVM. Non-maximum suppression (NMS) (Hosang, Benenson, & Schiele, 2017) is used to reduce the number of overlapped bounding boxes by merging bounding boxes based on confidence score S and an overlap threshold λ_{nms} , (Felzenszwalb, Girshick, McAllester, & Ramanan, 2009) (Girshick, Donahue, Darrell, & Malik, 2017) (J. Redmon, S. Divvala, Girshick, & Farhadi, 2016) (Liu, et al., 2010) (Girshick, et al., 2015)

For the NMS implementation, the threshold λ_{nms} is set at 0.8. The empirical study on the results determines this, $\lambda_{nms} = 0.8$ is optimal in discriminating bounding boxes of different visual objects.

R-CNN records the highest mAP (mean Average Precision) on VOC 2010 test, leading by nearly 10% in mAP (Girshick, Donahue, Darrell, & Malik, 2017). This study (Girshick, Donahue, Darrell, & Malik, 2017) experimented on different depths of CNN feature layers and shows the effectiveness of deep neural network (Krizhevsky, Sutskever, & Hinton, 2017). In this experiment, R-CNN was tested given different conditions and their performances are recorded for evaluation.

2.5 Fast R-CNN

Fast R-CNN (Girshick, 2015) is an improvement of R-CNN with achieved higher computation speed as well as prediction accuracy. In order to adapt Fast R-CNN for virus recognition and localization, we summarize novel proposals in Fast R-CNN and test other alternatives to see if there is any improvement. This research made a number of modifications to the internal structure of Fast R-CNN for adaptation to the virus work.

- **Region of Interest (RoI) and Pooling Layer**

One major novelty of Fast R-CNN is at the pooling layer. In Fast R-CNN, an $H \times W$ grid of rectangular windows with each cell having a size of (approximate) $h/H \times w/W$ is fed into a max pooling layer, where filters extract information by using max operator on each subwindow. This implementation draws similarities to pyramid SSPnets (spatial pyramid pooling networks) (He, Zhang, Ren, & Sun, 2015) except that there is only one pyramid level.

There are no conclusive evaluations in the literature review (Gua, et al., 2018) regarding relative superiority in feature extraction for max and average pooling method, in this thesis, we test another pooling method (average) to see whether there is any improvement in prediction accuracy. However, it is worth noticing that the performance is slightly dropped in comparison with that from max pooling method. Hence, this thesis kept the max pooling method unchanged for all Fast R-CNN implementations.

- **Pretrained ImageNet Networks**

It should be noted that the source data is responsible for ImageNet, network training does not include viral objects on electron microscopy images. Thus, extracted features from ImageNet networks do not represent objects in this experiment. Initialization requires to take into account pretrained ImageNet (Marmanis, Datcu, Esch, & Stilla, 2015) networks. A network is retrained in the same way as to how an ImageNet network is trained for general object classification and localization. In detail, given a pretrained network (e.g., Fast R-CNN), the network consists of five convolutional layers and five max pooling layers), this proposal simply adds two fully connected layer together for the structure being resembled in this proposal (Krizhevsky, Sutskever, & Hinton, 2012), the training on source data (virus particles on electron microscopy images) for classification. It is expected that the retrained network is adjusted for visual features presented on virus electron microscopy images.

In Fast R-CNN, the last max pooling layer of the pretrained network is replaced by using an RoI pooling layer which is organized through setting H and W to be compatible

with the input shape of the pretrained network at its first dense layer. This research work has included that how regions of interest from the 2,000 electron microscopy images with a size of H and W should be adapted for training, the input and output layers for feature extraction should be compatible with electron microscopy source images as well as proposed regions.

Another novel proposal in Fast R-CNN is the last dense layer and softmax of the pre-trained network being replaced with a new dense layer and softmax for class prediction in association with bounding box regression. The task in this experiment is as same as in Fast R-CNN, there is no modification to the network output (for category prediction and bounding box regression).

- **Multitask Loss Function**

Multitask loss function refers to two sibling output layers for category prediction and bounding box regression. Given K categories for prediction, one output layer produces a discrete probability distribution over $K + 1$ categories per RoI by softmax, $y' = (y'_0, y'_1, \dots, y'_K)$. The other is bounding box four-tuple $t^k = (t_r^k, t_c^k, t_h^k, t_h^k)$, where $k \in \{1, 2, 3, \dots, K\}$ as the category label index. A multitask cost L on regions of interest for object classification and bounding box regression is expressed as

$$L(y', y, t_u, v) = L_{cls}(y', y) + \lambda[u \geq 1]L_{loc}(t^u, v) \quad (2.3)$$

where $L_{cls}(y', y) = -\log y'_y$ stands for the logarithmic loss function for a true class y given the proposed probability y' . The second term is for localization denoted as L_{loc} over a tuple of the anchor and shape descriptor of a predicted bounding box $t^k = (t_r^k, t_c^k, t_h^k, t_h^k)$ and ground truth bounding box $v = (v_r, v_c, v_h, v_w)$ for the class y .

$$[u \geq 1] = \begin{cases} 1, & u \geq 1 \\ 0, & otherwise \end{cases} \quad (2.4)$$

For bounding box regression loss, the computation is given as

$$L_{loc}(t^u, v) = \sum_{i \in \{r, c, h, w\}} smooth_{L_1}(t_i^u - v_i) \quad (2.5)$$

where

$$smooth_{L_1}(x) \begin{cases} 0.5x^2, & |x| < 1 \\ |x| - 0.5, & otherwise \end{cases}$$

where L_1 loss is less sensitive to outliers in comparison to that of L_2 as implemented in R-CNN and SPPnet. There exists a hyper-parameter λ to reach a trade-off between the two task losses. $\lambda = 1$ is set for an equally balanced loss computation strategy.

Given prior morphological knowledge of the objects in question, it is feasible to take advantage of such knowledge as illustrated in the aforementioned chapters. By introducing a new loss term $L_{coronaV}(y'_i, y_i)$, to reflect prior knowledge, there is

$$L(y', y, t_u, v) = L_{cls}(y', y) + \lambda_1[u \geq 1]L_{loc}(t^u, v) + \lambda_2 L_{coronaV}(y', y, t'_i) \quad (2.6)$$

where λ_1 and λ_2 are weight parameters for controlling the importance of $L_{loc}(t^u, v)$ and $L_{coronaV}(y', y)$. $\lambda_1 = 1$ is set as same as Fast R-CNN implementation. $\lambda_2 = 0.01$ is set. The reason for $\lambda_2 = 0.01$ is explained in the analysis chapter.

Generally speaking, in this research project, we impose this loss function for training to punish predictions that are vastly contradictive to what prior morphological knowledge would suggest, but do not want to overwhelm $L(y', y, t_u, v)$ with this kind of costs, so that the network training would still be concentrated on classification and localization errors.

- **Mini-batch sampling**

Minibatch sampling in Fast R-CNN sets $N = 2$ images for each batch and $R = 128$ that gives 64 regions of interest per image per mini-batch. Only does the proposed region with IoU overlap λ of equal to or more than 0.5 with ground truth regions being taken into consideration. Images are horizontally flipped on the probability condition of 0.5 and there is no other data augmentation implemented.

In this experiment, minibatch size is adjusted so that the training for Fast R-CNN model can better fit the virus classification and localization work. The details of these specifications are elaborated in the Training Section.

- **Backpropagation through the region of interest pooling layers**

Backpropagation through the region of interest pooling layers is important in loss convergence. Hereinafter, we define $x_i \in \mathbb{R}$ as the i -th input to a region of interest layer and y_{rj} as j -th output of this layer for the r -th region of interest, $y_{rj} = x_{i^*(r,j)}$, where $(r, j) = \operatorname{argmax}_{i \in R(r,j)} x_i'$ and $R(r, j)$ is denoted as the index set of inputs.

The derivative of the pooling layer with respect to x_i is

$$\frac{\partial L}{\partial x_i} = \sum_r \sum_j [i = i^*(r, j)] \frac{\partial L}{\partial y_{rj}}. \quad (2.7)$$

The Fast R-CNN implementation in this thesis does not propose changes to this back-propagation technique; however, it is worth noting that $L_{coronaV}(y', y)$ does not discriminate between different regions. $L_{coronaV}(y', y)$ is treated as a simple representation of contradiction to prior virus morphological knowledge.

For weight initialization and learning, the initialization of weights takes zero-mean Gaussian distributions with standard deviations of 0.01 and 0.001 for softmax classification and bounding box regression predictors, respectively. Biases are initiated to 0. Global learning rate is set at 0.001 and decayed to 0.0001 after a few training iterations. In this experiment, we test a number of learning rates.

- **Sample source image method**

Scale invariance in statistics is a concept describing an object or law being persistent in an aspect of scale, energy or other attributes are “dilated” or “compressed” (multiplied by a common factor). Fast R-CNN utilizes an image pyramid to sample source image data (Girshick, et al., 2015). In Fast R-CNN implementation, the sampling method is similar to that of Fast R-CNN, the image pyramid is in proportion to image or input sizes.

2.6 Faster R-CNN

Faster R-CNN (Ren, He, Girshick, & Sun, 2015) is a further improvement of R-CNN to achieve faster computation speed and higher prediction accuracy. It proposes a Region Proposal Network (RPN) in combination with Fast R-CNN, and incorporates attention (Chorowski, Bahdanau, Serdyuk, Cho, & Bengio, 2015) mechanisms.

A Region Proposal Network (RPN) is a deep convolutional neural network that makes the practice of images with arbitrary size and returns one or multiple of rectangular bounding boxes that identify object locations (Long, Shelhamer, & Darrell, 2015). Faster R-CNN is based on a number of base neural networks with different levels of shareable convolutional layers (Simonyan & Zisserman, 2014) (Matthew & Fergus, 2014) whose final feature layers are processed and mapped to a low dimensional feature map, by which two sibling fully connected layers are trained for bounding box regression and box classification, respectively.

Similar to region proposals of Fast R-CNN based on extracted feature map, Faster R-CNN relies on convolutional feature map output. As suggested in this study (Krizhevsky, Sutskever, & Hinton, 2012), the deep neural network is more likely than a shallow one to record high accuracy prediction, with the help of novel networks such as ResNet (He, Zhang, Ren, & Sun, 2016), VGG (Simonyan & Zisserman, 2014), and GoogLeNet (Szegedy, et al., 2015). That recorded the best prediction result in various global competitions, the deep networks for feature map output are tested for virus morphological feature extraction. The feature extraction process is as same as what has been proposed in Faster R-CNN (Ren, et al., 2015), two sibling networks are taken into consideration for the control of the output features for classification and localization work.

The deep neural networks are pretrained based on ImageNet and retrained based on the electron microscopy images for virus particle recognition. There are various proposals of ResNet, VGG, and GoogleNet, which are different in terms of the number of layers and other minor modifications, e.g., applied activation functions. The selected variants of

ResNet and VGG are ResNet152 and VGG11. The considerations are hardware limitations and prediction accuracy, the determination of selections is a tradeoff between them. The details of the network specifications for training and classification results are illustrated in model training and resultant analysis further.

Anchors for a bounding box have consisted of four-tuples (r, c, h, w) so that k sliding windows should have $4k$ outputs. An anchor (r, c) is centred at a sliding window. By default, Faster R-CNN adopted three scales and three aspect ratios ($width/height$) that give nine anchors for each sliding position. Translation invariance refers to a concept in geometry (computer vision) that object does not change in terms of semantic representation (still recognizable) when points (pixels) move by the same ratio, this is addressed by using the implementation of multiscale and multiracial anchoring.

More ratios between *width* and *height* are expected to enhance the capability of neural networks related to visual features. However, due to hardware limitations as well as the prior knowledge of virus morphological features and “white dots” on electron microscopy images, there is no demand of increasing the number ratios to high, as simple rectangles with nine different shapes (Ren, et al., 2015) are sufficient to cover a virus particle fully. Despite keeping the number of ratios unchanged, there are minor modifications to ratio values, and again, given the shape of a typical virus particle being roughly spherical, a bounding box should be approximately square rather than a long rectangle. The loss function of RPN is designed to specifically target the two tasks: Bounding box regression and object classification.

$$L(y'_i, t'_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(y'_i, y) + \lambda \frac{1}{N_{reg}} \sum_i y' L_{reg}(t'_i, t_i) \quad (2.8)$$

where i is an anchor index, y' and t' are predictions of classification and bounding box regression, respectively. Accordingly, y and t are the ground truth for the object label and anchored bounding box. L_{cls} is logarithmic loss and L_{reg} is smooth L_1 . N_{cls} and N_{reg} are two normalization terms. The two types of loss are weighted by $\lambda = 10$ to exert importance on bounding box regression errors.

Given Faster R-CNN loss function, we introduce a new loss term $L_{coronaV}$, because general virus morphological characteristics are prior knowledge, when a detection mechanism discovers prominent outlier vision features, e.g., viruses are roughly spherical but a detected object enclosed in a bounding box is a polygon which is vastly dissimilar from a circle. Consequently, the new loss function is expressed as

$$L(y'_i, t'_i) = \frac{1}{N_{cls}} \sum_i L_{cls}(y'_i, y) + \lambda_1 \frac{1}{N_{reg}} \sum_i y' L_{reg}(t'_i, t_i) + \lambda_2 \frac{1}{N_{cls}} \sum_i L_{coronaV}(y'_i, y, t'_i) \quad (2.9)$$

where λ_1 and λ_2 are weights for the two loss items, respectively. λ_1 is set to 10 as same as Faster R-CNN (Ren, et al., 2015), λ_2 is assigned as 0.01, most detected objects are spherical and do not pose as bizarrely foreign objects resulting in significant great $L_{coronaV}$.

General information flow of Faster R-CNN model for virus classification and localization is as same as what it has been proposed in this study (Ren, et al., 2015). Firstly, the RPN is initialized by using the ImageNet pretrained model specially designed for region proposal tasks. Then, an object detection network is trained by using Fast R-CNN given the proposed regions of interest output from the first step. The two networks do not share convolutional features. Object detection network is used to initialize RPN and fine-tune the unique RPN layer.

2.7 You Only Look Once (YOLO)

YOLO (You Only Look Once), a novel CNN structure that has achieved fast computation as well as high prediction accuracy (Redmon, Divvala, Girshick, & Farhadi, 2015). A number of modifications are introduced to YOLO as well to make YOLO more suitable for the area of virus classification and localization on electron microscopy images.

Source image is separated into a grid consisted of $S \times S$ cells. If a cell contains part of an object, the cell is accountable for this object detection. Each cell renders bounding boxes and confidence scores for the object.

In order to preserve visual information granularity, the number of cells in the $S \times S$ grid is proportional to the input size of an image. In practice, there are two sizes of input images: 64×64 and 128×128 . The number of cells for 64×64 input images is 4×4 , for 128×128 , it is 8×8 .

The confidence score is a probability of how confident a model is relating to the bounding box enclosing the target object. The confidence is defined as

$$confidence = Prob(Object) \times IOU_{predict}^{truth}. \quad (2.10)$$

where IOU is the intersection area between the ground truth and the predict (represented by a percentage, that 100% is a complete overlap). The bounding box is the shared area of the two unions. $Prob(Object)$ refers to the probability of an object (of an arbitrary label) which is contained in a bounding box.

In the virus classification, each object should have one label of the class, that means, each virus as visual objects only belong to one species. The confidence is expressed as the probability of a bounding box containing an object (one virus particle) of a class

$$\begin{aligned} confidence &= \\ Prob(class_i) \times IOU_{predict}^{truth} &= \\ Prob(class_i | Object) \times Prob(Object) \times IOU_{predict}^{truth}. \end{aligned} \quad (2.11)$$

Consequently, each cell predicts a total of five parameters: four parameters (four-tuple) that define a bounding box's location and size (r, c, w, h) in which a virus particle resides, and the probability of the virus particle class.

In Figure 2.5, for a source image split into $S \times S$ cells, each cell is responsible for predicting the label of an object denoted as $Prob(Class_i)$ from C classes, and the four-

tuple (r, c, w, h) of B proposed bounding boxes. Hence, a predictive model result is of an $S \times S \times (5 \times B + C)$ tensor. In the case of virus classification and localization, given two input sizes for SARS, HIV and COVID-19, with specified two bounding box predictions for each cell, the outputs are with the sizes $4 \times 4 \times (5 \times 2 + 3)$ and $8 \times 8 \times (5 \times 2 + 3)$.

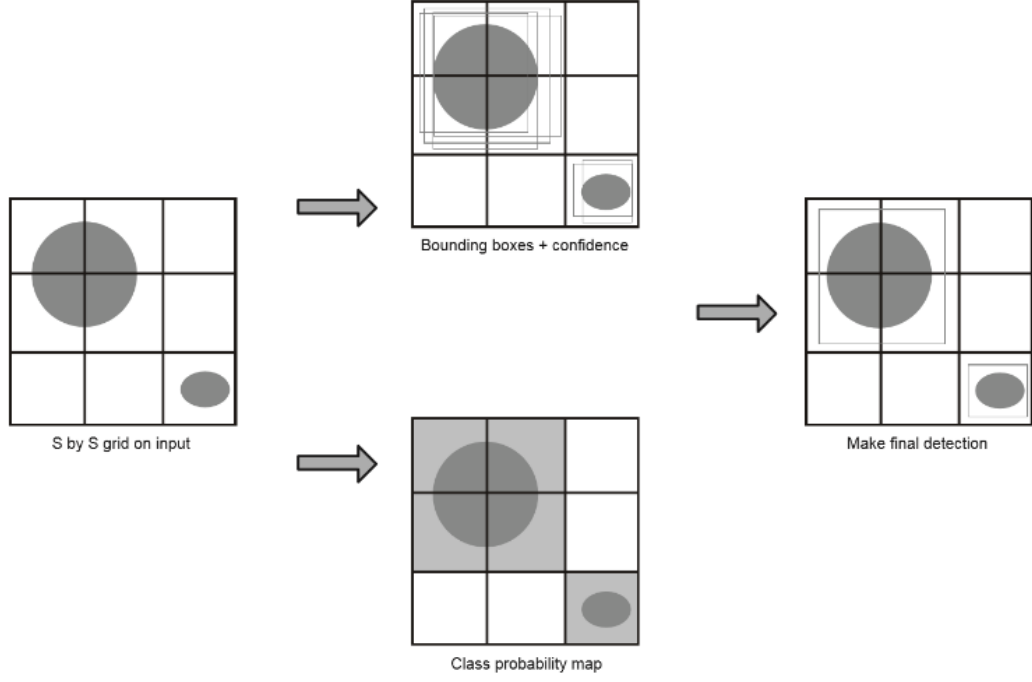


Figure 2.5: A typical YOLO process on an image

In YOLO (Redmon, Divvala, Girshick, & Farhadi, 2015), the adopted activation is Leaky ReLU. A number of other activation functions have been tested, and the performance results are recorded for evaluation. The implementation of YOLO loss function is illustrated as

$$L_{yolo} =$$

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{i,j}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] +$$

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{i,j}^{obj} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] +$$

$$\begin{aligned}
& \sum_{i=0}^{S^2} \sum_{j=0}^B I_{i,j}^{obj} [(C_i - \hat{C}_i)^2] + \\
& \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B I_{i,j}^{noobj} [(C_i - \hat{C}_i)^2] + \\
& \lambda_{noobj} \sum_{i=0}^{S^2} I_{i,j}^{obj} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 + \\
& \lambda_{coronaV} \sum_{j=0}^B L_{coronaV}(C_j, \hat{C}_j, RoI_j). \tag{2.12}
\end{aligned}$$

YOLO loss function consists of five components to impose penalties on wrong bounding box parameters r, c, w, h (denoted in this formula as (x, y, w, h)) and classification C for an image gridded into $S \times S$ cells with each cell responsible for predicting B bounding boxes, plus one loss term to penalize enclosed objects dissimilar from prior virus morphological knowledge. $\lambda_{coord} = 5$ and $\lambda_{noobj} = 0.5$ are scalars to control the penalties of bounding box coordinates and classification, respectively. $\lambda_{coronaV} = 0.01$ is as same as the networks mentioned above for the same reason. However, $\lambda_{coronaV}$ with recorded results is illustrated in the Results Section. RoI represents the four-tuple (r, c, w, h) that gives information about how to anchor a bounding box.

The model instability problems might be arisen given equally treating bounding box coordinate and object classification errors as cells without including any object (Redmon, Divvala, Girshick, & Farhadi, 2015). Penalty tends to zero in localization confidence scores. $I_{i,j}^{obj}$ is a binary operator (either output 0 or 1) that denotes the presence of an object for the i -th cell and the j -th proposal. It is reasonable to assume that the width and height of a bounding box should be tightly fit to the contour of an object so that heavy penalty for bounding box anchor (the four-tuple (r, c, w, h)) is applied to restrict the growth of large bounding box.

Prior to training, the base CNN for YOLO (all convolutional layers responsible for feature extraction) was first pretrained based on the source images. In the YOLO proposal (Redmon, Divvala, Girshick, & Farhadi, 2015) where the base CNN of YOLO was pretrained on ImageNet, we obtained the pretrained YOLO model and detached the base CNN from YOLO, and then we used the same method of how the first YOLO pretrains it is base CNN (Redmon, Divvala, Girshick, & Farhadi, 2015) by selecting the first 20 convolutional layers followed by an average pooling layer and a dense layer for classification, to pre-train the customized YOLO. The structure for the customized YOLO is totally identical as proposed in the first YOLO (Redmon, Divvala, Girshick, & Farhadi, 2015).

When information flows through convolution layers and then dense layers, a network does not take into account latent connections between extracted features before being flattened. What quite often is that images are comprised of various features that together build a concept of an object globally. Bilinear CNN (B-CNN) is proposed (Lin, RoyChowdhury, & Maji, 2015). B-CNN splits input matrices into two feature maps. The products of feature vector multiplication are flattened into a linear form, and the model then continues as what a typical dense layer would do. The matrix X from two streams $A = (a_1, a_2, \dots, a_n)$ and $B = (b_1, b_2, \dots, b_n)$ before linearization is shown as

$$X = \frac{1}{n} (\sum_{i=1}^n a_i b_i^T) + \varepsilon \quad (2.13)$$

The bilinear layer is adopted in YOLO between the last convolutional layer and the followed fully connected layer. The performance of YOLO with the added bilinear layer is offered for evaluation purpose. The predicted regions of interest are processed by non-maximum suppression (NMS) to reduce the number of bounding boxes.

2.8 SSD

Single Shot Detector (SSD) (Liu, et al., 2016) is a novel CNN structure that utilizes only one deep neural network for object detection (object classification and localization). The considerations of utilizing multi-scale features are reflected in the last layer of a base network, progressively decreasing in shape. The convolutional predictors are of various sizes with smallest ones to $3 \times 3 \times C$ (C for channel number, that in the case of this experiment, $C = 1$ for grey scale for electron microscopy images). Feature maps are downsized as network goes deep, and on each layer, there is a 3×3 filter convolving through the feature map for object detection.

A research study (Liu, et al., 2015) demonstrated high prediction accuracy for SSD in comparison with others such as YOLO given input images with relatively small sizes (low resolution indicative of a reduced level of visual information). One motivation of selecting SSD for this research project is that the electron microscopy images are often noisy even after data enhancement techniques such as denoising, and are low in terms of visual information content since the most visual features are simple morphological characteristics. SSD is proven well-performed for recognising objects given an information-scarce environment.

Default bounding boxes are configured to respect prior virus morphological information. The ratio between width w and height h of a bounding box vary by a lot in the SSD proposal (Liu, et al., 2015) for that SSD is trained on ImageNet dataset. Hence it is agnostic about objects. This is different in this thesis where virus as visual objects are already investigated for their morphological presence being roughly spherical. Given this understanding, the default bounding boxes for the SSD for virus classification and localization can be configured in advance to adjust w and h . In SSD terminology (Liu, et al., 2016), this is referred to as fixed default bounding box priors. The implemented fixed bounding boxes by using SSD (Liu, et al., 2016) include long rectangular (with high ratio between w and h , i.e., either high w/h or h/w), whereas, in this research project, the max ratio is set as 1.5 for the aspect ratio of the bounding box.

Similar to other base network implementations, SSD combines with VGG16 for feature extraction. VGG16 was pretrained on ImageNet dataset (Russakovsky, et al., 2015). In this research project, the pretrained VGG16 was retrained on the source images. The base network training is as same as what has been suggested in this work (Chen, Papandreou, Kokkinos, Murphy, & Yuille, 2014), that $fc6$ and $fc7$ are converted to convolutional layers and parameters are subsampled from $gc6$ and $fc7$; $pool5$ is transformed from $2 \times 2 - s2$ to $3 \times 3 - s1$ (for $size_{kernel} \times size_{kernel} \times stride$) with a trous (Holschneider, Kronland-Martinet, Morlet, & Tchamitchian, 1990) to fill “holes”. $fc8$ is removed and all dropout layers are disabled (no dropouts).

The improvements can be observed for SSD prediction in combination with other networks or with adjustments for specific contexts, that draw inspiration for this research. Deconvolutional SSD (Fu, Liu, Ranga, Tyagi, & Berg, 2017) records increased mAP over standard datasets such as PASCAL VOC and COCO dataset with added deconvolutional layers. RefineDet (Zhang, Wen, Bian, Lei, & Li, 2017) takes advantage of SSD and enhances the prediction capability via adjustments of anchors. An attention mechanism is used that is dedicated to text region image detection (He, et al., 2017). A feature-focused network with a bidirectional network instructing semantic feature circulation saw increases in accuracy (Wang, et al., 2019). For human face detection, a context-assisted SSD is constructed with novel contextual anchors (Tang, Du, He, & Liu, 2018).

Given the knowledge as mentioned above of possible combinations with other networks for prediction improvements, other state-of-the-art CNNs are tested as the base networks, such as GoogleNet and ResNet. The pretrainings of GoogleNet and ResNet are similar to that of VGG16, the final convolutional layers are processed with the same technique proposed in the first SSD proposal (Liu, et al, 2015).

GoogLeNet (already pretrained on ImageNet) was directly implemented for training on the electron microscopy images. There intermediate outputs from middle layers that are used to prevent gradient vanishing problems. The intermediate output layers are not used for extraction. The feature extraction for SSD is conducted by using the last

convolutional layer of GoogLeNet, a concatenated convolutional output from previous inception modules, whose output is processed with a number of filters that are consistent with what SSD has suggested in its first proposal (Liu, et al., 2015). The resultant feature maps may vary in shape, but filtering processes by using convolution, the pooling are identical.

ResNet (also, already pretrained on ImageNet) followed the same process as that of GoogLeNet. The last convolutional layer of ResNet is applied to the same filtering processes with the decreasing resultant feature maps for prediction.

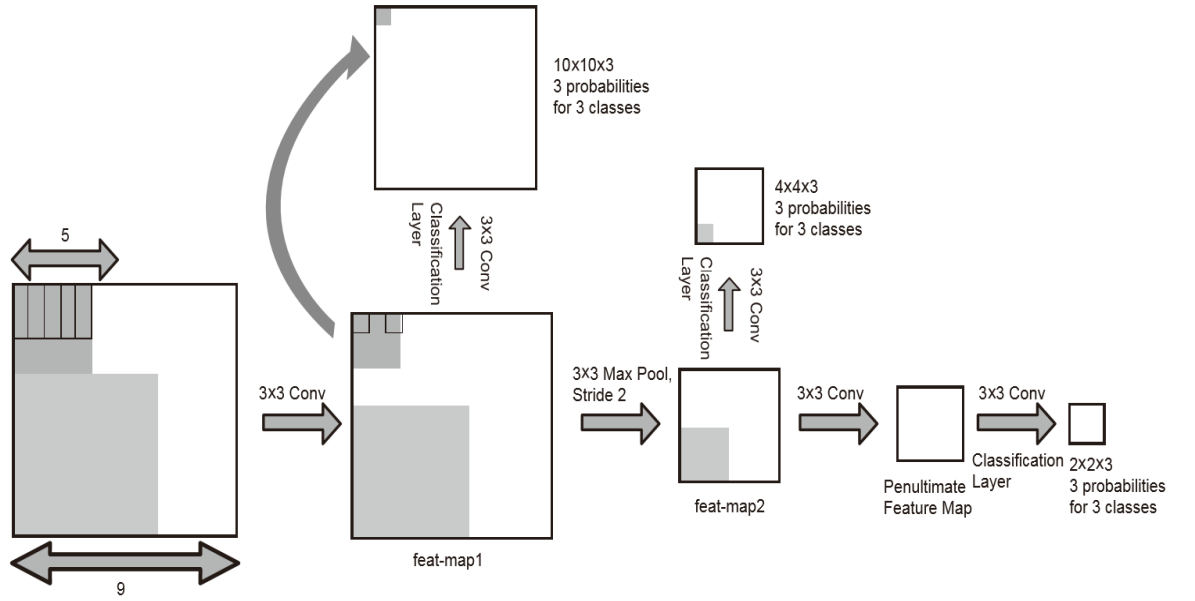


Figure 1.6: SSD convolutional layers on which predictions are made (parameters shown in this figure are indicative only)

The extracted features from each decreased feature map are processed by using two networks for bounding box regression and object classification. In order to illustrate the loss function in combination with virus prior knowledge, hereinafter, we denote $x_{i,j}^p = \{1,0\}$ that matches the i -th box to the j -th ground truth box of class p . Given this matching scheme, the sum should be $\sum_i x_{i,j}^p \geq 1$. Consequently, the loss function is

$$L(x, c, l, g) = \frac{1}{N} \left(L_{conf}(x, c) + \alpha L_{loc}(x, l, g) \right) \quad (2.14)$$

where N is the number of default boxes that correspond to ground truth boxes. For $N = 0$, loss $L = 0$ is configured, α is a weight term set to 1 ($\alpha = 1$). $L_{conf}(x, c)$ is confidence loss which refers to the softmax loss over class confidence c . Equation (2.15) states that for each positive prediction (the object being detected), there should be penalties applied to the wrong label data. There is no penalty on non-object bounding boxes.

$$L_{conf}(x, c) = -\sum_{i \in Positive}^N x_{i,j}^p \log(\hat{c}^p) - \sum_{i \in Negative} \log(\hat{c}^0) \quad (2.15)$$

$$\hat{c}^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)} \quad (2.16)$$

where $L_{loc}(x, l, g)$ is the loss for punishing the wrong localizations, the expression is given as equation (2.17)

$$L_{loc}(x, l, g) = -\sum_{i \in Positive}^N \sum_{m \in \{cx, cy, w, h\}} x_{i,j}^p smooth_{L1}(l_i^m - \hat{g}_j^m) \quad (2.17)$$

where ground truth box g and prediction box l with the anchor (cx, cy) as the centre of the default bounding box d are used for loss computation. The revised loss function reflects the punishment on objects enclosed in bounding boxes being contradictory to prior virus morphological information, the loss function is expressed as equation (2.18)

$$L_{coronaV}(x, c, l, g) = \frac{1}{N} (L_{conf}(x, c) + \alpha L_{loc}(x, l, g) + \alpha_{coronaV} L_{coronaV}(\hat{c}, c, l)). \quad (2.18)$$

In the revised $L_{coronaV}$, $\alpha_{coronaV} = 0.01$ is configured for the same reasons explained in the preceding chapters, that $L_{coronaV}$ does not excessively influence on training.

Chapter 3

Methodology

This chapter chiefly provides an elaboration of a novel loss term that reflects prediction errors against well-known virus morphological information and selected predictive models as well as the modifications to the models for compatibility, where the predictive models better reflect the information that the source images contain. Additionally, data preprocessing is justified in this chapter.

3.1 Research Design

In this thesis, original images and preprocessed images were studied as datasets. There are 6,000 electron microscopy images evenly (approximately) divided for the four categories: SARS, MESR, HIV, and COVID-19, with each one category having more than 1,500 images. The 6,000 images vary in sizes and are greyscale. All image regions have five labels: A four-tuple (r, c, w, h) representing the enclosed bounding box and label representing the class (species) of the detected object (virus particle). Each image has at least one region of interest (one bounding box containing one virus particle), only contains objects with the same class (but the predictive models referenced in this research support multi-class prediction for a given image). The data preprocessing methods used in this research work is based on electron microscopy image augmentation as well as image quality enhancement.

3.2 Dataset Preprocessing

3.2.1 Grey-Scaling Transformation

Virus images obtained by electron microscopy are in nature greyscale, multichannel images (e.g., RGB) are unlikely to hold useful information. All source images are converted into greyscale images if they are colour one, even if images are displayed in greyscale, they are often encoded in three RBG format. For an RGB image, given three channels R , G , B , the computed greyscale pixel p is

$$p = 0.2989 \cdot R + 0.5870 \cdot G + 0.1140 \cdot B \quad (3.1)$$

3.2.2 Image Resizing

The convolutional neural network requires consistent sizes of input, and all source image data was resized to $64 \times 64 \times 1$ and $128 \times 128 \times 1$, which is a typical size for computer vision tasks.

The selected algorithm for image resizing is the nearest neighbour interpolation (Roma & Sousa, 2011), also known as proximal interpolation, which is an approximation to a non-given point by simply taking the nearest point value. For the purpose of downsampling an image, image pixels are removed in proportion to image shrinkage ratio.

3.2.3 Image Brightness Adjustment

This method is to adjust the image brightness level to render images being natural to match human perception as well as easy for virus visual feature recognition. A preview of collected source images reveals that electron microscopy images are with various brightness levels. This adjustment aims CNN to only focus on morphological features instead of brightness levels when images display the same virus specimen of a class but considerably vary in terms of brightness level. This approach is only used in training not in validation or test.

In this research project, regions of interest (RoI) for each image are cropped and processed to derive pixel histograms from seeing an overall brightness representation of the image regions. If an image region is dark, this image region is enhanced in terms of brightness level by adjusting each pixel value to a normal state. An empirical study on sampled images determines a normalized image region. In general, a normalized region should have an average of pixel intensity, for any image region with large deviation from this average of pixel values. The image region is determined being either excessive bright or dark brightness adjustments are made to the whole image so that the enhanced image is of a normal brightness state. After this, we denote all regions of interest in an image as

$R_I = \{R_{I_1}, R_{I_2}, \dots, R_{I_i}, \dots, R_{I_n}\}$ for an image I of n regions with each region denoted by index i , the average pixel value p_{ave} for image I by using

$$p_{ave} = \frac{1}{n} \sum_n \left(\frac{1}{N_{R_{I_i}}} \sum_{R_{I_i}} p_j \right) \quad (3.2)$$

where j is the index of a pixel in the image region R_{I_i} . The brightness adjustment is expressed as

$$I = \begin{cases} I, & |p_{norm} - p_{ave}| < p_{deviation} \\ I + \frac{p_{norm} - p_{ave}}{3}, & otherwise \end{cases} \quad (3.3)$$

In equation (3.3), $(p_{norm} - p_{ave})/3$ is a heuristic determined by using an empirical overview. The criterion is how appealing an enhanced (added or reduced brightness level) image is to human visual perception system, as well as resembling the natural colour of a typical electron microscopy image with balanced darkness and lightness.

The brightness adjustment for data augmentation takes into account regions of interest and determines whether an image overall is too dark or bright, and applies brightness level adjustment to the whole image.

This transformation is exemplified in COVID-19 image preprocessing. Perceptually in human eyes, the displayed image is of low illumination despite being observable of virus particle contours. Given the labelled regions of interest, four rectangular regions are cropped to compute required metrics for brightness enhancement. Given the four cropped image regions, p_{ave} is computed. In addition, Table 3.1 provides medians for the regions of cropped images.

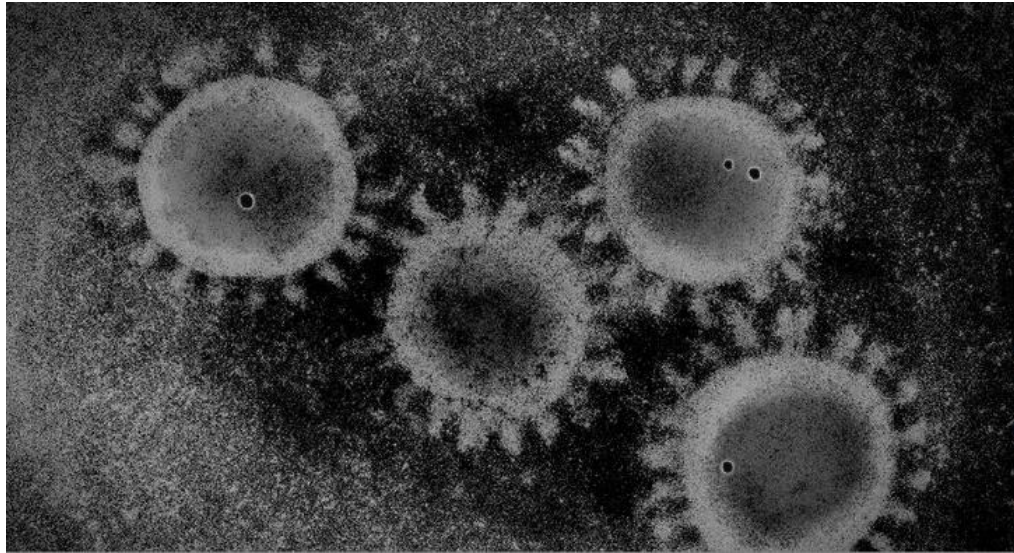
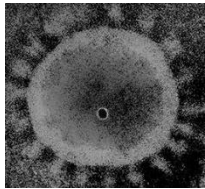
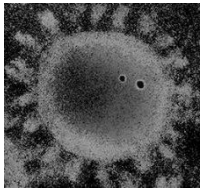
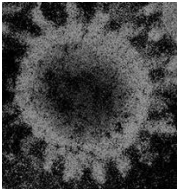
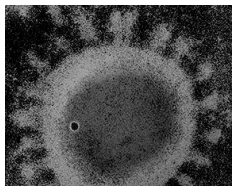


Figure 3.1: A COVID-19 image (perceptually dark)

Table 3.2: For Image brightness enhancement, four regions of interest are computed for their mean and median

Images				
Intensity Average	62.09	66.97	78.64	69.36
Median	59	65	82	69
Average of All Four Regions	73.16			
Median of All Four Regions	75			

Given the defined threshold $p_{norm} - p_{deviation}$, the mean of pixel intensities over all regions of interest is below 80, the brightness adjustment is applied to add extra pixel value. Figure 3.2 illustrates the results before and after the process (top row), revealed in the histograms before and after (bottom row) applied brightness enhancement, the postprocessed image is slightly more illuminative than the source image without perceptually losing virus morphological information.

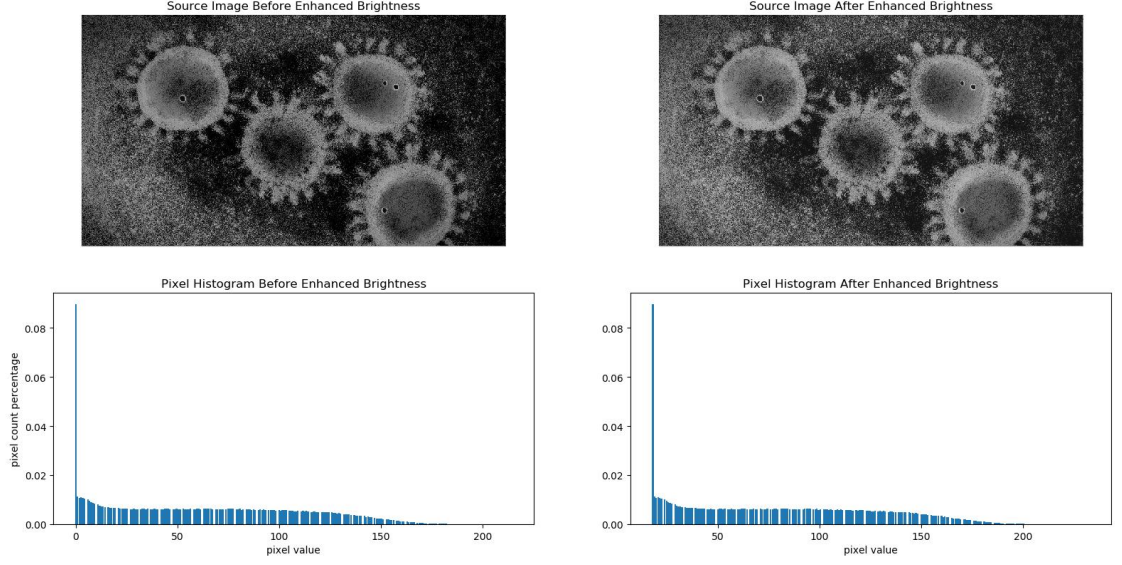


Figure 3.2: An example of a COVID-19 virus particle image before and after applied enhanced brightness level

3.2.4 Image Contrast Adjustment

The contrast adjustment allows the reduction of latent information given excessive or derisory ambient light exposure. Regions of interest in the contrast adjusted images are more evident than that in source images because different visible light spectrums have distinct contrast results. In this experiment, virus particles appeared on electron microscopy images are in greyscale, and their morphological characteristics are described in relative brightness levels. It is expected that contrast would render viral morphological characteristics more salient than un-processed.

Given a 3D image I and each pixel value $v_{x,y,z}$ corresponding to respective coordinate in the image, there exists a contrast factor $f_{contrast}$. It renders a pixel as same as the mean intensity of all pixels of an image when $f_{contrast} = 0$, and keeps the pixel value unchanged for $f_{contrast} = 1$. Pixel intensity variation increases along with growing $f_{contrast}$. The relationship between $f_{contrast}$ and input/output pixel intensities is described as

$$v_{x_{new},y_{new},z_{new}} = f_{contrast}v_{x,y,z}. \quad (3.4)$$

where $v_{\min i}$ is denoted as the minimum pixel value of an image, accordingly, $v_{\max i}$ is the maximum pixel intensity, $v_{\min o}$ and $v_{\max o}$ are the minimum and maximum pixel intensities for the processed image, the following expression states the update,

$$v_{x_{new}, y_{new}, z_{new}} = (v_{x,y,z} - v_{\min i}) \times \left(\frac{v_{\max o} - v_{\min o}}{v_{\max i} - v_{\min i}} + v_{\min o} \right). \quad (3.5)$$

A number of $f_{contrast}$ candidate values are experimented. $f_{contrast} = 1.2$ is determined in accordance to human perceptions to which degree images are after applied contrast, that derived images are inclusive of necessary visual information while being enhanced adequately to deliver granularities that are favourable to neural network training.

The result is shown in Figure 3.3, the morphological characteristics as highlighted between black and white besides the virus contours on the images. The pixel histograms before and after the applied contrast with a factor of 1.2 shows high concentration level of the absolute white pixel (255 in greyscale).

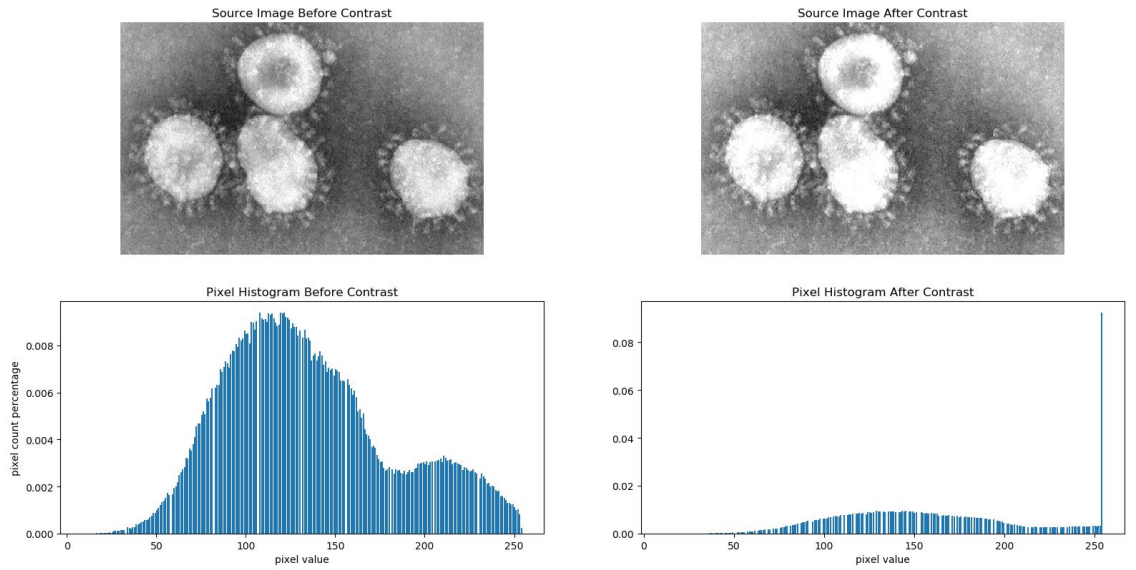


Figure 3.3: Image before and after contrast adjustment

3.2.5 Image Sharpening

Interpolation and extrapolation are used to sharpen images for expected enhanced image quality (Maurya, Mishra, Singh, & Misra, 2012). A 3×3 filter is defined as equation (3.6)

$$kernel_{smooth} = \frac{1}{13} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 5 & 1 \\ 1 & 1 & 1 \end{pmatrix}. \quad (3.6)$$

Given the size of input images, it is expected that a 3×3 filter should be capable of detecting necessary information in an image. We denote a source image I_{source} and a convolution result image I_{smooth} , there exists a relationship

$$I_{smooth} = I_{source} \cdot kernel_{smooth}, \quad (3.7)$$

where \cdot denotes a convolution multiplication operator. To sharpen an image, we define a sharpness factor $f_{sharpen}$, and denote I_{blend} as the processed image, the image derivation equation is expressed as equation (3.8).

$$I_{blend} = (1 - f_{sharpen})I_{smooth} + f_{sharpen}I_{source}. \quad (3.8)$$

where $f_{sharpen}$ controls the importance of I_{smooth} 's effects on the source image I_{source} . This technique is convenient for blurring or sharpening an image. By setting $f_{sharpen} \in (0, 1)$, it renders image I_{source} being partially blurred, and $f_{sharpen} \in (1, +\infty)$ inverses are smoothing to sharpening.

After sampled tests on different $f_{sharpen}$ s on virus electron microscopy images, $f_{sharpen} = 2.7$ is determined. Figure 3.4 demonstrates a processed image with sharpening, the effect is evident as displayed on the two histograms.

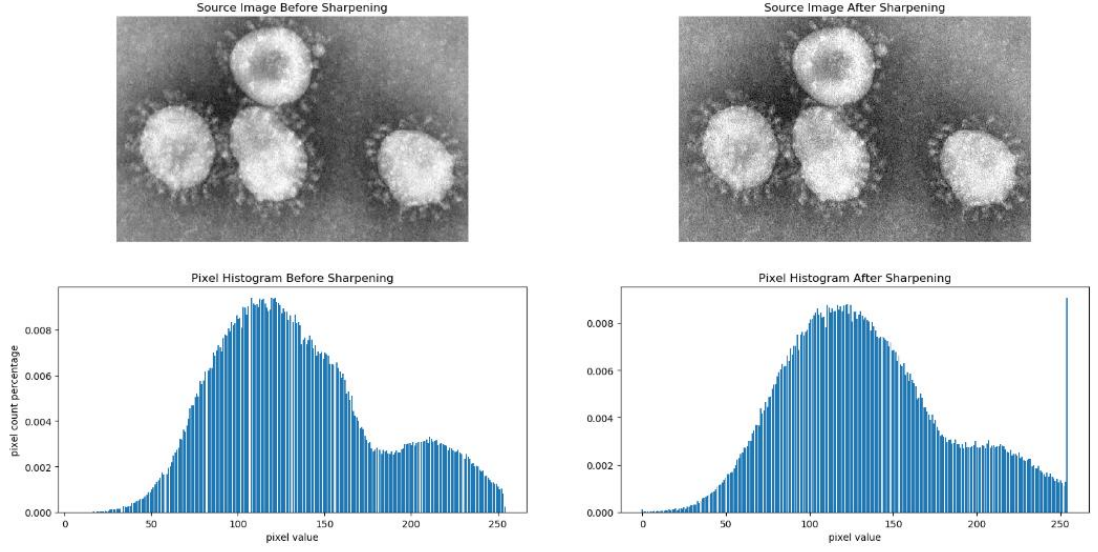


Figure 3.4: An example of a sharpened image

3.2.6 Image Rotation

Rotation introduces nonlinearity for predictive models being more robust against mis-recognised image features after rotation. All images are randomly rotated with an angle between 0° and 360° by the equation illustrated.

Hereinafter, we denote an image I , a two-dimensional matrix with corresponding coordinates (x, y) for pixel intensity v ,

$$I(x, y) = v_{x,y}. \quad (3.9)$$

We denote a rotation matrix as R given rotation angle θ ,

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}. \quad (3.10)$$

For an arbitrary rotation angle θ , there are

$$[x_{new}, y_{new}] = [x, y] \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \quad (3.11)$$

and

$$v_{x_{new},y_{new}} = v_{x,y}. \quad (3.12)$$

The new image I_{new} is denoted as

$$I_{new}(x, y, \theta) = I \left([x, y] \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \right). \quad (3.13)$$

After a rotation, unless θ is a multiple of 180° , it is expected that there are lost image regions and hollows. An image with 45° rotation renders 25% loss of information. The lost image regions are filled with pixel 0 (black), and is displayed in Figure 3.5.

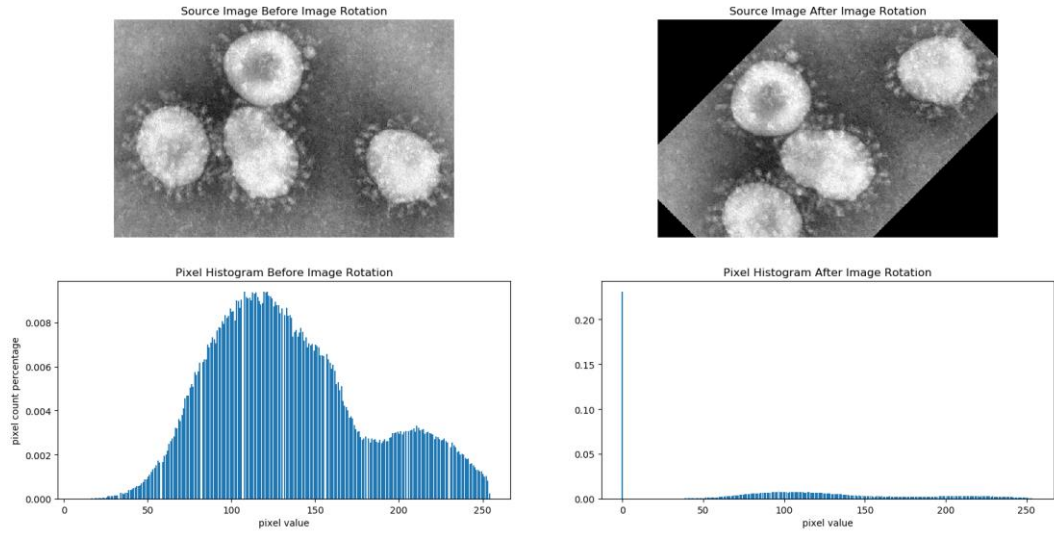


Figure 3.5: An image with a rotation angle of 45 degrees

Given this consideration, to preserve scale invariance as well as all image information, all images are rotated with an angle of 180° (flipping). In this experiment, 90% of images are flipped while 10% of rotated with an arbitrary angle between 0° and 360° .

3.2.7 Image Noise Removal

Image derivation is prone to noises for limitations such as vibration when an image is taken by the camera and device-dependent inaccuracy of how photosensors perceive light. Images are firstly converted into a CIELAB colour space (i.e., known as CIE L^*a^*b or

abbreviated as LAB colour space). CIE is an acronym for International Commission on Illumination who defines colour as a collective representation of three values: L^* for lightness ranging from black (0) to white (100), a^* for a range from green (−) to red (+) and b^* for blue (−) to yellow (+). This representation is appealing to the human perception of natural colour given spectral power distribution.

A source RGB image is device-dependent and is first transformed to a device-independent RGB format such as CIEXYZ (CIE 1931 XYZ), from which CIELAB encoding is derived. A CIEXYZ-CIELAB transformation is given as

$$\begin{aligned} L^* &= 116f\left(\frac{Y}{Y_N}\right) - 16 \\ a^* &= 500\left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right) \\ b^* &= 500\left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right) \end{aligned} \quad (3.14)$$

and

$$f(t) = \begin{cases} t^{\frac{1}{3}}, & t > \delta^3 \\ \frac{t}{3\delta^2} + \frac{4}{29}, & \text{otherwise} \end{cases} \quad (3.15)$$

Moreover,

$$\delta = \frac{6}{29}, \quad (3.16)$$

where X_n , Y_n , and Z_n are CIEXYZ tristimulus values with respect to white point, n for normalization. We define an illumination control factor h , h is used to denoise L^* and a^*b^* components given two windows: A template window and a search window to compute the weights of pixel surroundings.

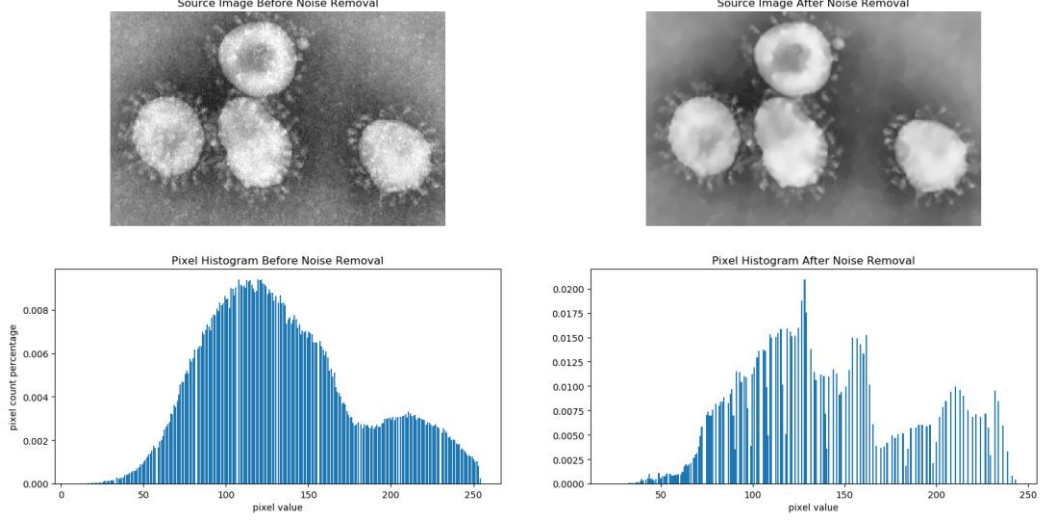


Figure 3.6: An example of noise removal

3.2.8 Image Random Region Removal

Image region random removal (Zhong, Zheng, Kang, Li, & Yang, 2017) is an image augmentation technique that addresses generalization issues by artificially introducing the absence of part of image regions. By removing regions of an image, this technique is expected to enhance the robustness of a neural network for better recognition in case of insufficient visual information. Hereinafter, we define a source image I with w_I and h_I representing its width and height then defines two integers $x_{start} \in [0, w_I]$ and $y_{start} \in [0, h_I]$ as a random start coordinate (x_{start}, y_{start}) . A removal region with a ratio r_b is defined as in proportion to image width and height, w_I and h_I .

In this experiment, $r_b = 0.15$. The two coordinates (bottom left (x_{start}, y_{start}) and top right (x_{end}, y_{end})) of a removed image region are expressed as below

$$(x_{start}, y_{start}) = (\text{random}(0, w_I), \text{random}(0, h_I)) \quad (3.17)$$

$$(x_{end}, y_{end}) = (x_{start} + r_b w_I, y_{start} + r_b h_I) \quad (3.18)$$

The random removal region selection process repeats five times, shown as in Figure 3.7.

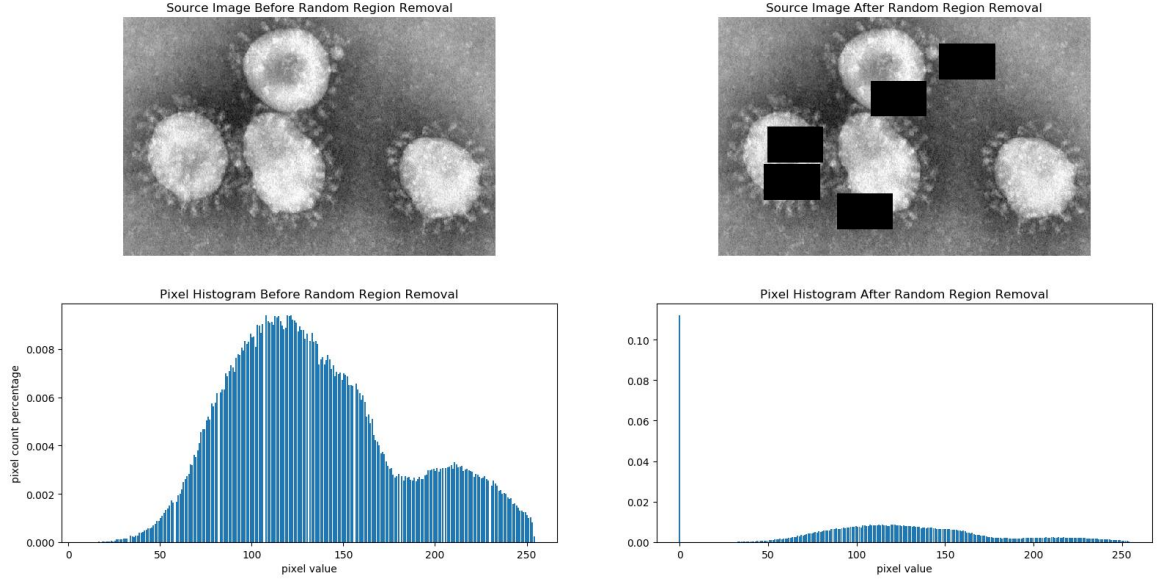


Figure 3.7: An example of random image region removal

3.3 Jaccard Index Crop

The number of source electron microscopy images is limited as the images are quite costly. The number of image samples plays an important role in generalization. If selected images within a mini-batch are similar while others are distant in resemblance, the learned errors fluctuate substantially. This phenomenon is detrimental to training and should be addressed. Thus, Jaccard overlap is selected to compute source image similarity (Kotu & Deshpande, 2019). This image similarity index is computed given equation (3.19).

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}, \quad (3.19)$$

where A and B are two given images. If A and B are both empty, $J(A, B)$ is defined as 1 ($J(A, B) = 1$). If we randomly select a large image region that satisfies a Jaccard index between $[0.75, 0.95]$, then the cropped image region is resized to the same shape as that of the raw image. The histogram illustrates that there are no significant changes in pixel distribution. The image background is removed as well as the virus regions. There are reductions in the number of surrounding protein projections.

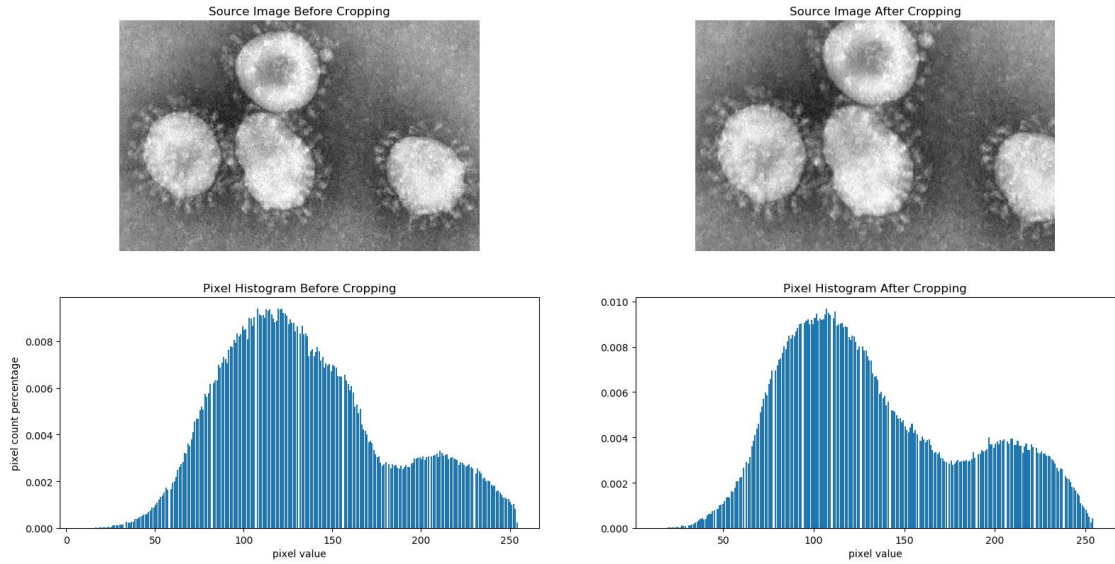


Figure 3.8: An example of the random image cropping by the defined Jaccard Index condition

3.4 An Image Preprocessing Pipeline

The processing result of the pipeline is a collection of 6000 electron microscopy images for four virus species: HIV, COVID-19, SARS, and MERS, with each species, approximately having 1500 images, the data as mentioned above preprocessing transformations are applied to the follows:

- Image denoising
- Image brightness adjustment
- Image contrast
- Image sharpening
- Image rotation (0.1 probability of occurrence)
- Image random region removal (0.1 probability of occurrence)
- Image Jaccard Index Crop (0.1 probability of occurrence)
- Image resizing

Provided the preprocessing for an input image, the first four steps are considered for data enhancement for better image quality, the source electron microscopy images are likely to be noisy, followed by three data augmentation that aims to twist the source images with added noises in the attempt to improve the robustness of a predictive model. The end of the preprocessing is to resize images, that means, different predictive models require different sizes of input.

The result of the data enhancement is shown in Figure 3.9. The most significant operation is based on image denoising. Image contrast enhancement is apparent in terms that the virus regions see increases of illumination degrees; in the pixel histogram, the number of absolute white pixel surges a significant quantity. The contours of virus regions are clear after sharpening, despite being less visually observable.

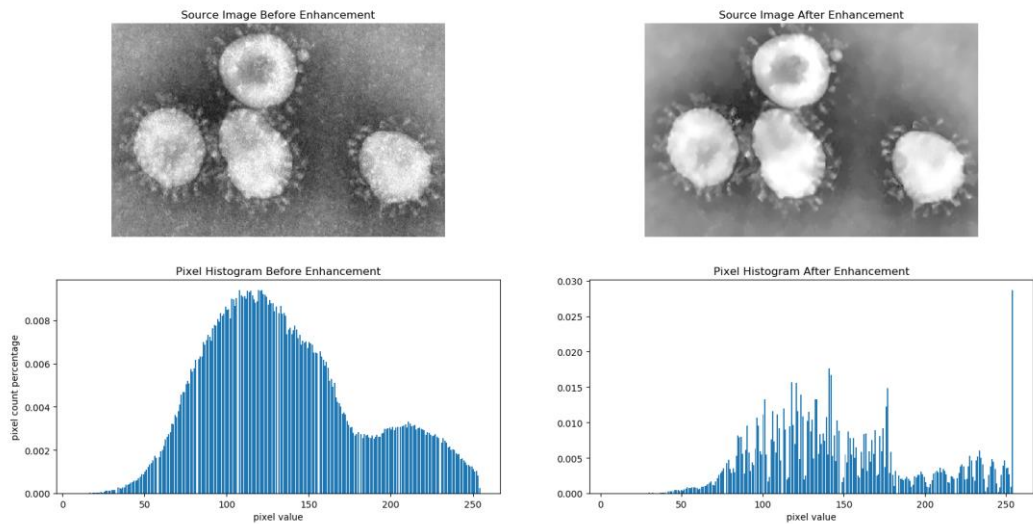


Figure 3.9: An example of four data enhancement pre-processing techniques, displayed as before and after

The data augmentation part (the 5-th, 6-th, and 7-th steps) aims to improve the predictive capability of a model by adding disturbances to the raw visual features presented in source images. Given a probability of 0.1 for the occurrence of the three events for each enhanced image, there are 23% of images going through at least one of the data augmentations, and the rest are kept unchanged.

3.5 Prior Virus Morphological Knowledge

In this experiment, we propose a novel R-CNN specifically designed for three virus species recognition and localization. It includes a group of well-known morphological attributes, e.g., spherical shape and virus regions surrounded by spike protein projections for coronavirus. In addition, it is known that foreign objects might disturb virus visual information, e.g., air bubbles. These are useful information regarding virus identification.

In this research project, we developed a mechanism that takes into account the information by adding an extra loss term to R-CNN. For example, if the predictor detects an object contained in a bounding box without observing the main object surrounded by using proportionally small spherical objects, while predicting the object as a species of coronavirus, there is an added loss to reflect this error. In this study, there are a number of known virus morphological features being identified and summarised them to a comprehensive loss term.

Regions of interest are first transformed to binary only to preserve most important morphological information, as illustrated in Figure 3.10. A number of image pre-processing techniques are applied for the transformation, first by denoising then to get binary representation by using the watershed algorithm. Given the predicted bounding box anchor expressed by a four-tuple (r, c, w, h) , the proposed regions of interest are clipped from the source image.

This research work has the object detection mechanisms based on features of the coronavirus. The first one is primitively to identify the main body of the clipped image region (the largest area by using white pixels) and the surrounding scattered white dots (spike protein projections, other white pixel counts with sizes no more than 5% of the main body area). The number of white dots is one critical metric that indicates the included object is a coronavirus region.



Figure 3.10: A coronavirus micrograph after thresholding

The finding of isolated “white dots” is a typical question of counting the number of connected components in an undirected graph, which can be solved by using either Depth First Search (DFS) or Breadth First Search (BFS) (Kaur, Sharma, & Verma, 2014). Upon discoveries, the areas $area_i$ for the i -th dot is computed. Here we define a set of white dot areas $area = \{area_1, area_2, \dots, area_n\}$, hence, $mainBody = \max(area)$. Then, we remove the white dots greater than 5% of $mainBody$ (5% is determined upon examining sampled electron microscopy images for general sizes of spike protein projections relative to main bodies), and derive $areaProj$

Algorithm 3.1: Get valid spike protein projection areas <i>areaProj</i>
Input: <i>area</i> = $\{area_1, area_2, \dots, area_n\}$
Output: <i>areaProj</i>
<i>areaProj</i> = \emptyset
<i>mainBody</i> = $\max(area)$
For <i>area_i</i> in <i>area</i>
If <i>area_i</i> < 0.05 <i>mainBody</i> Then
<i>areaProj</i> = <i>area_i</i> \cup <i>areaProj</i>

There are a number of coronavirus electron micrographs examined and determined that even for improperly processed images, e.g., failed denoised images rendering a small

number of white dots, the count c of white dots should not be less than 5.0, by which a loss term L_1 given the count c of *areaProj* is devised

$$L_1(c) = \begin{cases} 2, & c = 0 \\ \frac{1}{c}, & 0 < c < 5 \\ 0, & c \geq 5 \end{cases} \quad (3.20)$$

The second metric is geometrical location of the white dots related to the main body. All white pixel areas are treated as polygons in computation and their centroids are computed. Euclidean distances between the region centroid to the main body centroid are calculated. By comparing all distances, our conclusion is drawn.



Figure 3.11: A region of interest from a denoised coronavirus electron micrograph

The centroid, or geometric centre of a polygon, is the arithmetic mean of all points of this polygon, which holds true for n -dimensional polygon (Barton, 1972). We define a set of geometric centres $gc = \{gc_1, gc_2, \dots, gc_n\}$ for *areaProj*, given a Euclidian distance $ED(p, q)$ for n -dimensional inputs which is expressed as

$$ED(p, q) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (3.21)$$

Algorithm 3.2: Measuring Euclidian distance ***ED*** between main body and surroundings

Input: $gc = \{gc_1, gc_2, \dots, gc_n\}$

Output: Euclidian distance ***EucDis***

EucDis = \emptyset

$gc_{main} = \text{centroid}(\text{mainBody})$

For gc_i in gc

EucDis = ***EucDis.append(ED(gc_i, gc_{main}))***

Pertaining to *EucDis*, the standard deviation σ is computed as a heuristic to judge how spherical the surrounding white dots are. The greater σ is, the less likely white dots are surrounded main body by a circle. The second loss term is expressed as

$$L_2(\sigma) = k\sigma^3 \quad (3.22)$$

where we add nonlinearity with a power of 3 to amplify penalty to large σ value and k for controlling the weight of this loss term.

The third metric is the main body of sphere similarity. Given the centroid of a virus particle main body, a circle grows from the centroid point until just fully containing the main body.



Figure 3.12: A region of interest from a coronavirus electron micrograph with spherical enclosing

The third loss term is the ratio of the two areas: *mainBody* and *circleArea*, as $r = \text{mainBody}/\text{circleArea}$. Weight k and nonlinearity penalty with a power of 3 are applied

$$L_3(r) = k \left(1 - \frac{1}{r}\right)^3. \quad (3.23)$$

However, it is observable that the presented virus is unlikely of a perfect circle on an electron microscopy image, there should be no penalty on a roughly round object. Thus, upon an empirical study on sampled virus electron microscopy images, a threshold can be set to relieve penalty on normal roughly round objects, where $r = 0.3$ is the threshold, and the following expression is derived

$$L_3(r) = \begin{cases} L_3(r), & r \geq 0.3 \\ 0, & r < 0.3 \end{cases}. \quad (3.24)$$

The fourth loss term is the ratio between width denoted as w and height as h . Because virus regions are roughly spherical, they usually appear on electron microscopy images with a shape of a circle (the contour), the bounding box should be of a shape of square rather than a long rectangle. Hence, the loss term is expressed as

$$L_4(w, h) = k \left(\frac{|1 - \frac{w}{h}| + |1 - \frac{h}{w}|}{2} \right), \quad (3.25)$$

where k is a scalar that controls the importance of the penalty.

Similar to the third loss term that there barely exists a perfect spherical virus particle, and penalty on normal approximately round objects should be avoided. Given this consideration, this equation is shown as

$$L_4(w, h) = \begin{cases} L_4(w, h), & \left|1 - \frac{w}{h}\right| \geq 0.2 \\ L_4(w, h), & \left|1 - \frac{h}{w}\right| \geq 0.2 \\ 0, & \text{otherwise} \end{cases}. \quad (3.26)$$

We summarize all aforementioned loss terms with respective weights w , a coronavirus prior knowledge judgement loss term L_{coronaV} is derived as

$$L_{coronaV}(y', y, t') = w_1 L_1 + w_2 L_2 + w_3 L_3 + w_4 L_4, \quad (3.27)$$

where y' and y are predicted object class and t' is the predicted bounding box that gives information about the shape and anchor of a rectangular image region, t' is a four-tuple.

3.6 Attention Mechanism

Attention in the neural network is a broad concept referring to highlighting specific contextual information that a network should be looking at (Vaswani, et al., 2017). An attention mechanism can access previous latent states and measure them by using relevancy to the current state. The motivation behind this innovation is to input sequence representation by using the encoder-decoder structure (Sukhbaatar, szlam, Weston, & Fergus, 2015) that generates a sequence of estimated representations given input.

A typical attention unit learns three weight matrices: Query weight W_Q , key weight W_K and value weight W_V . For every input token x_i , there are $q_i = x_i W_Q$, $k_i = x_i W_K$, $v_i = x_i W_V$ for the query, key value of x_i . Attention weight a_{ij} from token x_i to token x_j is the dot product of q_i and k_j . Here we denote d_k as the number of dimensions of q_i , and $a_{ij}/\sqrt{d_k}$ is the stabilized attention weight ($\sqrt{d_k}$ can stabilize gradient during training). Softmax is applied to the output. Thus, we define $Q = \{q_1, q_2, \dots, q_n\}$, $K = \{k_1, k_2, \dots, k_n\}$, $V = \{v_1, v_2, \dots, v_n\}$, the attention layer (scaled dot product attention) is represented as

$$attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \quad (3.28)$$

In computer vision, an attention mechanism (of a 3×3 filter) is illustrated as Figure 3.13, where each cell represents a pixel. A query W_Q looks at the current pixel while keys W_K and W_V are taken into consideration of the pixel's neighbours.

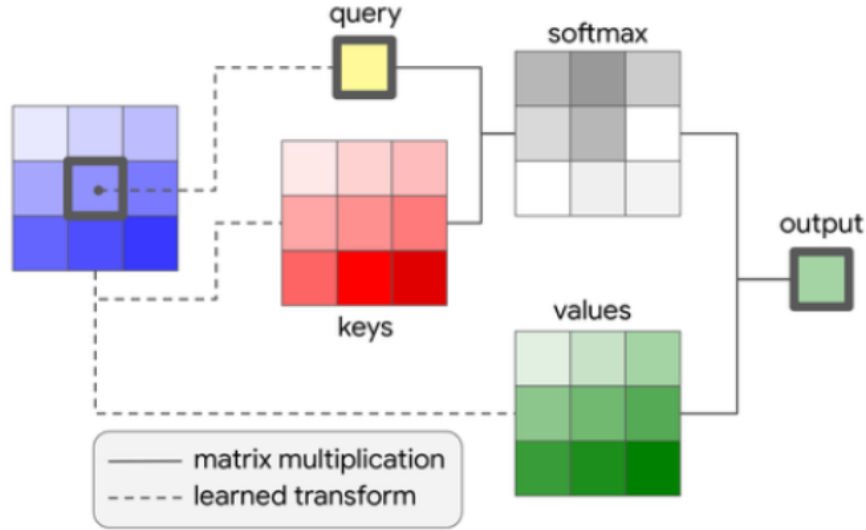


Figure 3.13: An attention mechanism in computer vision

For this experiment, it is observable that virus morphological features are largely contextual. In other words, in order to recognize a virus particle on an electron microscopy image, it is highly unlikely only to consider local visual information, e.g., a spike (or a “white dot”) can hardly indicate the existence of the virus particle to which it belongs. Upon this reflection, a large filter is used while a size of 7×7 is used, which is different from the example of a 3×3 filter. The query filter remains unchanged with a size of 1×1 . Attention mechanism aims to weigh source image pixels and mask background or foreign objects, preserve virus regions for Faster R-CNN training.

Given the example of an electron microscopy image of four coronavirus regions, in order to train an attention layer, the label for each pixel and the surroundings of the pixels is shown in Figure 3.14. Regions of interest are labelled in white while backgrounds or the irrelevant are black.

The desired output from the attention mechanism is highlights of regions of interest. In practice, the attention mechanism cannot invariably find the optimal virus regions, a hard mask should be discarded. Instead, in this experiment, the soft mask is selected to filter source images.

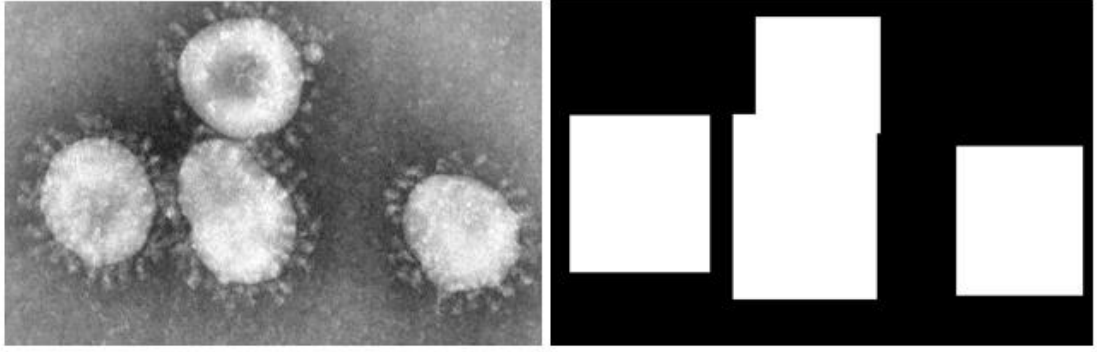


Figure 3.14: An electron microscopy image of four coronavirus regions and their masks

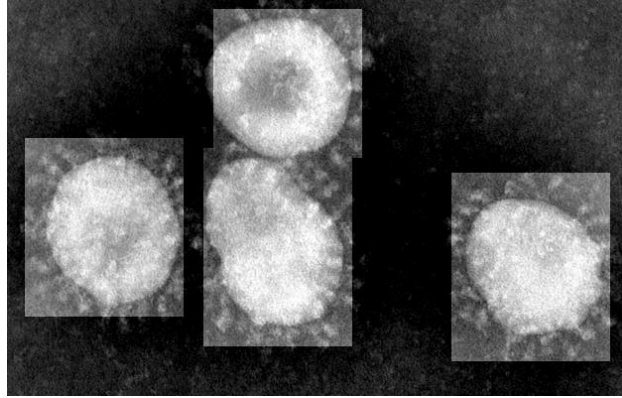


Figure 3.15: A soft masked source image by using attention mechanism (this exemplary image is presented as the ground truth instead of a real output)

The implemented attention mechanism in this research project is a simple self-attention network as described in this research. The simple self-attention network mechanism is regarded as a simple encoder-decoder network that only imports the source images, so that irrelevant areas are masked while the regions of interest are highlighted or preserved for visual feature extraction (Ramachandran, et al., 2019).

Chapter 4

Training

In this chapter, we describe the details of neural network training. Not only does this chapter expounds the proposed models and hyperparameters, but also it illustrates the modifications to the networks in practice after the training.

4.1 Training Toolboxes

In this experiment, MATLAB deep learning toolbox is selected, which has built-in R-CNN algorithms for model training. In addition, in order to exploit the latest machine learning and computer vision algorithms, PyTorch and associated toolboxes are selected for data preprocessing and model training.

4.2 Supervised Training

The network training for virus recognition belongs to a supervised learning process. There are two training models, one is for the attention network whose outputs (weighted source image pixels) are fed into Faster R-CNN for object localization and classification.

4.3 Activation Functions

Two activation functions are tested in this research. All network structures are kept unaltered with activation functions replaced with alternatives. The used activation functions for replacement are ReLU and Leaky ReLU (Nair & Hinton, 2010) (Maas, Hannun, & Ng, 2013). The expressions are listed as:

ReLU:

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (4.1)$$

Leaky ReLU:

$$f(x) = \begin{cases} 0.01, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (4.2)$$

4.4 Hyperparameters

Two hyper parameters are tested, in detail, they are:

4.4.1 Learning Rate

Learning rate is an adaptive parameter that controls updates on network parameters (Chandra & Sharma, 2016), e.g., weights, etc. Learning rate is mainly maintained by two components: Decay and momentum. Decay refers to declines in learning rate over iterations and momentum is a speed control parameter which adjusts updates in accordance with scale and direction of back propagated loss, so that when the gradient is large, it speeds up (increase learning rate η) the update; when the loss fluctuates, it slows down. A number of learning rate update schemes are adopted.

Constant

$$\eta_{n+1} = \eta_0 \quad (4.3)$$

where η_0 is the initialized learning rate. $\eta_0 = 0.005$ is set during the training.

Step-based decay

$$\eta_{n+1} = \eta_0 d^{\text{floor}(\frac{1+n}{r})}, \quad (4.4)$$

in which η_0 is the initialized learning rate, d is a decay parameter describing the percentage of decreasing, usually set at 0.5, r is value drop rate.

In this experiment, we set η_0 as 0.01 with a decaying factor d as 0.1. In addition, a relaxation term was added to avoid vanishing learning rate when n is inordinate large and η_n becomes inordinate small for training. The settling term is a simple constraint $\eta_n = \eta_{n-1}$ if $n > 30$. The equation (4.5) reflects this thresholding.

$$\eta_n = \begin{cases} \eta_0 e^{-dn}, & n \leq 30 \\ \eta_{n-1}, & n > 30 \end{cases} \quad (4.5)$$

Time-based decay

$$\eta_{n+1} = \frac{\eta_n}{1+dn}, \quad (4.6)$$

where η_n is the learning rate, d is a decay parameter given an iteration step n . At the first iteration, $\eta_0 = 0.01$ is set with $d = 0.5$. Thresholding is used to avoid vanishing learning rate when n is undue small.

$$\eta_n = \begin{cases} \frac{\eta_n}{1+dn}, & n \leq 30 \\ \eta_{n-1}, & n > 30 \end{cases} \quad (4.7)$$

Exponential

$$\eta_n = \eta_0 e^{-dn}, \quad (4.8)$$

The configuration parameters share the same semantics as above mentioned. The learning rate update in this experiment is configured with $\eta_0 = 0.01$ and $d = 0.5$. Also, thresholding is implemented to prevent the vanishing learning rate.

$$\eta_n = \begin{cases} \eta_0 e^{-dn}, & n \leq 30 \\ \eta_{n-1}, & n > 30 \end{cases} \quad (4.9)$$

4.4.2 Mini-Batch Size

Batch size plays a vital role in optimization. A batch is a small collection of source data samples, a large size permits a high level of parallelization of stochastic gradient descent and small size has the opposite effect (Peng, et al., 2018). The performance of a neural network regarding generalization is affected by it as well. The empirical study shows a strong correlation between the performance of generalization and batch size. Computation cost increases significantly when batch size increases.

A number of mini-batches sizes have been tested out of considerations including hardware limitations, optimal performance, trained model generality and computation time. It was finally set

$$miniBatchSize = 4 \quad (4.10)$$

for the training period.

4.5 Regularization

Regularization is a process to tune a predictor and prevents overfitting by adding a regularization term $R(f)$ to a loss function C for a predictive function $f(\cdot)$ given input samples $x = \{x_1, x_2, x_3, \dots, x_n\}$ with corresponding label $y = \{y_1, y_2, y_3, \dots, y_n\}$. It minimizes the following function

$$\text{Min}_f \sum_{i=1}^n C(f(x_i), y_i) + \lambda R(f) \quad (4.11)$$

where $R(f)$ is a penalty imposed on $f(\cdot)$ to punish overfitting and smooth it (Cheng, et al., 2011), λ is an important control parameter that amplifies or limits the effect of $R(f)$.

Dropout (Hinton, Srivastava, Krizhevsky, Sutskever, & Salakhutdinov, 2012) (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014) is a frequently used neural network regularization technique to address overfitting problem due to co-adaption on training data. This technique is proven effective for considerable dataset training. The core concept of this approach is randomly dropping neurons so that the network can be more robust. This process is illustrated in equation (4.12)

$$out = m * \sigma(Wx), \quad (4.12)$$

where $*$ stands for an element wise product operator and m for a binary mask vector. m follows $m_j \sim \text{Bernoulli}(p)$ where j is drawn independently, p is a manually specified probability for an output layer's result to be kept.

DropConnect (Wan, Zeiler, Zhang, Cun, & Fergus, 2013) is another regularization tool similar to dropout mechanism except that it disables connections. Equation (4.13) illustrates this operation

$$out = \sigma((M * W)x), \quad (4.13)$$

where M follows $M_{i,j} \sim \text{Bernoulli}(p)$ with i and j drawn independently given a manually specified probability p of retainment of an output layer's results.

In this experiment, a dropout probability of 0.1 is selected. Similarly, the drop Connect probability of 0.1 is selected. However, R-CNN does not use regularizations because the final prediction component is SVM. The regularizations only appear in dense layers.

4.6 Training Parameters

- The iteration number is set as 50.
- Training and validation set split ratio as 1:9, it is about 1350 images per class for training and the rests are for validation.

4.7 Attention Mechanism Training

Prior to the training, after preprocessing, image data goes through a simple attention network for masking. Since this is not the focus of this research project and the implementation of attention serves more as a data preprocessing method, in this experiment, the simple attention network training is carried out with a small learning rate 0.00001 along with an iteration number of 10.

This setup is a result of empirical performance study, in which the attention mechanism for self-encoding the regions of interest cannot guarantee the high accuracy

of region anchoring. However, it is safe to say that the attention mechanism would not introduce excessive disturbance (totally chaotic highlighted image areas for attention). Therefore, in this research project, the attention adjustment is made trivial so that there is not significant changes to the input images.

4.8 Base Networks for Feature Extraction

In this thesis, three deep CNNs are used as the base networks for visual extraction: ResNet, GoogLeNet, and VGG. The three base networks are already pretrained based on ImageNet prior to implementation in this experiment. The pretrained networks are then trained on electron microscopy images for classification, then the end layers (usually end fully connected layers) are truncated and the outputs (usually) from the last convolutional layer serve as the features for various prediction mechanism.

The preprocessed electron microscopy images are imported for training with the aid of attention mechanism to match the highlight image regions with irrelevant areas soft masks. The processed images are treated as the input data for CNNs training. Image resizing occurs to meet the input size requirements for various base networks. In this research project, a number of training rate schemes were tested and the top performers are selected. Activation functions are replaced with an alternative to see if there is any improvement in prediction accuracy, the best performed activation functions are selected for the final version of base networks later for virus detection (classification and localization).

- **R-CNN**

AlexNet is used as the base network only, AlexNet in R-CNN works on region proposals and does not require pre-training (Krizhevsky, Sutskever, & Hinton, 2012).

- **Fast R-CNN**

AlexNet is selected to be pretrained on the electron microscopy images for a typical classification task (Krizhevsky, Sutskever, & Hinton, 2012). After training, the last max pooling layer of AlexNet is replaced with an RoI pooling layer and the last softmax prediction layer is replaced with two sibling layers for bounding box regression and classification.

- Faster R-CNN

VGG16, ResNet, and GoogLeNet are offered for feature map generation. The three base networks are pretrained on the electron microscopy images for a typical classification task, the outputs of the last convolutional layers are intercepted and fed into an RPN.

- YOLO

There are 20 convolutional layers followed by an average pooling layer, one fully connected layer is constructed (denoted as YOLO base Network) for pretraining. The weights of the first 20 convolutional layers are preserved, four convolutional layers are followed by using two fully connected layers, which are added with randomly initialized weights. This approach is as same as what YOLO shows (Redmon, Divvala, Girshick, & Farhadi, 2015).

- SSD

VGG16, ResNet, and GoogLeNet are pretrained on the electron microscopy images for a typical classification. The output of the last convolutional layer of the three networks is intercepted which are applied to a number of filters (added convolutional layers). The results of all convolutions from the extra convolutional layers are fed into a detector for object classification and localization.

4.9 Training for Virus Detection (Classification and Localization)

Given the pretrained networks based on electron microscopy images, with the modifications to the output layer for bounding box regression and virus species classification, the networks, as mentioned above, are used for virus detection.

Similar to the training arrangements for the base networks (also referred as pretraining), the training for virus detection adopts the same training parameters, including learning rate schemes, activation functions, number of training epoch, and mini-batch size.

- R-CNN

The feature output by using R-CNN based network is scored by SVM for each class; the proposed bounding boxes for object localization are using selective search and applied with Non-Maximum Suppression (NMS) to reduce the number of proposed bounding boxes to a reasonable level.

- Fast R-CNN

As already described, the RoI pooling layer and two sibling prediction layers are introduced to replace the last AlexNet layers. The added layers are initialized randomly between (0,1).

- Faster R-CNN

The proposed base networks for Faster R-CNN are VGG16, ResNet, and GoogLeNet. The network parameters (mainly weights) are preserved, the network is truncated with layers served as the outputs. The outputs are fed into RPN to produce Regions of Interest (RoIs). Based on RoIs, base network features are cropped and selected for object detection

(classification and localization) by using two sibling layers. The network for classification is as same as the final dense layers of base networks, the network for bounding box regression is another network with the same structure for classification.

- SSD

The final feature layers of pretrained base CNNs are intercepted and fed into a number of convolutional layers with decreasing filter sizes for scale invariant feature extraction.

- YOLO

Similar to that of SSD, given the pretrained base network, the layers after the final convolutional layer are truncated. In addition to the implementation above, regularizations are employed. Prior to the final prediction layer, *dropout* and *dropConnect* are placed both with a probability of 0.1.

Chapter 5

Results

The dominant content of this chapter is to bring in the selected evaluation methods and list performance metrics reflecting research outcomes in the virus classification and bounding box regression from digital images.

5.1 Evaluation Methods

- **mAP (mean Average Precision)**

In this thesis, the mAP is adopted for classification evaluation for multiple classifiers. Given a precision-recall curve, by incrementing a true-false threshold, mAP computes the average precision under different thresholds. Weights can be assigned to samples of different labels, so that imbalance samples concerning all labels have defined weights (this implementation is referred to as “macro mode”). The evaluation metrics are of macro mode and do not discriminate between samples of different labels. The expression is shown as

$$mAP = \frac{1}{Q} \sum_Q AveP(q), \quad (5.1)$$

where Q is the number of queries for the average precision score.

- **Precision**

Precision is another classification evaluation tool. Precision is a general term describing information retrieval. It is calculated as the fraction of retrieved relevant records to the total queries. Hereinafter, we denote the number of total retrieved query records as $numQ$, over a query set $Q = \{q_1, q_2, \dots, q_n\}$, the number of relevant records as $numR$, over a relevant record set $R = \{r_1, r_2, \dots, r_m\}$ and the expression is displayed as

$$Precision = \frac{count(R \cap Q)}{count(Q)}. \quad (5.2)$$

In this research project, the threshold for classifying between true-or-false from a set of probability results is 0.5.

- **Recall**

Recall is used to measure the sensitivity of the probability that a relevant record can be retrieved from a set of queries. Similar to that of precision, here we define a query set

$Q = \{q_1, q_2, \dots, q_n\}$ and a relevant record set $R = \{r_1, r_2, \dots, r_m\}$, the computation is given as

$$Recall = \frac{count(R \cap Q)}{count(R)}. \quad (5.3)$$

- **F-Score**

The mostly used F-Score formula is F1 score given by equation (5.4)

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}, \quad (5.4)$$

where *Precision* and *Recall* are derived by the expressions above. F1 Score is summarised as the harmonic mean of precision and recall (Hand & Christen, 2018). The F1 score is selected along with precision and recall to evaluate classification performances of the developed classifiers in this research.

- **MSE (Mean Squared Error)**

MSE (Tuchler, Singer & Koetter, 2002), or otherwise referred to as Mean Square Deviation (MSD), measures the average of error squares. Given n predictions \hat{y} generated from one or more predictive models, corresponding to ground truth labels y , the computation is shown as

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2. \quad (5.5)$$

The employed loss metrics see the heavy implementation of MSE.

- **IOU (Intersection of Union) percentage**

In order to have fair comparisons between different methods, in this project, IOU is used to evaluate the proposed bounding boxes against the ground truth bounding boxes. A proposed bounding box compares all ground truth bounding boxes of an image, and the highest IOU percentage is selected as the IOU of the bounding box. The mean IOU

percentage of all bounding boxes produced by a regressor is regarded as the bounding box proposal benchmark for this regressor. A bounding box only takes into account the overlapping areas; thus, any classification error is not reflected. Algorithm 5.1 demonstrates this evaluation process.

Algorithm 5.1: Find mean bounding box IOU for a regressor
Input: \widehat{B}_i // proposed bounding boxes of image I_i
B_i // ground truth bounding boxes of image I_i
I // image set
Output: <i>iou_mean</i> // mean IOU of all proposed bounding boxes yielded by one regressor
<i>iou</i> = \emptyset
For I_i in I
For $\widehat{B}_{i,j}$ in \widehat{B}_i Then
<i>iou</i> _{B_{tmp}} = 0
For $B_{i,j}$ in B_i Then
<i>iou</i> _{B_{tmp}} = $\max\left(\text{IOU}(\widehat{B}_{i,j}, B_{i,j}), \text{iou}_{B_{tmp}}\right)$
<i>iou</i> = <i>iou.append(iou_{B_{tmp}})</i>
<i>iou_mean</i> = <i>average(iou)</i>

- **Absolute error for bounding box anchoring**

Another issue of evaluating bounding box anchoring is that there might exist multiple proposed bounding boxes for one ground truth bounding box, while for other ground truth, there is no matched proposed bounding boxes. To address this problem, for each image I_i , all ground truth regions of interest are summed in terms of total areas $area_{y_i}$, as well as the proposed bounding box areas $area_{\widehat{y}_i}$.

MSE is computed for each image I_i . In order to present the MSE in a reasonable manner (MSEs may be considerably large values), the result MSE is divided by the square

of the image size $(N_i \times M_i)^2$ for an image I_i with the size of $N_i \times M_i$. This consideration is summarized in eq. (5.6)

$$MSE_{bnd} = \frac{1}{n} \sum_{i=1}^n \frac{(area_{y_i} - area_{\hat{y}})^2}{(N_i \times M_i)^2}, \quad (5.6)$$

where n is the number of input images.

- **Loss and logarithmic loss**

In this project, the loss records of all predictive models are recorded to evaluate the performance of our models. In addition, the loss reflects that the prior knowledge is extracted to see the contribution of the correction by prior morphological knowledge adjustments. To illustrate the training loss changes when gradients gradually vanish for the later part of the training, the loss values are transformed in logarithm to amplify granularities of the loss trend.

5.2 Performance Metrics

Our research experiment employs a number of R-CNNs to perform the classification work. Our predictive models and corresponding mAPs for the four virus species are shown in Table 5.1. Despite pf close mAPs, the mAP of HIV virus detection is slightly higher than that of other four virus species. It should be noted that all predictive models show high mAPs up to 93%. Faster R-CNN generally performs better than its predecessors.

Table 5.1: A summary of classification results (mAP) per classifier per class

	SARS	MERS	COVID-19	HIV
R-CNN	93.32 %	93.45 %	93.21 %	94.32 %
Fast R-CNN	93.41 %	93.31 %	93.37 %	94.71 %
Faster R-CNN	94.11 %	94.63 %	94.45 %	95.43 %
SSD	94.08 %	94.58 %	94.27 %	95.18 %
YOLO	93.92 %	94.93 %	94.32 %	95.01 %

In this project, the pretrained results are recorded and displayed in mAP for each object class. All base networks display high mAPs above 91%. The results are resembled that of classifiers, that HIV virus detection is relatively higher than other base networks. The detailed data are shown in Table 5.2.

Table 5.2: A summary of classification results (mAP) for the base networks

	SARS	MERS	COVID-19	HIV
VGG11	92.94 %	92.95 %	92.38 %	93.47 %
VGG16	93.12 %	93.51 %	93.14 %	93.94 %
GoogLeNet	93.51 %	93.33 %	93.72 %	93.96 %
ResNet152	93.62 %	93.15 %	93.78 %	93.69 %
AlexNet	92.22 %	92.38 %	92.25 %	92.71 %

The means of IOUs and MSE for bounding box areas are calculated for evaluating the performances of the predictive models. SSD performs the best regarding the means of IOU, while Faster R-CNN has the lowest MSE_{bnd} . There is no large discrepancy regarding both means of IOUs and MSE of bounding boxes.

Table 5.3: The means of IOUs for different predictive models

	Means	MSE_{bnd}
R-CNN	83.54 %	0.141
Fast R-CNN	84.91 %	0.134
Faster R-CNN	84.26 %	0.133
SSD	85.38 %	0.135
YOLO	83.83 %	0.138

In this research project, the particular loss reflects the error $L_{coronav}$ against the defined prior knowledge, which is recorded per classifier per class. Given the applied small weight on $L_{coronav}$, the loss value for $L_{coronav}$ is multiplied by 100.0 for display.

Table 3.4: The loss (x100) against the prior knowledge

	SARS	MERS	COVID-19	HIV
R-CNN	0.381	0.382	0.380	0.342
Fast R-CNN	0.421	0.423	0.421	0.431
Faster R-CNN	0.410	0.409	0.410	0.473
SSD	0.395	0.397	0.394	0.352
YOLO	0.432	0.431	0.433	0.409

The total loss obtained through different learning rates is recorded shown in Table 5.5, arranged by per method per classifier. The displayed losses are verified by using the validation set.

R-CNN sees that the most significant errors regardless of the choice of training rate schemes, in contrast to that of other predictive models with noticeable improvements in terms of loss values by all kinds of learning rate schemes. Between different updating scheme of learning rates, the constant is the least favourable with greater loss values than the rest of schemes. There is no significant superiority of any particular schemes among the rest three (step-based decay, time-based decay, and exponential).

Table 5.5: The results by using different update schemes, measured by using total loss for each predictive model

	Constant	Step-based decay	Time-based decay	Exponential decay
R-CNN	0.052	0.040	0.041	0.047
Fast R-CNN	0.035	0.029	0.027	0.028
Faster R-CNN	0.037	0.030	0.029	0.031
SSD	0.036	0.028	0.029	0.031
YOLO	0.038	0.029	0.027	0.032

5.2.1 R-CNN

The metrics for measuring R-CNN classification are shown in Table 5.6, followed by the training loss results. R-CNN shows relatively low accuracy compared with that of other

predictive models. The loss convergence plot shows strong convergence but halts at 0.040, which almost doubles other predictive models.

Table 5.6: R-CNN classification performance metrics by using the validation set

	Precision	Recall	F1
SARS	0.918	0.921	0.919
HIV	0.924	0.939	0.931
COVID-19	0.917	0.921	0.918
MERS	0.919	0.913	0.915

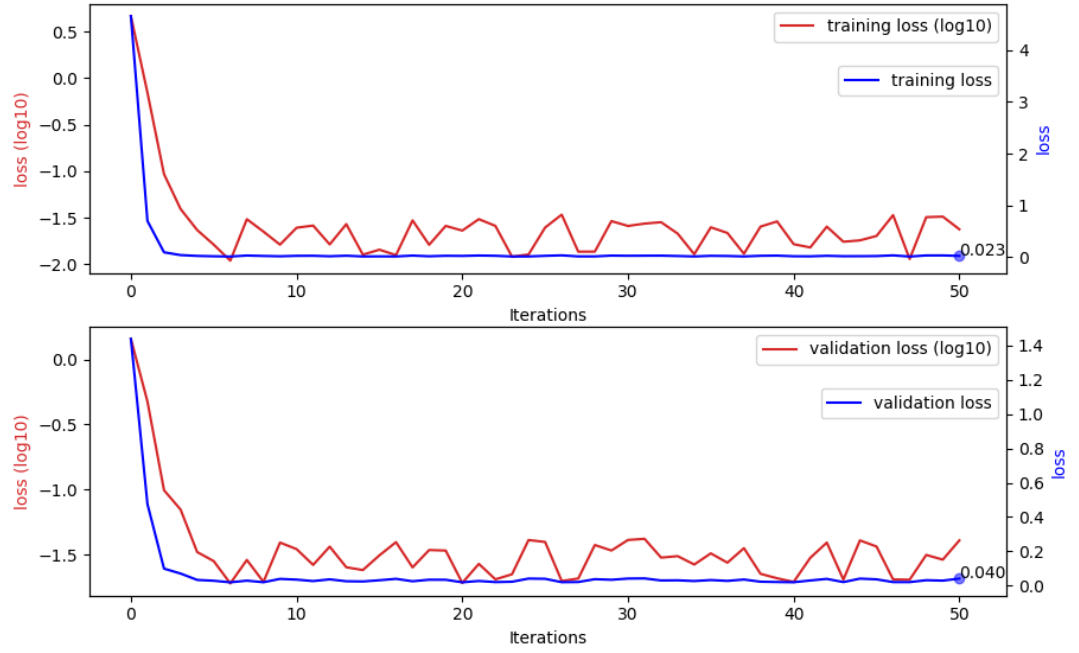


Figure 5.1: The trends of R-CNN training loss

5.2.2 Fast R-CNN

The test based on Fast R-CNN shows that, in comparison to that of faster R-CNN, there are minor differences in performance. Again, the performance metric displays similar prediction capability, for SARS and COVID-19, the results are less satisfying than that of for HIV. Despite claimed superior prediction capability in accuracy (Ren, et al., 2015), this experiment only reveals a small gap between Faster R-CNN and Fast R-CNN.

Table 5.7: Fast R-CNN performance metrics for virus detection by using the validation set

	Precision	Recall	F1
SARS	0.934	0.929	0.931
HIV	0.946	0.949	0.947
COVID-19	0.933	0.930	0.931
MERS	0.927	0.923	0.925

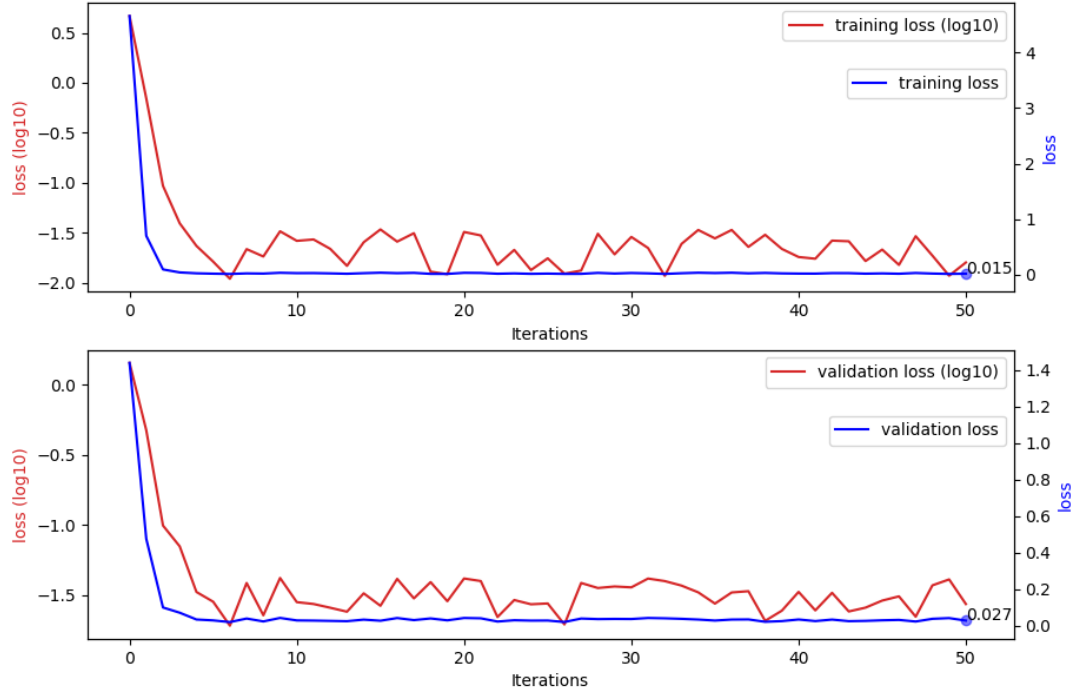


Figure 5.2: The trends of Fast R-CNN training loss

5.2.3 Faster R-CNN

Table 5.8 shows the evaluation metrics for Faster R-CNN. On average, Faster R-CNN achieved 0.95 on precision, recall, and F1. It is noticeable that the metrics of HIV virus detection are higher than those of SARS and COVID-19, as revealed in the evaluation metrics.

Table 5.8: Faster R-CNN metrics by using the validation set

	Precision	Recall	F1
SARS	0.943	0.933	0.938
HIV	0.956	0.957	0.956
COVID-19	0.932	0.934	0.953
MERS	0.935	0.932	0.933

The convergence plots of Faster R-CNN training are shown in Figure 5.3. Stochastic gradient descent shows strong convergence trends for bounding box regression and virus type classification. In order to amplify loss changes when training proceeds to a large iteration index, the plots show logarithms with a base of 10 for the loss values.

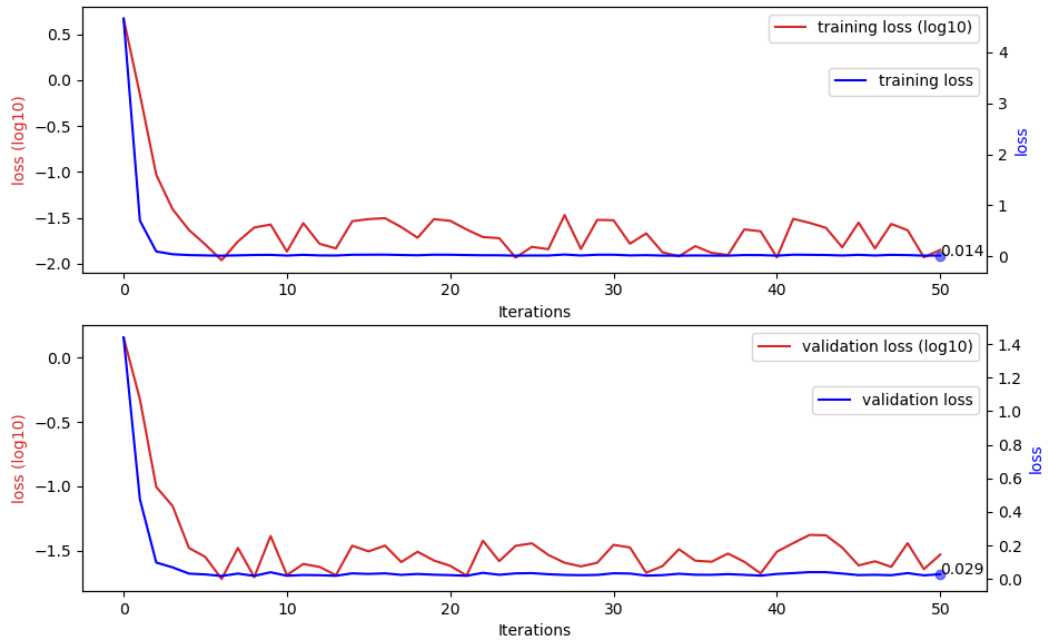


Figure 5.3: The loss trends of Faster R-CNN training

5.2.4 YOLO

The metrics for evaluating YOLO classification performance are shown in Table 5.9. In comparisons with other predictive models, there is no significant advantage. The loss changes during the training process are illustrated in the loss trends. Both the training and validation sets show strong convergence.

Table 5.9: The metrics for evaluating YOLO classification by using the validation set

	Precision	Recall	F1
SARS	0.938	0.931	0.934
HIV	0.947	0.951	0.948
COVID-19	0.933	0.938	0.935
MERS	0.930	0.931	0.931

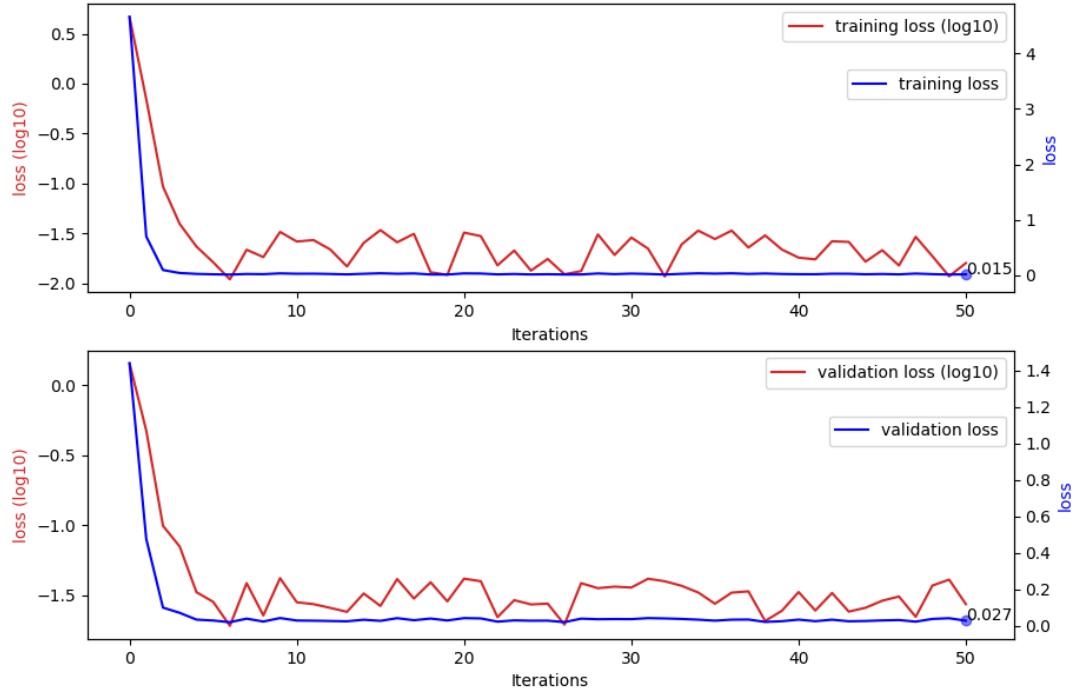


Figure 5.4: The trends of YOLO training losses

5.2.5 SSD

The experimental results based SSD classification are displayed in Table 5.10. The evaluation metrics could not show a prominent advantage over other predictive models. Among the classes, HIV virus detection has higher success rates than others. Regarding loss convergence, the loss plots of SSD training for both training and validation sets show strong convergence, which is similar to other predictive models.

Table 5.10: The metrics for evaluating SSD classification by using the validation set

	Precision	Recall	F1
SARS	0.939	0.932	0.935
HIV	0.944	0.948	0.945
COVID-19	0.931	0.940	0.935
MERS	0.935	0.929	0.932

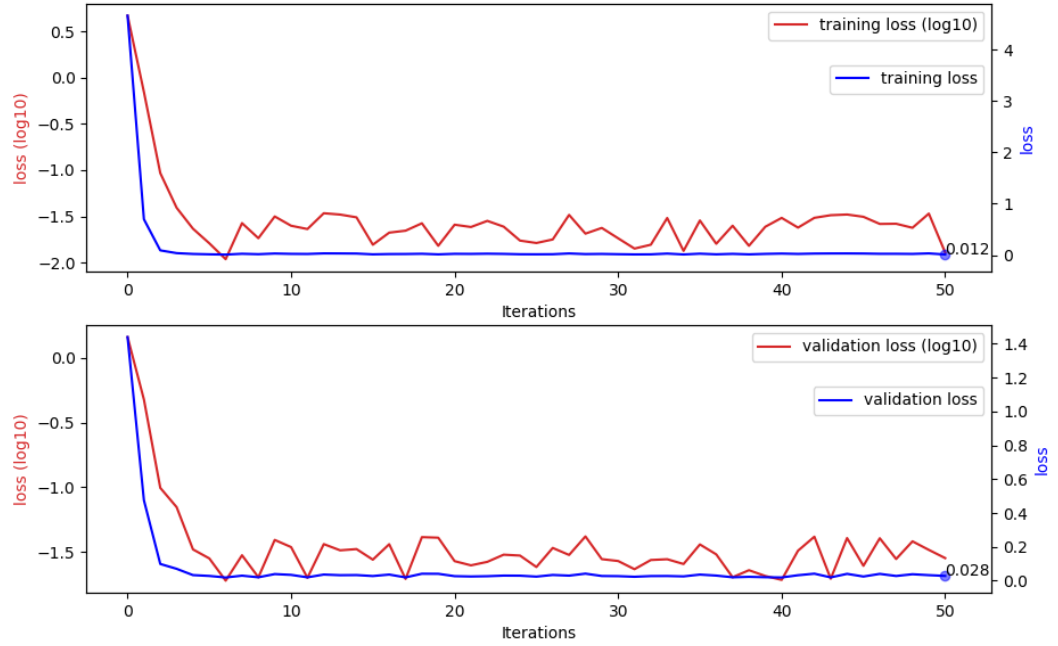


Figure 5.5: SSD training loss trends

Judging by using the aforementioned performance metrics, SSD is selected for demonstration purposes (Faster R-CNN and other detectors see great performances as well). The first example, as Figure 5.6, is a highly successful virus detection from an image, where all four virus regions are detected with right virus labels. The bounding boxes are marked as all bounding boxes fully enclose the objects without irrelevant areas.

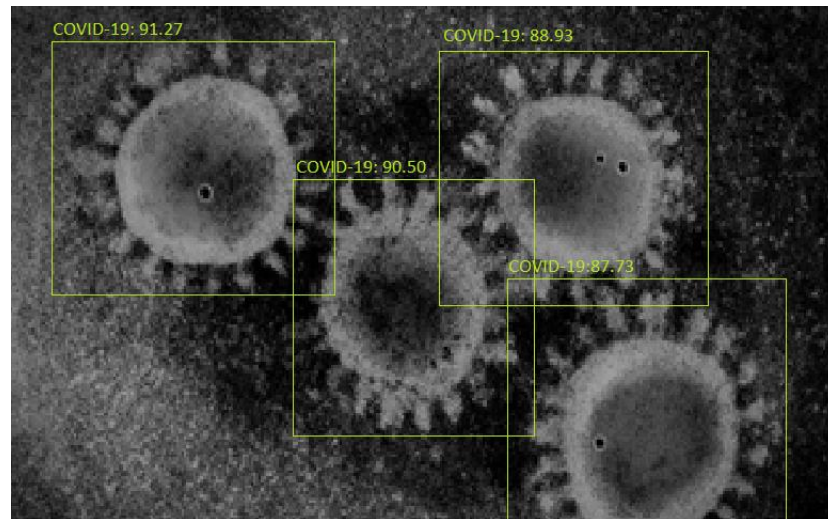


Figure 5.6: The example of virus detection 1 (COVID-19)

Another example is about a MERS image where three MERS virus regions are presented, but only two are detected. The third object is low in illumination and the morphological features are vague.

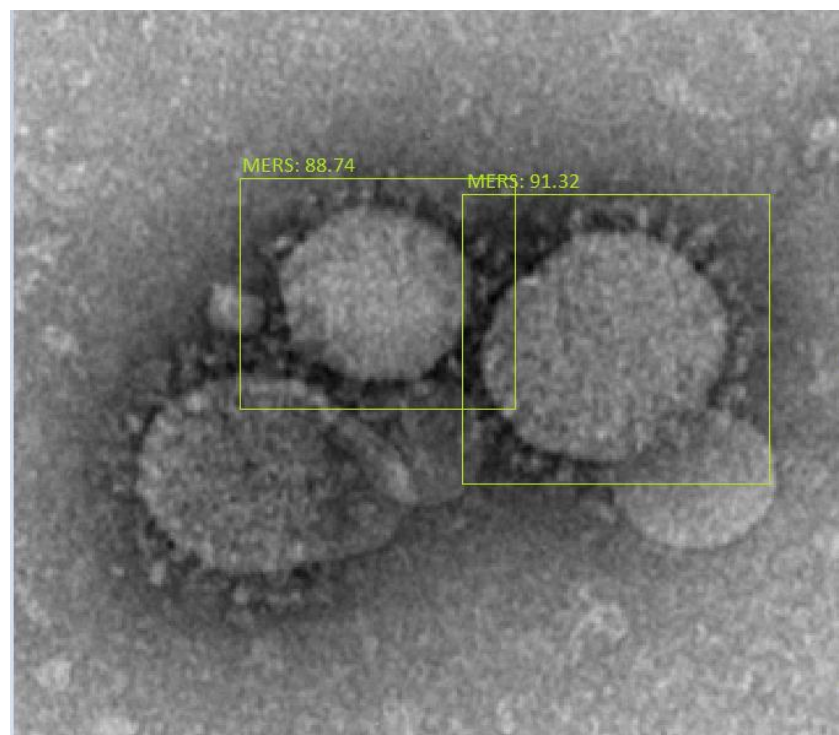


Figure 5.7: The example of virus detection 2 (MERS)

Figure 5.8 shown the ground truth bounding boxes should have been assigned with SARS labels. It is noticeable that the bounding boxes for the virus contain relatively large

irrelevant areas that lead to misrecognition by using the detector while considering more significant areas as regions of interest.

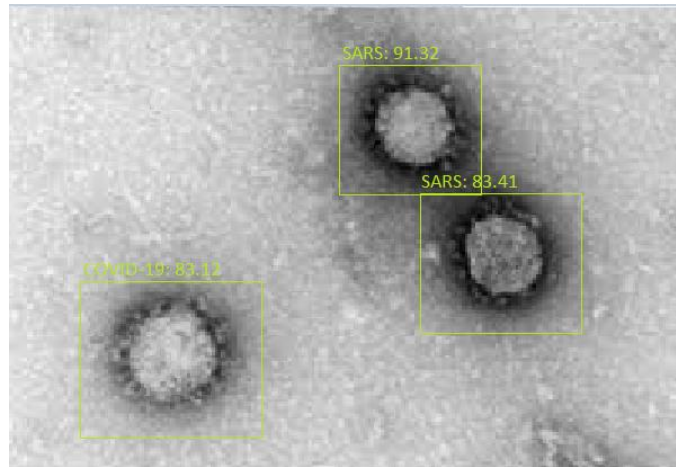


Figure 5.8: The example of virus detection 3 (SARS)

Figure 5.9 is for HIV virus detection. For HIV virus regions in this image, the confidence score is high and the proposed bounding boxes are tightly enclosed the virus.

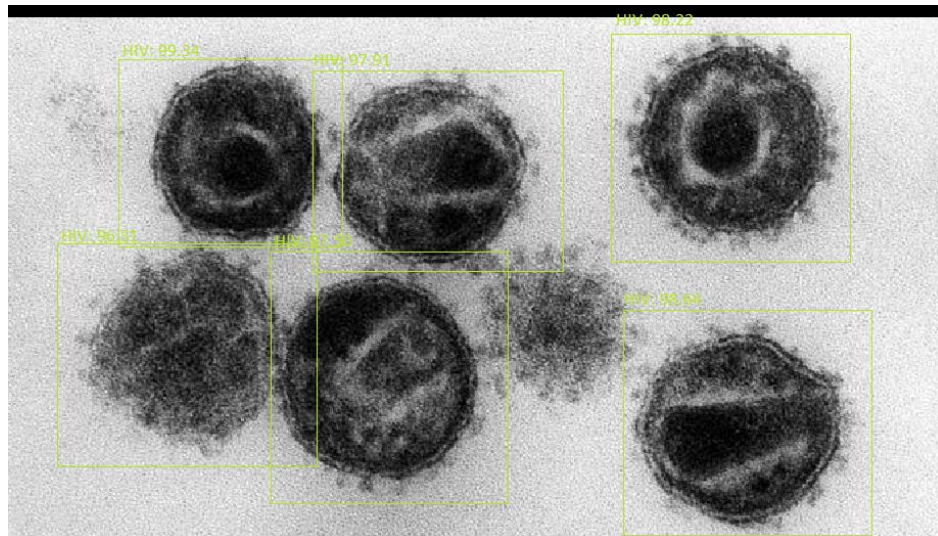


Figure 5.9: The example of virus detection 4 (HIV)

During the review of image samples, there is not suspiciously long rectangular bounding box, which is a strong indicator of the prior knowledge being successful. However, this may play a role in the failure of detecting virus regions with irregular shapes.

Chapter 6

Analysis and Discussions

In this chapter, our experimental results will be clearly analysed and discussed. The general performance of a particular predictive model, along with others, are fulfilled and explicated. It also shed light on possible reasons why virus detection was failed in the predictive models.

6.1 Analysis

The virus electron microscopy images are tested on a number of R-CNNs. From the displayed performance metrics, there are several issues.

6.1.1 Protein Projections and Morphological Features

Due to the limited quality of electron microscopy image, the majority of the source images display high levels of background noises. This reduces prediction capability significantly as spike proteins on the viral surface are likely blurred into image background. During the data pre-processing work, in this research project, we removed image noises and found that protein spikes are treated as background noises for high degrees of similarity to the background noises. A number of noise removal parameters were tested, but this problem remained unsolved.

Spike proteins are crucial morphological features for virus electron micrography classification and undistinguished spikes for resemblances to background noises. For highlighting the spike protein regarding better prediction performance, in this project, we introduced a number of data augmentation techniques. However, none of them achieved satisfactory results. Background noises tend to be amplified when we expect to enhance the image regions of virus objects, e.g., using image sharpening by highlighting edges of the viral envelope and making the morphological features obvious for recognition.

6.1.2 Morphological Feature for Each Type of Virus

SARS and COVID-19 are both appeared with spike proteins that give them the power of being highly contagious. The two types of viruses are from the same species (coronavirus), thus share a high degree of similarity in morphology. Consequently, the classification results are less satisfactory on HIV as it has fewer similarities with the other two. The strong similarities to corona visual features are found in MERS as well.

6.1.3 Attention Mechanism

Despite the rising popularity of the attention concept in particular in the area of NLP (Natural Language Processing), as well as the implementations in computer vision, in this research project, there is no evident positive contribution to the viral visual information encoding and decoding. However, it should be noted that this is not a denial of the capability of attention mechanism. The attention mechanism in this thesis is not fine-tuned, since this is not the focus of this thesis. Deep neural networks may be required to allow better visual feature interpretation so that the key matrix can be better adapted to our work.

6.2 Discussions

6.2.1 R-CNN

In this thesis, we notice the similar prediction powers of the three classifiers of the R-CNN family. In the results, R-CNN, Fast R-CNN, and Faster R-CNN have a similar performance. The proposed Faster R-CNN does not display high accuracy with a large gap than the other two. This is likely a consequence of insufficient source electron microscopy images for training. Based on Figure 5.1, Figure 5.2 and Figure 5.3, we see that they converge fast. If this experiment could collect more image data, there existed a greater probability of having better prediction capability, and the three predictors would differentiate themselves in terms of performance metrics.

The convergence plots of Faster R-CNN, Figure 5.3 show steadily declines for bounding box regression and classification loss. However, the plots converge in the first few iterations and remain halting at the 6-th iteration. By optimizing the hyperparameters such as learning rates, there exists a likelihood of improving stochastic gradient descent convergence on the two tasks (bounding box regression and classification).

6.2.2 Performance Discussions

SSD sees comparable results with that of Faster R-CNN, regarding classification and bounding box regression performance. The results produced by other predictive models do not display a considerable performance gap behind the top performers. To take into account of loss convergence trends, all predictive models converge in the first few iterations.

YOLO does not produce great prediction results as that of SSD and Faster R-CNN in general (by object classification and bounding box regression). One major advantage of YOLO is computation speed, however, it is not evaluated in this thesis. Other predictive models achieve higher prediction accuracy than YOLO at the expense of increased computation costs.

6.2.3 The Base Networks

The classification results for base networks show different performance metrics by using different networks. It is observable that the depth of a network plays an essential role in prediction accuracy, that a deep network has a higher probability of producing a great result than a shallow one. This is evident by observing the performance of AlexNet.

6.2.4 The Bounding Box

For a typical virus object with a nice and clear presence in an electron microscopy image, it is confident to conclude that the virus object can be detected in a bounding box, as this virus particle shows evident morphological characteristics. The metrics for evaluating bounding box regressors exhibit high IOUs and low absolute errors, the demonstrated exemplar images show that most detected objects are tightly confined in bounding boxes.

6.2.5 Failed Prediction Discussions

A significant number of virus objects are overlapping with each other, and they combined as an extremely complex polygon that is totally foreign to the well-known shapes of the viral envelope (typical spheres), the bounding box may give inaccurate inclusion of virus objects.

As demonstrated in the results, the virus as visual object on the images with significant vagueness has complex contours that are dissimilar with prior virus morphological knowledge. There exists a probability that data preprocessing treats virus objects as background noises, hence, it smooths the virus regions and further distorts the virus morphological information.

6.2.6 Learning Rates

In this research project, different update schemes of learning rates are tested. There is no conclusive findings for the absolute advantage of any particular schemes of learning rates. However, it is observable that the learning rate scheme is the least favourite as it introduces more errors than that of others. This is a consequence of settling at the minima where gradients are likely small, great learning rate will lead to great leaps and furthermore causes fluctuations.

6.2.7 The Contribution of Prior Knowledge

Given the prior virus morphological characteristics, the detection results reveal that no rectangular bounding boxes with high aspect ratios are proposed. The prior knowledge for loss term shows the penalty on predictions contradictive against prior knowledge. However, the imposed penalty on prior knowledge contradiction errors is not heavy.

Chapter 7

Conclusion and Future Work

In this project, five types of deep learning models for popular virus identification are justified, the research results and methods are expounded on details. In this chapter, it also integrates the conclusion into context as well as illustrates the limitations of this experiment, meanwhile we point out our future work at the end of this thesis.

7.1 Conclusion

This study first examines the morphological characteristics of the images of virus electron microscopy that shed light on automatic virus recognition by using computer vision with deep learning approaches. Upon careful examinations, MERS, SARS, and COVID-19 share more similarities than they are with HIV in terms of morphological appearance. Based on this consideration, in this research project, we propose a number of image quality enhancement and data augmentation methods aiming to improve data quality for expected prediction results.

Given the aforementioned virus morphological knowledge, in this thesis, we propose a novel loss term to reflect our predictions. The novel loss function mainly looks at surrounding protein projections appeared on electron microscopy images as “white dots”, and roughly spherical physical shape (appeared on electron microscopy images as an approximately round polygon). A bounding box shape is taken into account which is constrained from being of a high aspect ratio. However, the loss term is not heavily weighted so that a predictive model can maintain the focus on classification and bounding box regression.

In this study, we employ a total of five predictive models: R-CNN, Fast R-CNN, Faster R-CNN, SSD, and YOLO. Generally speaking, Faster R-CNN and SSD produced the best results in terms of classification and bounding box regression. However, different predictive models see different performance rankings, there is not any particular network across all performance metrics. In this project, we evaluate the training processes for different networks by measuring loss convergence, all networks see strong convergence trends.

7.2 Limitations

One limitation of this research project is data quality. The collected images in this thesis have not significant variations in terms of the way of how viruses are placed naturally. In order to put the research discovery in practical use, more real images should be taken into consideration. The presented viruses are often isolated from their hosts or where they inhabit, and foreign objects, e.g., cells and large chemical compounds, are removed. The developed predictive models are not equipped with the capability of detecting virus particles in a complex environment instead of the plain background shown as in the figures mentioned above. Without filtering out the viruses from placements adjacent to numerous foreign objects, electron microscopy images are likely inclusive of high-level noises, because the fluids on microscopy images are complex.

Another limitation is network optimization. The employed base networks in this research are directly derived from pretrained networks based on ImageNet consisting of 1,000 object labels with highly diverse image contents. The classification results of ImageNet datasets are above 90%, which is comparable with results produced by using neural networks employed in this research experiment. However, the employed networks in this research project only have four object classes. This contrast implies possible network compression where the number of network neurons can be significantly reduced, but the results remain the high accuracy. A reduction of the network size can render a faster computation.

7.3 Future Work

In order to overcome the listed limitations, our future work includes extensive considerations into viruses placements so that the predictive models can recognize viruses among complex foreign objects. The work includes a biological fluid environment, and then we will propose a virus detection mechanism that distinguishes desired virus image regions from noises. A large collection of electron microscopy images are required.

The selected networks for the objects in question are deep networks, though AlexNet is relatively short. There is no noticeable decrease in result accuracy by using AlexNet because it is short. This proves that shallow networks are capable of catching crucial virus morphological information. Our future work takes into consideration of this phenomenon and will try shallow network structures. Fine-tuning the implemented networks for virus classification and localization can improve the accuracy of our experiments, this will be attempted in the near future.

References

- Abbod, M. F. (2007). Application of artificial intelligence to the management of urological cancer. *The Journal of Urology*, 1150–1156
- Aghdam, H. H., & Heravi, E. J. (2017). *Guide to Convolutional Neural Networks*. New York, NY: Springer
- Al-Sarayreha, M, Reis, M., Yan, W., Klette, R. (2020) Potential of deep learning and snapshot hyperspectral imaging for classification of species in meat. *Food Control*
- Antonovsky, A. (1984). *The application of colour to sem imaging for increased definition* Micron and Microscopica Acta 15(2), Pages 77-84
- Barton, A. (1972). College Calculus with Analytic Geometry. Addison-Wesley
- Chan, D., Fass, D., Berger, J., & Kim, P. (1997). Core structure of GP41 from the HIV envelope glycoprotein. *Cell*, 89(2), 263–273
- Chandra, B., & Sharma, R. K. (2016). Deep learning with adaptive learning rate using laplacian score. *Expert Systems with Applications*, 63, 1-7
- Chang, K. C., H., Hou, M., Chang, F. C., Hsiao, D. C., & Huang, H. T. (2014). The SARS coronavirus nucleocapsid protein--forms and functions. *Antiviral Research*, 103, 39–50
- Chen, C., Liu, M. Y., Tuzel, O., & Xiao, J. (2016). R-CNN for small object detection. *In Asian Conference on Computer Vision (pp. 214-230)*. Springer
- Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. (2014). Semantic image segmentation with deep convolutional nets and fully connected crfs. *ArXiv preprint ArXiv:1412.7062*
- Chen, N., Zhou, M., Dong, X., Qu, J., Gong, F., Han, Y., . . . Wei, Y. (2020). Epidemiological and clinical characteristics of 99 cases of 2019 novel coronavirus pneumonia in Wuhan, China: a descriptive study. *The Lancet*, 395(10223), 507-513

- Chen, Y.-Y., Lin, Y.-H., Kung, C.-C., Chung, M.-H., & Yen, I.-H. (2019). Design and implementation of cloud analytics-assisted smart power meters considering advanced artificial intelligence as edge analytics in demand-side management for smart homes. *Sensors*, 19(9), 2047
- Cheng, F., Shen, J., Yu, Y., Li, W., Liu, G., Lee, P. W., & Tang, Y. (2011). In silico prediction of *Tetrahymena pyriformis* toxicity for diverse industrial chemicals with substructure pattern recognition and machine learning methods. *Chemosphere*, 82(11), 1636-1643
- Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. *International Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1724–1734
- Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., & Bengio, Y. (2015). Attention-based models for speech recognition. *Neural Information Processing Systems* (pp. 577-585)
- Ciresan, D., Meier, U., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2011). Flexible, high performance convolutional neural networks for image classification. *International Joint Conference on Artificial Intelligence*, 2, 1237–1242
- Comaniciu, D., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24, 603-619
- Compendium, H. S. (2008). Carla Kuiken, Thomas Leitner, Brian Foley, Beatrice Hahn, Preston Marx, Francine McCutchan, Steven Wolinsky, and Bette Korber. (2008) *Theoretical Biology and Biophysics*, Los Alamos, New Mexico. LA-UR, 08-03719
- Danilatos, G. (1986). Colour micrographs for backscattered electron signals in the SEM. *Scanning*, 8(1), 9-18
- Dawson, C. W., & Wilby, R. (1998). An artificial neural network approach to rainfall-runoff modelling. *Hydrological Sciences Journal*, 43(1), 47-66

- Denton, E. L., Zaremba, W., Bruna, J., LeCun, Y., & Fergus, R. (2014). Exploiting linear structure within convolutional networks for efficient evaluation. *International Conference on Neural Information Processing Systems* (pp. 1269-1277)
- Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2), 179–211
- Erni, R., Rossell, M., Kisielowski, C., & Dahmen, U. (2009). Atomic-resolution imaging with a sub-50-pm electron probe. *Physical Review Letters*, 102 (9), 96-101
- Fehr, R. A., & Perlman, S. (2015). An overview of their replication and pathogenesis. *Coronaviruses Methods in Molecular Biology*, 1282
- Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2009). Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9), 1627-1645
- Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient graph based image segmentation. *International Journal of Computer Vision*, 59, 167–181
- Fu, C.-Y., Liu, W., Ranga, A., Tyagi, A., & Berg, A. C. (2017). DSSD : Deconvolutional single shot detector. *ArXiv:1701.06659*
- Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4), 193–202
- Gilbert, P., McKeague, I., Eisen, G., Mullins, C., Guéye-NDiaye, A., Mboup, S., & Kanki, P. (2003). A common mechanism underlying promiscuous inhibitors from virtual and high-throughput screening. *Statistics in Medicine*, 22(4), 573–593
- Girshick, R. (2015). Fast R-CNN. *International Journal of Computer Vision*, 1440-1448.
- Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2017). Rich feature hierarchies for accurate object detection and semantic segmentation. *International Conference on Computer Vision and Pattern Recognition*, 1-21

- Goldsmith, C. S., Tatti, K. M., Ksiazek, T. G., Rollin, P. E., Comer, J. A., Lee, W. W., . . . Zaki, S. R. (2004). Ultrastructural characterization of SARS coronavirus. *Emerging Infectious Diseases*, 10(2), 320
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Machine learning basics. *Deep learning*, 98-164
- Graves, A., Liwicki, M., Fernandez, S., Bertolami, R., Bunke, H., & Schmidhuber, J. (2009). A novel connectionist system for improved unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5), 855–868
- Groot, R. J., Baker, S. C., Baric, R. S., Brown, C. S., Drosten, C., Enjuanes, L., . . . S, E. J. (2013). Middle East respiratory syndrome coronavirus (MERS-CoV): Announcement of the coronavirus study group. *Journal of Virology*, 87(14), 7790–7792
- Gua, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B. S., . . . Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, 77, 354-377
- Guimarães, P., & McGreavy, C. (1995). Flow of information through an artificial neural network. *Computers & Chemical Engineering*, 19, 741-746
- Hand, D., & Christen, P. (2018). A note on using the F-measure for evaluating record linkage algorithms. *Statistics and Computing*, 28(3), 539-547
- Haralick, R., Dinstein, & Shanmugam, K. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, pp. 610–621
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9), 1904-1916

- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770-778)
- He, P., Huang, W., He, T., Zhu, Q., Qiao, Y., & Li, X. (2017). Single shot text detector with regional attention. In *IEEE International Conference on Computer Vision* (pp. 3047-3055)
- Hebb, D. (1949). *Organization of behavior*. *J. Clin. Psychol*, 6(3), 335-307
- Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *arXiv:1207.0580*
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780
- Holschneider, M., Kronland-Martinet, R., Morlet, J., & T. P. (1990). A real-time algorithm for signal analysis with the help of the wavelet transform. In *Wavelets* (pp. 286-297)
- Hortolà, P. (2010). Using digital colour to increase the realistic appearance of SEM micrographs of bloodstains. *Micron*, 41(7), 904-908
- Hosang, J., Benenson, R., & Schiele, B. (2017). Learning non-maximum suppression. *International Conference on Computer Vision and Pattern Recognition*, 4507-4515
- Hubel, D. H., & Wiesel, T. N. (1968). Receptive fields and functional architecture of monkey striate cortex. *The Journal of Physiology*, 195(1), 215–243
- Ishtiaq, U., Abdul Kareem, S., Abdullah, E. R. M. F., Mujtaba, G., Jahangir, R., & Ghafoor, H. Y. (2020). Diabetic retinopathy detection through artificial intelligent techniques: A review and open issues. *Multimedia Tools & Applications*, 79(21/22), 15209–15252
- Ji, H., Liu, Z., Yan, W., Klette, R. (2019) Early diagnosis of Alzheimer's disease based on selective kernel network with spatial attention. *ACPR* (pp. 503-515)

- Ji, H., Yan, W., Klette, R. (2019) Early diagnosis of Alzheimer's disease using deep learning. ICCCV
- Johnson, H. C., Gossner, C. M., Colzani, E., Kinsman, J., Alexakis, L., Beauté, J., . . . Ekdahl, K. (2020). Potential scenarios for the progression of a COVID-19 epidemic in the European Union and the European Economic Area, *Eurosurveillance*, 25(9), 2000202
- Jordan, M. I. (1997). Serial order: A parallel distributed processing approach. *Advances in Psychology*, 121, 471–495
- Kaur, M. A., Sharma, M. P., & Verma, M. A. (2014). A appraisal paper on Breadth-first search, Depth-first search and Red black tree. *International Journal of Scientific and Research Publications*, 4(3), 2-4
- Kleene, S. (1956). Representation of events in nerve nets and finite automata. *Annals of Mathematics Studies*, 3–41
- Klein, J. S., Bjorkman, P. J., & Rall, G. F. (2010). Few and far between: How HIV may be evading antibody avidity. *PLOS Pathogens*, 6(5)
- Kotu, V., & Deshpande, B. (2019). *Data Science Concepts and Practice (2nd Edition)*. Morgan Kaufmann
- Krizhevsky, A., Sutskever, I., & Hinton, G. (2012). ImageNet classification with deep convolutional neural networks. *International Conference on Neural Information Processing Systems* (pp. 1097-1105)
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6), 84–90
- Kylberg, G. U., & Sintorn, I. (2011). Virus texture analysis using local binary patterns and radial density profiles. *Iberoamerican Congress on Pattern Recognition*, 573-580

- Lin, T. Y., RoyChowdhury, A., & Maji, S. (2015) Bilinear CNNs for fine-grained visual recognition. *ArXiv:1504.07889*
- Liu, C., Sharan, L., Adelson, E. H., & Rosenholtz, R. (2010). Exploring features in a bayesian framework for material recognition. *International Computer Vision and Pattern Recognition* (pp. 239-246)
- Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2015). SSD: Single shot multibox detector. *IEEE ICCV* (pp. 21-37)
- Lu, J., Nguyen, M., Yan, W. (2018) Pedestrian detection using deep learning. *IEEE AVSS*
- Liu, Z., Yan, W., Yang, B. Image denoising based on a CNN model. *IEEE ICCAR*. (pp., 389-393)
- Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 3431-3440)
- Lyumkis, D., Julien, J.-P., de Val, N., Cupo, A., Potter, C. S., Klasse, P.-J., . . . Moore, J. P. (2003). Cryo-EM structure of a fully glycosylated soluble cleaved HIV-1 envelope trimer. *Science*, 342(6165), 1484–1490
- Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. *International Conference on International Conference on Machine Learning*, 30(1)
- Marmanis, D., Datcu, M., Esch, T., & Stilla, U. (2015). Deep learning earth observation classification using ImageNet pretrained networks. *IEEE Geoscience and Remote Sensing Letters*, 13(1), 105-109
- Masters, P. S. (2019). Coronavirus genomic RNA packaging. *Virology*, 537, 198-207
- Matthew, D., & Fergus, R. (2014). Visualizing and understanding convolutional neural networks. *In European Conference Computer Vision and Pattern Recognition, Zurich, Switzerland* (pp. 6-12)

- Matusugu, M., Mori, K., Mitari, Y., & Kaneda, Y. (2003). Subject independent facial expression recognition with robust face detection using a convolutional neural network. *Neural Networks*, 16(5), 555–559
- Maurya, S. K., Mishra, P. K., Singh, R. K., & Misra, A. K. (2012). Image enhancement by intensity based interpolation and selective threshold. In *International Conference on Communication Systems and Network Technologies* (pp. 174-178)
- McCulloch, W., & Walter, P. (1943). A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 115–133
- McGovern, S. L., Caselli, E., Grigorieff, N., & Shoichet, B. K. (2002). A common mechanism underlying promiscuous inhibitors from virtual and high-throughput screening. *Journal of Medicinal Chemistry*, 45(8), 1712-1722
- McIntosh, K., Dees, J., & Becker, W. (1967). Recovery in tracheal organ cultures of novel viruses from patients with respiratory disease. *Proceedings of the National Academy of Sciences of the United States of America*, 57(933)
- Michael, M., Cavanagh, C., & Lai, D. (1997). The molecular biology of coronaviruses. *Advances in Virus Research*, 48, 1-100
- Miljanovic, M. (2012). Comparative analysis of recurrent and finite impulse response neural networks in time series prediction. *Indian Journal of Computer and Engineering*, 3(1)
- Miller, D. D., & Brown, E. W. (2018). Artificial intelligence in medical practice: The question to the answer? *The American Journal of Medicine*, 131(2), 129-133
- Mittal, S. (2020). A survey of FPGA-based accelerators for convolutional neural networks. *Neural Computing and Applications*, 32, 1109–1139
- Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. *International Conference on International Conference on Machine Learning*, 807–814

- Nanni, L., Paci, M., Brahnam, S., Ghidoni, S., & Menegatti, E. (2013). Virus image classification using different texture descriptors. *International Conference on Bioinformatics and Computational Biology*, 56–61
- Neuman, B. W., Adair, B. D., Yoshioka, C., Quispe, J. D., Orca, G., Kuhn, P., . . . Buchmeier, M. J. (2006). Supramolecular architecture of severe acute respiratory syndrome coronavirus revealed by electron cryomicroscopy. *Journal of Virology*, 80(16), 7918-7928
- Nie, D., Gao, Y., Wang, L., & Shen, D. (2018). ASDNet: Attention based semi-supervised deep networks for medical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (pp. 370-378)
- Oho, E., Ichise, N., Martin, W. H., & Peters, K.-R. (1995). Practical method for noise removal in scanning electron microscopy. *Scanning*, 18, 50-54
- Ojala, T., Pietikäinen, M., & Mäenpää, T. (2002). Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7), 971–987
- Peng, C., Xiao, T., Li, Z., Jiang, Y., Zhang, X., Jia, K., ... & Sun, J. (2018). MegDet: A large mini-batch object detector. *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 6181-6189)
- Rabaan, A., Al-Ahmed, S., Haque, S., Sah, R., Tiwari, R., Malik, Y., . . . Rodriguez-Morales, A. (2020). SARS-CoV-2, SARS-CoV, and MERS-CoV: A comparative overview. *Le infezioni in medicina: Rivista Periodica Di Eziologia, Epidemiologia, Diagnostica, Clinica e Terapia Delle Patologie Infettive*, 28(2), 174-184
- Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., & Shlens, J. (2019). Stand-alone self-attention in vision models. *International Conference on Neural Information Processing Systems* (pp. 68-80)
- Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2015). You only look once: Unified, real-time object detection. In *ICCV* (pp. 779-788)

- Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *International Conference on Neural Information Processing Systems*, (pp. 91-99)
- Roma, N., & Sousa, L. (2011). A tutorial overview on the properties of the discrete cosine transform for encoded image and video processing. *Signal Processing*, 91(11), 2443-2464
- Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 386–408
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., . . . Li, F.-F. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3), 211–252
- Saey, T. H. (2013). Story one: Scientists race to understand deadly new virus: SARS-like infection causes severe illness, but may not spread quickly among people. *Science News*, 183(6), 5–6
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 85–117
- Samek, W., Wiegand, T., & Müller, K. R. (2017). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. *ArXiv preprint ArXiv:1708.08296*
- Shakri, A. A., Saidi, S. A., Jaafar, H., Mansor, M. N., Mustafa, W. A., & Junoh, A. K. (2017). Entropy virus microscopy images recognition via neural network classifiers. *IEEE International Conference on Control System, Computing and Engineering*, 348–351
- Shen, D., Wu, G., & Suk, H.-I. (2017). Deep learning in medical image analysis. *Annual Review of Biomedical Engineering*, 19, 221-248
- Shen, D., Xin, C., Nguyen, M., Yan, W. (2018). Flame detection using deep learning
IEEE ICCAR

- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *ArXiv:1409.1556*
- Srivastava, N., Hinton, G. E., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56), 1929–1958
- Sukhbaatar, S., szlam, a., Weston, J., & Fergus, R. (2015). End-to-end memory network. *Advances in Neural Information Processing Systems*, 2440–2448
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 1-9)
- Tang, X., Du, D. K., He, Z., & Liu, J. (2018). PyramidBox: A context-assisted single ShotFace detector. *ECCV*, 1-17
- Tuceryan, M., & Jain, A. K. (1998). The Handbook of Pattern Recognition and Computer Vision. *World Scientific Publishing Co.*
- Tuchler, M., Singer, A. C., & Koetter, R. (2002). Minimum mean squared error equalization using a priori information. *IEEE Transactions on Signal processing*, 50(3), 673-683
- Uijlings, J., Sande, K. v., Gevers, T., & Smeulders, A. (2013). Selective search for object recognition. *International Journal of Computer Vision*, 104(2), 154-171
- Valueva, M., Nagornov, N., Lyakhov, P., Valuev, G., & Chervyakov, N. (2020). Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. *Mathematics and Computers in Simulation*, 177, 232–243
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *In Advances in neural information processing systems* (pp. 5998-6008)

- Wan, L., Zeiler, M., Zhang, S., Cun, Y. L., & Fergus, R. (2013). Regularization of neural networks using DropConnect. *In International Conference on Machine Learning*, 28(3), pp. 1058-1066
- Wang, S. H., Phillips, P., Sui, Y., Liu, B., Yang, M., & Cheng, H. (2018). Classification of Alzheimer's disease based on eight-layer convolutional neural network with leaky rectified linear unit and max pooling. *Journal of medical systems*, 42(5), 85
- Wang, T., Anwer, R. M., Cholakkal, H., Khan, F. S., Pang, Y., & Shao, L. (2019). Learning rich features at high-speed for single-shot object detection. *ICCV*, 1971-1980
- Wang, X., Yang, M., Zhu, S., & Lin, Y. (2013). Regionlets for generic object detection. *International Conference on Computer Vision* (pp. 17-24)
- Williams, R. J., Hinton, G. E., & Rumelhart, D. E. (1986). Learning representations by back-propagating errors. *Nature*, 323 (6088), 533–536
- Willis, M. J., Montague, G. A., Di Massimo, C., Tham, M. T., & Morris, A. J. (1992). Artificial neural networks in process estimation and control. *Automatica*, 28(6), 1181-1187
- Wong, C., Li, X., K., Lau, S., C., & Woo, P. (2019). Global epidemiology of bat coronaviruses. *Viruses*, 11(2), 174
- Wrapp, D., Wang, N., Corbett, K. S., Goldsmith, J. A., Hsieh, C.-L., Abiona, O., . . . McLellan, J. S. (2020). Cryo-EM structure of the 2019-nCoV spike in the prefusion conformation. *Science*, 367(6483), 1260-1263
- Wu, C. M., Chen, Y. C., & Hsieh, K. S. (1992). Texture features for classification of ultrasonic liver images. *IEEE Transactions on Medical Imaging*, 11(2), 141–152
- Xin, C., Nguyen, M., Yan, W. (2020). Multiple flames recognition using deep learning. *In Handbook of Research on Multimedia Cyber Security*, IGI Global (pp. 296-307)
- Xue, J., Li, J., & Gong, Y. (2013). Restructuring of deep neuralnetwork acoustic models with singular value decomposition. *Interspeech* (pp. 2365-2369)

- Yan, W. (2019) Introduction to Intelligent Surveillance. Springer.
- Zell, A., Mache, N., Hübner, R., Mamier, G., Vogt, M., Schmalzl, M., & Herrmann, K. U. (1994). SNNS: Stuttgart Neural Network Simulator. *In Neural Network Simulation Environments* (pp. 165-186)
- Zhang, L., Lin, D., Sun, X., Curth, U., Drosten, C., Sauerhering, L., . . . Hilgenfeld, R. (2020). Crystal structure of SARS-CoV-2 main protease provides a basis for design of improved α -ketoamide inhibitors. *Science*, 368(6489), 409-412
- Zhang, P., Xue, J., Lan, C., Zeng, W., Gao, Z., & Zheng, N. (2018). Adding attentiveness to the neurons in recurrent neural networks. *In European Conference on Computer Vision (ECCV)* (pp. 135-151)
- Zhang, Q., W Yan, W. (2018) Currency recognition using deep learning. *IEEE AVSS*.
- Zhang, S., Wen, L., Bian, X., Lei, Z., & Li, S. Z. (2017). Single-shot refinement neural network for object detection. *IEEE Conference on Computer Vision and Pattern Recognition* (pp. 4203-4212)
- Zhong, Z., Zheng, L., Kang, G., Li, S., & Yang, Y. (2017). Random erasing data augmentation. *IEEE International Conference on Communication Technology (ICCT)*, 1699–1703