題目：使用不同的深度學習模型來檢測 COVID-19

Title：Using Different Deep Learning Models to Detect COVID-19

姓　名 Name	：	許天浩
學　號 Student No.	：	1709853J-I011-0223
學　院 Faculty	：	資訊科技學院
課　程 Program	：	軟件技術及其應用
專　業 Major	：	理學碩士
指導老師 Supervisor	：	Dr. Subrota Kumar Mondal
日　期 Date	：	30/04/2021

Using Different Deep Learning Models to Detect COVID-19

by

Tianhao Xu
(Student ID: 1709853J-I011-0223)

Supervisor: Dr. Subrota Kumar Mondal

A thesis
submitted to the Faculty of Information Technology
and the School of Graduate Studies of
Macau University of Science and Technology
in partial fulfillment of the requirements for the degree of
(Bachelor of Science)
in
(Software Technology And Its Application)

(30/04/2021)

摘要

背景:

截至4月21日，根據實時疫情數據報告，中國共診斷出103376例新冠疫情病例，累計死亡人數達4856人。而在國外，截至目前共診斷出144183089例，累計死亡人數達3054198人。全球疫情已經是一個不爭的事實，值得全世界人民深思。在這個沒有火藥的戰場上，隱蔽的戰爭遠比赤裸裸的戰鬥更具殺傷力。新型冠狀動脈肺炎COVID-19偽裝得非常好，前線戰場上的醫生很難第一時間 "看清 "它。

目的:

本研究旨在利用深度學習技術建立早期篩查模型，將COVID-19肺炎與健康病例的肺部CT圖像區分開來，從人工智能角度幫助醫生進行快速診斷，提高診斷效率。

材料和方法:

公開的COVID-19CT圖像數據集。數據主要通過影印本中CT圖以及其他公共來源收集，並通過醫院和醫生間接收集。
使用了多種深度學習模型去對新冠肺炎的CT圖進行圖像分類。

結果:

在多個深度學習模型對於圖像分類的對比實驗中，DResUnet的實驗結果為85.54 Accuracy(%)，87.02 AUC(%)，分類效果最佳。

結論:

研究表明，開發的基於人工智能的圖像分析可以在檢測冠狀病毒實現比較高的精確度。希望能夠在醫院得到廣泛地使用，提升醫生的診斷效率與準確度，同時幫助醫生在其圖像提取的特征中，更好的對臨床癥狀進行總結。

關鍵詞：COVID-19，新型冠狀肺炎病毒，深度學習，卷積神經網絡

Abstract

Background:

As of April 21, according to the real-time epidemic data report, a total of 103,376 cases of new crown epidemic were diagnosed in China, and the cumulative number of deaths reached 4,856. Overseas, a total of 144,183,089 cases have been diagnosed so far, with a cumulative death toll of 3,054,198. The global epidemic is an indisputable fact that deserves deep consideration by people all over the world. In this battlefield without gunpowder, the hidden war is far more lethal than the naked battle. The new coronary pneumonia COVID-19 is so well disguised that it is difficult for doctors on the frontline battlefield to "see" it at first glance.

Purpose:

The aim of this study is to build an early screening model using deep learning techniques to distinguish COVID-19 pneumonia from CT images of lungs in healthy cases, and to help doctors make rapid diagnoses and improve diagnostic efficiency from an artificial intelligence perspective.

Materials and Methods:

The publicly available COVID-19 CT image dataset. The data were collected mainly through CT images in photocopies and other public sources, and indirectly through hospitals and physicians. Various deep learning models were used to classify the CT images of new coronary pneumonia.

Results:

Among several deep learning models for image classification, DResUnet had

the best results with 85.54% Accuracy and 87.02% AUC.

Conclusion:
The study showed that the developed artificial intelligence-based image analysis can achieve higher accuracy in detecting coronavirus. It is hoped that it can be widely used in hospitals to improve the efficiency and accuracy of physicians' diagnosis and to help them better summarize the clinical symptoms in their image extraction features.

Keywords: COVID-19, Coronavirus disease 2019 pneumonia, Deep learning, Convolution neural network.

List of Figures

List of Tables

Chapter 1. Introduction

1.1 Current status of COVID-19

Coronavirus infection, or COVID-19, has surprised the world with its rapid transmission, potential virulence, and potentially far-reaching overall impact on the lives of billions of people from a safety and economic perspective. To date, there are approximately 144,183,089 confirmed cases, of which 103,419 are in "mainland China", with 4,856 deaths and a mortality rate of 4.7% (figure 1.2).

The 2019 coronavirus disease (COVID-19), pandemic is an ongoing pandemic caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). the SARS-CoV-2 virus is easily transmitted between people through small droplets produced by coughing, sneezing, and talking. the COVID-19 is not only easily transmitted, but also poses a serious threat to human life. Patients infected with COVID-19 usually present with pneumonia-like symptoms such as fever, dry cough and dyspnea, and gastrointestinal symptoms, followed by a severe acute respiratory infection. the incubation period for COVID-19 is usually 1 to 14 days. Many patients with COVID-19 are not even aware that they are infected and have no symptoms. In the absence of any symptoms, this can easily cause a delay in treatment and lead to a sudden exacerbation of the disease. Therefore, a rapid and accurate method of diagnosing COVID-19 infection is essential.

Figure 1.1 Global Epidemic Distribution Map

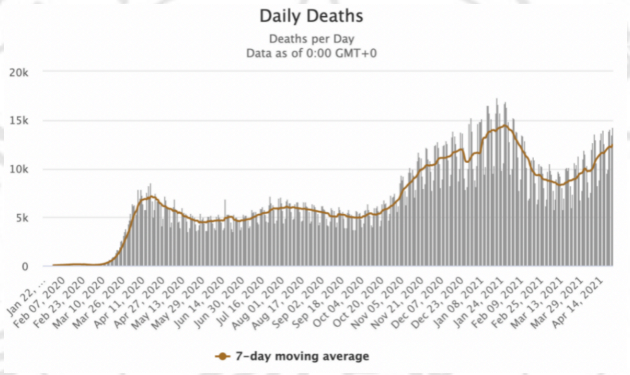


Figure 1.2 Daily death rate curve of COVID-19

1.2 Common diagnostic methods for COVID-19

It is well known that there are two most common and authoritative methods for detecting emerging coronaviruses: pharyngeal swab - kit assay, and CT Diagram assay.

1.2.1 The pharyngeal swab-kit assay

The pharyngeal swab-kit assay is also a viral assay that uses real-time reverse transcription-post hoc chain reaction (rRT- PCR) 1.3 to detect viral

RNA fragments, and its collection operation does not require equipment, is relatively convenient, relatively low cost, and is currently the predominant assay.

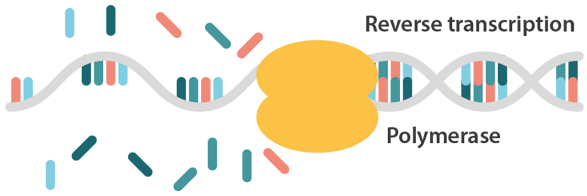


Figure 1.3 Reverse Transcription process:Conversion of RNA to DNA

However, a large number of reports have shown that because the sensitivity of RT-PCR is not high enough so the pharyngeal swab-kit assay has serious problems of insufficient precision and cannot achieve the purpose of early detection and treatment of patients presumed to have COVID-19, for example, a patient with severe macroscopic symptoms and obvious suspicion was tested 7 times before a positive viral nucleic acid reaction. Many hospitals, doctors, and patients have misjudged and delayed treatment as a result. So much so that, during the CCTV remote interview, Academician Wang Chen introduced that the current pharyngeal swab test method is not sensitive enough, and there are still differences in the sensitivity of different manufacturers' products, and many patients with significant macroscopic symptoms, cannot be measured. Wuhan Fangcai Hospital, which concentrates on the treatment of confirmed patients, no longer uses the pharyngeal swab test as the only criterion. Macroscopic disease assessment has been added, as long as the macroscopic symptoms are basically matched, they are also included in the scope of admission. The main symptoms are cough,

fever, malaise, etc. In combination with the "leukocyte mucovirus theory of novel coronavirus" proposed by the previously proposed experts, I believe that the lack of sensitivity of the pharyngeal swab assay is due to the lack of sensitivity of the specific reagents used in this method, but more importantly, it is also due to the fact that the conventional method of pharyngeal swab collection is seriously flawed for the collection of this virus. The new coronavirus has a strong adhesive property to alveolar cells and mainly accumulates in the lungs to multiply, and rarely in the pharyngeal area. In addition, it is estimated that saliva, etc. has some killing and digestive effect on this bacterium, resulting in very few neo-coronaviruses in the pharynx.

1.2.2 Diagnostic CT diagram

COVID-19 causes severe respiratory symptoms which is also associated with relatively high rates of ICU admission and mortality. Current clinical experience in treating these patients shows that RT-PCR for detection of viral RNA from sputum or nasopharyngeal swabs has a low rate of positivity in the early stages. However, a high percentage of abnormal chest CT images have been obtained from patients with this disease.

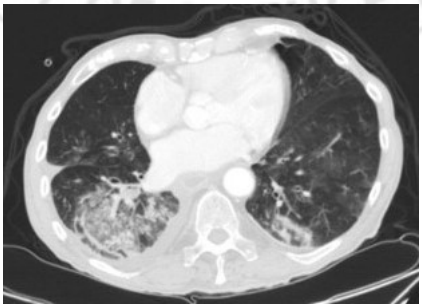


Figure 1.4 Images of healthy lungs

CT, a noninvasive imaging method, can depict certain characteristic man-

Figure 1.5 COVID-19 Characteristic manifestations of the lung

ifestations of the lung associated with COVID-19 as shown in figure 1.5.(CT of Therefore, clinicians are calling for replacing nucleic acid testing with CT of the lungs as soon as possible as one of the effective methods for early screening and diagnosis of novel pneumonia.

1.2.3 Motivation

Ai et al 2020 [1] compared the validity of the two diagnostic methods and concluded that the detection of chest CT from initial negative to positive was faster than rRT-PCR. However, the manual analysis and diagnostic process based on CT images is highly dependent on expertise, and it is also time-consuming to analyze the features of CT images. Therefore, I tried to also use deep learning (DL) methods to assist in COVID-19 diagnosis of CT scan images with reference to recent related studies, because deep learning has a very high feature extraction capability and can quickly help us to discriminate CT images of neo-coronary pneumonia from CT images of lungs of healthy patients, and I used various deep learning models such as vgg, cnn, especially focus on the use of unet and three improvements on the unet model. At the same time I added cam technology to enhance the interpretability of the model. Using mixup and other data enhancement tricks to achieve the dichotomous classification of COVID-19 image recognition. In this way,

doctors are able to initially screen suspected cases of COVID-19 in just ten to twenty seconds, greatly improving the efficiency and accuracy of diagnosis.

Chapter 2. Related work

2.1 Dataset

In recent years, DL techniques have been shown to be effective for disease diagnosis from CT images (Lit- jens et al., 2017) [2]. To enable the application of DL techniques to help detect COVID-19, an increasing number of publicly available COVID-19 datasets have been proposed.

I divided the publicly available datasets into two different categories: pre-pandemic datasets and post-pandemic datasets, which differ mainly in quality and quantity.

In the pre-pandemic period Collecting datasets for COVID-19 was a difficult task because there was not enough data to collect. Most of the datasets in this period were collected from medical papers or uploaded by public institutions. The IEEE8023 Coivd-chestxray-dataset (Cohen, Morrison, and Dao 2020) [3] is a dataset of COVID-19 cases with chest x-ray and CT images collected from public sources. However, its quality is not guaranteed because these images have not been validated by medical experts.Covid-ct-dataset (Yang et al. 2020) [4] is another CT dataset for COVID-19, consisting mainly of CT images extracted from COVID-19 research papers, and the utility of this dataset has been confirmed, so I conducted a related study on the basis of the sub-dataset.

During the pandemic The number of confirmed COVID-19 cases increased rapidly, which brought about many high-quality COVID-19 chest CT scan datasets, such as CC-CCII (Zhang et al., 2020b.) [5]

2.2 Methods of classification

DL-based method for COVID-19 detection

Many studies have been conducted on CT images, but there is a lack of research on the 3D intra-formation of CT images, such as the work of (He et al., 2020; Mobiny et al., 2020; Singh et al., 2020) [6]. These works mainly proposed 2D DL models for COVID-19 detection. (Ardakani et al., 2020) [7] Benchmarked 10 2D CNNs and compared their performance for classification of 2D CT images on their private dataset, along with 102 test images. On the other hand, there are very few studies using 3D CT images, mainly due to the lack of 3D COVID-19 CT scan datasets. (Li et al. 2020; Zheng et al. 2020) [8] proposed 3D CNNs using their private 3D CT dataset. There are also some other studies performed on X-ray images. For example, (Narin, Kaya, and Pamuk 2020) [9] proposed three 2D DL models for COVID-19 detection.(Zhang et al. 2020a) [10] Introduced a deep abnormality detection model for fast and reliable screening.(Ghoshal and Tucker 2020) [11]Investigated the detection of abnormalities by Bayesian CNNs for uncertainty and interpretability determination of X-ray images.(Alom et al. 2020) [12] used X-ray images and CT images to do segmentation and detection.

2.3 Model

2.3.1 SimpleCNN

SimpleCNN that convolutional neural network which is easy to read and use, I used a very simple 3*3 convolution here, because the structure is not complicated in figure 2.1, so the accuracy is not very high after training. The reason mainly for me to use is to do the Comparison experiments with the

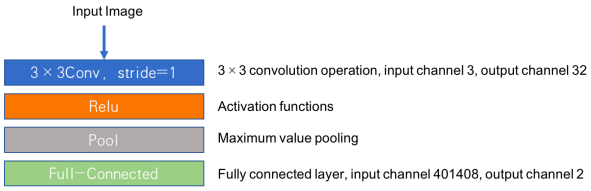


Figure 2.1 Model Structure Diagram of CNN

model we will use later.

2.3.2 VGG

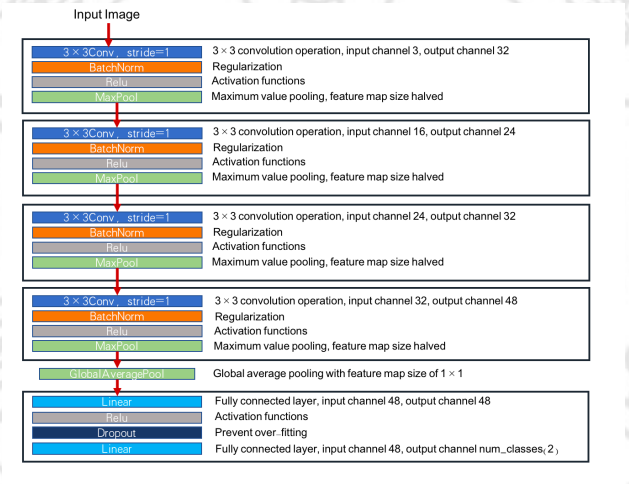


Figure 2.2 Model Structure Diagram of VGG

VGG is a little more complex than SimpleCNN. Here is the VGG model structure 2.2, I choose to use four convolutional layers and two fully connected layers to train the dataset.

VGGNet explores the relationship between the depth of a convolutional neural network and its performance, which successfully constructs a convolutional neural network with 16 19 layers deep, proving that increasing the depth of the network can affect the final performance of the network to a

certain extent, resulting in a significant decrease in error rate, while being highly scalable and generalizing to other image data.
The main features are as follows:

Small convolution kernel

The authors replaced all the convolution kernels with 3x3 (1x1 was used very rarely)

Koike chemical nucleus

In contrast to AlexNet's 3x3 pooling kernels, VGG has all 2x2 pooling kernels.

Deeper layers and wider feature maps Based on the first two points in addition: the increase in computation slows down as the convolution kernel focuses on expanding the number of channels and pooling focuses on shrinking the width and height, making the model architecturally deeper and wider at the same time.

Fully connected to convolution The network test phase replaces the three full connections in the training phase with three convolutions and the test reuses the parameters from the training, so that the full convolutional network obtained from the test can receive inputs of arbitrary width or height because there is no restriction of full connections.

2.3.3 UNet

UNet was first published in MICCAI 2015, and in just a few years, the number of citations has now reached several thousands, which is enough to see its influence. And then it became the baseline for most of the tasks doing semantic segmentation of medical images and also inspired a large number of

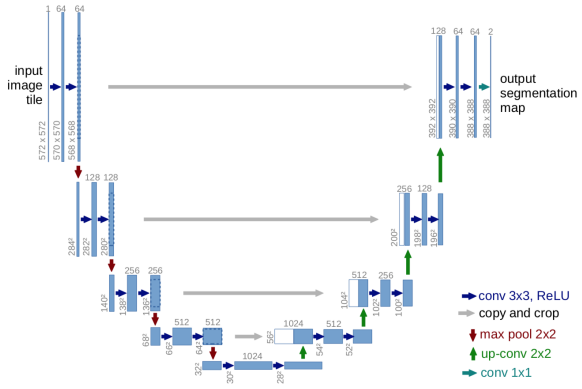


Figure 2.3 Example of Unet implementation

researchers to think about U-shaped semantic segmentation networks. Nowadays, in natural image understanding, more and more semantic segmentation and object detection SOTA models have started to focus on and use U-shaped structures, such as Semantic Segmentation Discriminative Feature Network (DFN) (CVPR2018) [13], Feature Pyramid Networks for Object Detection (FPN) for Object Detection (FPN) (CVPR 2017) [14], etc.

The structure of UNet in figure I think, has two biggest features, U-shaped structure and skip-connection (as shown in the figure 2.3)

Underlying/deep information Low resolution information after multiple downsampling. A feature that can provide contextual semantic information about the segmented target throughout the image, which can be understood as a response to the relationship between the target and its environment. This feature helps to determine the class of the object, so the classification problem usually requires only low-resolution/deep information and does not involve multi-scale fusion.

High-level/shallow information High-resolution information passed directly from encoder to decoder of the same height after concatenate opera-

tion. Capable of providing more fine-grained features for segmentation, such as gradients, etc.

What kind of characteristics does medical imaging have?

1. The image semantics are simpler and the structure is more fixed.

   We use brain CT and brain MRI for brain, chest CT for chest X-ray, and fundus OCT for fundus, all of which are imaging of a fixed organ, not the whole body. Since the organ itself is fixed in structure and not particularly rich in semantic information, both high-level semantic information and low-level features are important (UNet's skip connection and U-shaped structure come in handy).

2. Small amount of data.

   It is relatively difficult to obtain data of medical images, and many competitions only provide less than 100 cases of data. So the model we designed should not be too big, too many parameters, which can easily lead to overfitting. The number of parameters of the original UNet is about 28M (the number of UNet parameters for upsampling with transpose convolution is about 31M), and the model can be smaller if the number of channels is reduced exponentially. By reducing the number of channels twice, the number of UNet parameters is 7.75 M. By reducing it four times, the number of model parameters can be reduced to less than 2 M, which is very lightweight. Personally, I have tried using SOTA network for natural image semantic segmentation such as Deeplab v3+ and DRN on my own project, and found that the effect is similar to UNet, but the number of parameters is much larger.

3. Interpretability matters.

Since medical imaging is ultimately an aid to clinical diagnosis, it is not enough for the network to tell the doctor whether a 3D CT is diseased or not, but the doctor also wants to know further, where the lesion is, where it is in which layer, where it is segmented, and what the volume can be? The doctor also wants to know why the results of classification and segmentation given by the network, so some neural network interpretable trick is useful, more commonly used is to draw activation map to see which areas of the network are activated, which will be described in detail in 2.5.

How did I implement Unet?

The UNet model structure is relatively more complex than the first two methods Simple CNN and VGG. The code consists of two parts, unet-parts.py and unet-model.py. Four classes are defined in unet-parts.py, namely DoubleConv, Down, Up, and OutConv, and the model structure of each of these four classes is given below:

DoubleConv It can be seen from the UNet network that each layer performs two successive convolution operations regardless of the downsampling process or upsampling process. This operation is repeated many times in the UNet network, and a separate DoubleConv module can be written.

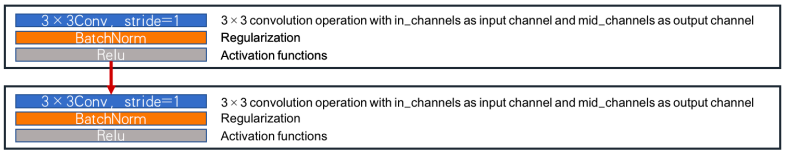


Figure 2.4 Model Structure Diagram of DoubleConv

Down module The UNet network has a total of 4 downsampling processes, a maxpool pooling layer for downsampling, and then a DoubleConv module. The pooling layer is chosen as a window size of 2 by 2, so the default is also a fill step of this size, and the size of the feature map after pooling is calculated in the same way as the convolution above. At this point, the down-sampling process of the left half of the UNet network has been completed, followed by the up-sampling process of the right half.

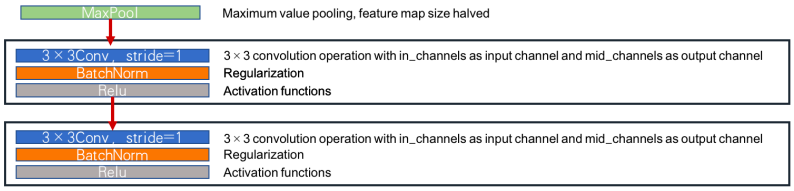


Figure 2.5 Model Structure Diagram of Down

UP module The upsampling process is of course the most used upsampling, in addition to the conventional upsampling operation, there is a fusion of features, in the code is the first init initialization function defined in the upsampling method and convolution using DoubleConv. Upsampling, two methods are defined: Upsample and ConvTranspose2d, that is, bilinear interpolation and deconvolution.

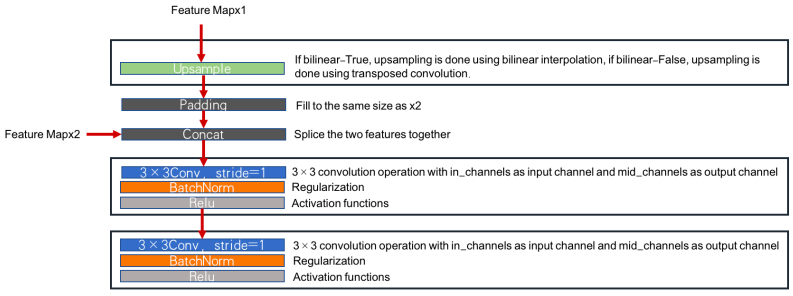


Figure 2.6 Model Structure Diagram of UP

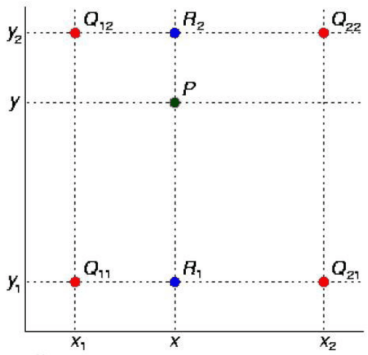


Figure 2.7 Schematic diagram of bilinear interpolation

In simple terms: four points Q11, Q12, Q21 and Q22 are known, and R1 is found by Q11 and Q21, then R2 is found by Q12 and Q22, and finally P is found by R1 and R2, and this process is bilinear interpolation 2.7. For a feature map, it is in fact to make up points in the middle of pixel points, and the value of the made up points is determined by the value of the neighboring pixel points. Deconvolution, as the name implies, is the inverse convolution. Convolution is to make the featuer map smaller and smaller, and deconvolution is to make the feature map larger and larger, such as Figure 2.8.

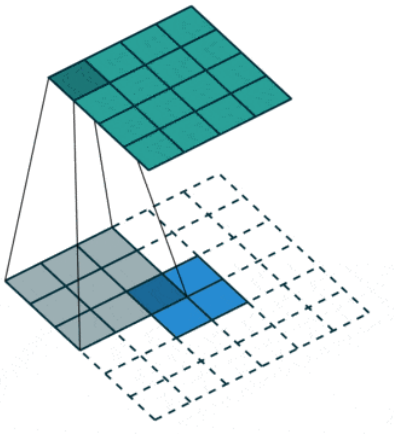


Figure 2.8 Schematic diagram of Deconvolution

The blue below is the original image, the white dashed squares around it are the padding result, usually 0, and the green above is the convolved image. This schematic diagram is a feature map process from 22 feature map to 44 feature map. In the forward propagation function, x1 receives the upsampled data and x2 receives the feature fusion data. If the two feature maps have different sizes, then the feature fusion methods can be of two kinds.

(1) Cropping the larger feature and concatting it.

(2) Fill the smaller feature and then concat.

I am using the second one, padding the smaller feature map first, and then concat.

OutConv The output of the UNet network needs to integrate the output channels according to the number of splits. The specific operation is the transformation of channel.

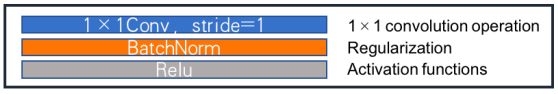


Figure 2.9 Model Structure Diagram of OutConv

This section is where the above modules are combined to form the entire UNet network in figure 2.10.

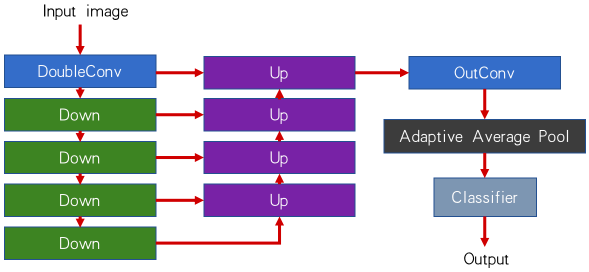


Figure 2.10 Model Structure Diagram of UNet

2.3.4 ScseResUNet

"ResUNet-ECSA", "ResUNet-FPRM" and "DResUNet which are three improved method based on the model of ScseResUNet, we firstly need understand the following small modules.

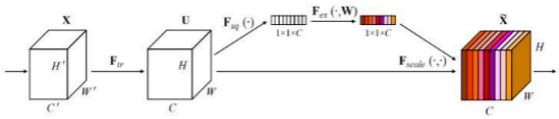


Figure 2.11 Squeeze and Excitation Module

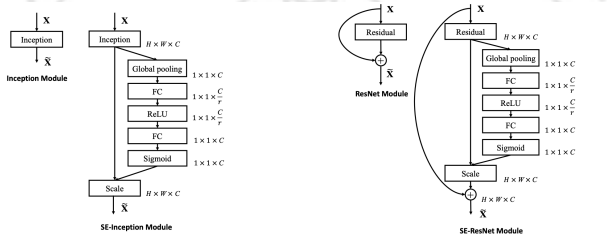


Figure 2.12 SE-Inception Module, SE-ResNet Module computation flow chart

SENet As shown in the figure 2.11: SENet introduces the Attention mechanism between channels by explicitly modeling the interdependencies between feature channels. It consists of two operations, Squeeze operation, and Excitation operation, which is an excitation operation. First, a feature map U of size (C,W,H) is input, which is subjected to the Squeeze operation. Specifically, as shown in Figure 2.12, is the first GP from the perspective of the channel that is global pooling, the space size to 1 * 1 (can be understood as a real number), the channel is still C featuremap, can be understood as C real numbers. Then comes a fully connected layer FC, in which the dimensionality of the input is generally reduced to 1/16, followed by a Relu layer and an FC layer that restores the dimensionality to the size of the input

(Excitation operation). Finally, the weights are normalized to between 0 and 1 by a Sigmoid layer, and then weighted to the previous features channel by channel by scale, thus completing a SE module operation.

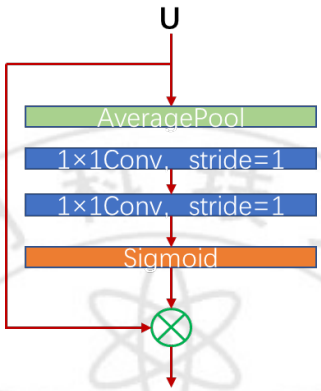


Figure 2.13 Model Structure Diagram of cSE module

cSE module Both the cSE module and sSE are improvements of the SE module, and the variants proposed on its basis can better enhance meaningful features and suppress useless ones.

In fact, the cSE module is essentially the same as the SE module. The reason is that the authors determined the hyperparameter r to be 2 or 16 through a large number of experiments, so that the accuracy of the model and the complexity of the operation are taken into account.

This module is similar to the Channel attention module in the BAM module:

1. Change the feature map from [C, H, W] to [C, 1, 1] by global average pooling method.

2. Then use two $1 \times 1 \times 1$ convolution to process the information, and finally get the C-dimensional vector.

3. Then use the sigmoid function for normalization, to get the corre-

sponding mask.

4. Finally, by channel-wise multiplication, we get the feature map calibrated by information.

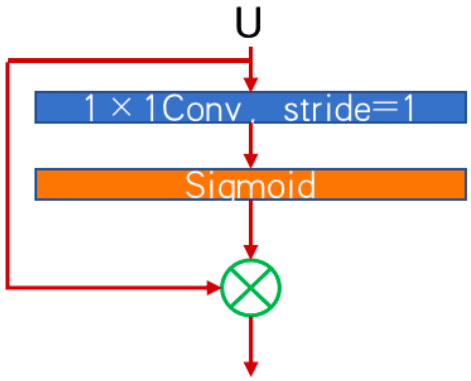


Figure 2.14 Model Structure Diagram of cSE module

sSE module The sSE module introduces the attention mechanism from another perspective, namely the spatial perspective. It starts with a 1*1 convolutional dimensionality reduction and activation with a Sigmoid function to obtain a feature map of 1*H*W dimensions. Then the features are rescaled and multiplied with the original U on the corresponding space to get U.

1. Directly use 1×1×1 convolution on the feature map, from [C, H, W] to [1, H, W] of the features. 2. Then use sigmoid to activate to get spatial attention map. 3. Then directly applied to the original feature map to complete the spatial information calibration.

ECA module The effective channel attention ECA module, which adds only a small number of parameters, achieves significant performance gains. Through the analysis of the channel attention module in senet, the authors'

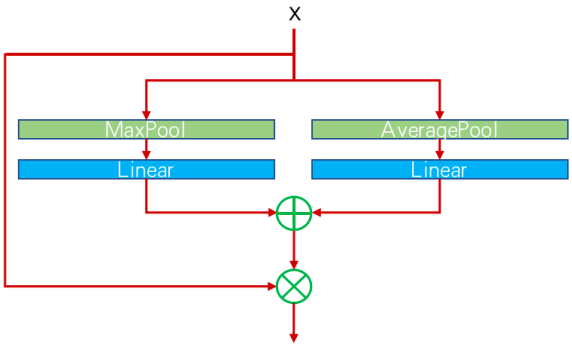


Figure 2.15 Model Structure Diagram of ECA module

experience shows that avoiding dimensionality reduction is very important for learning channel attention, and that appropriate cross-channel interactions can significantly reduce model complexity while maintaining performance, because the authors propose a local cross-channel interaction strategy without dimensionality reduction, which can be effectively implemented by one-dimensional convolution.

Here I use both a maximum pooling and an average pooling for the input x to make feature fusion in Figure 2.15 which can help us focus on the information we need better.

Maximum-pooling Pooling in CNN is used to reduce the dimensionality of features while extracting better features with stronger semantic information. More typically, for example, MaxPooling is used in VGGNet to reduce the dimensionality of the features, while extracting the largest and most strongly responsive parts of the features for input to the next stage of the module.

Average-pooling It is used when the information in the features all have a certain contribution, such as when the network goes deeper, when the H W

of the feature map are smaller and contain more semantic information, this time it is less appropriate to use MaxPooling again. Typically, for example, ResNet uses AvgPooling with Kernel Size = 7 to reduce the dimensionality before inputting the fully connected layer.

The performance of max-pooling and average-pooling is still very useful for designing convolutional networks. Although the pooling operation is not very effective in improving the overall accuracy, it is still very useful in reducing parameters, controlling overfitting, and improving model performance and saving computational power, so pooling is an indispensable operation for convolutional design. I will use it when I design the model in part 3.1.

2.4 Data enhancement

Data enhancement mainly refers to the data enhancement of images in the field of computer vision, so as to make up for the lack of training image data set and achieve the purpose of expanding the training data. Data enhancement is a data expansion method, which can be divided into two ways: similar enhancement (e.g., flip, rotate, scale, shift, blur, etc.) and mixed class enhancement (e.g., mixup).

2.4.1 Peer Enhancement

(1) Flip Flip

Can be divided into horizontal flip, vertical flip.

(2) Rotation.

(3) Scaling(Outward Scaling,Inward Scaling)

When scaling outward, the final image size will be larger than the original image size, most image frames cut out a part from the new image and its size is equal to the original image. Scaling inward, because it will reduce the image size which forces us to make assumptions about what is beyond the boundary.

(4) Random Crop (Random Crop)

Unlike scaling, random crop is just a random sampling of a part from the original image, and then we resize this part to the size of the original image.

(5) Shift (Translation)

Translation involves moving the image only in the X or Y direction (or both). This enhancement method is very useful because most objects can be located almost anywhere in the image and we need to make assumptions about boundaries when shifting.

(6) Blurring (Gaussian Noise)

Overfitting usually occurs when your neural network tries to learn high frequency features (features that occur in large numbers) that may be useless. Gaussian noise with zero mean has data points in essentially all frequencies, thus effectively distorting high frequency features. But this also means that data at lower frequencies (usually your expected data) will also be distorted, but your neural network can learn to outperform it. Adding the right amount of noise can enhance the network's ability to learn.

2.4.2 Multi-graph fusion

Mixup is mainly used for image classification, two samples are randomly selected from the training samples for a simple random weighted summation, while the labels of the samples also correspond to the weighted summation then the predicted results are lost with the labels after the weighted summation and the parameters are updated in the reverse derivation the formula is defined as figure follows.

$$\lambda = Beta(\alpha, \beta)$$

$$mixed\_batch_x = \lambda * batch_{x1} + (1 - \lambda) * batch_{x2}$$

$$mixed\_batch_y = \lambda * batch_{y1} + (1 - \lambda) * batch_{y_2}$$

As can be seen from the formula, the weighted fusion acts on both image and label dimensions.

The specific use will be described in the method section.

CutMix is also relatively simple, also for a pair of images to do the same operation. Simply speaking is to randomly generate a crop box Box, cropping off the corresponding position of the A picture, and then use the corresponding position of the B picture ROI into the cropped area of the A picture to form a new sample. The same weighted summation is used to calculate the loss of the solution.

The operation of merging two pictures is defined. The sampling equation

$$\begin{aligned} \tilde{x} &= \mathbf{M} \odot x_A + (\mathbf{1} - \mathbf{M}) \odot x_B \\ \tilde{y} &= \lambda y_A + (1 - \lambda) y_B, \end{aligned} \tag{1}$$

for the bounding box of the cropped area is as follows.

$$\begin{aligned} r_x &\sim \text{Unif}\ (0, W)\,, \quad r_w = W\sqrt{1-\lambda}, \\ r_y &\sim \text{Unif}\ (0, H)\,, \quad r_h = H\sqrt{1-\lambda} \end{aligned} \tag{2}$$

W, H is the width and height size of the binary mask matrix M and the scale of the clipping region satisfies as follows.

$$\frac{r_w r_h}{WH} = 1 - \lambda$$

After determining the cropping region B, setting the cropping region B in the binary mask M to 0 and the other regions to 1. This completes the sampling of the mask M. Then, the M dot product A removes the cropping region B from sample A, and the (1-M) dot product B fills the cropping region B in sample B with cropping to sample A, forming a brand new sample.

Mosaic can be said to be one of the highlights of YOLOv4 Mosaic mixes 4 training images, thus mixing 4 different contexts, enriching the context of the detected objects and calculating four images at once during BN calculation, while CutMix only mixes 2 input images, which is the reason why Mosaic is stronger. We can understand that Mosaic mixes more images to create more possibilities to see more.

Implementation ideas:

1.Read four images at a time.

2.Respectively, the four images are flipped, scaled, color gamut changes, etc., and placed in accordance with the four directional positions.

3.Perform the combination of pictures and the combination of frames.

2.5 Class Activation Mapping Algorithm

Everyone should have seen the image generated by the thermal imager on TV as the figure 2.16. The animal or person in the image can be clearly seen because of the heat emitted. The CAM (Class Activation Mapping) that produces a similar CAM map, which shows the basis of its decision in the form of a Heat map. when we need a model to explain the reason for its classification, just like telling us in the darkness of the night where there are hot objects.

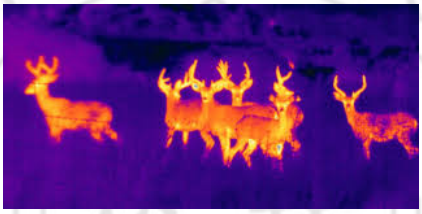


Figure 2.16 Thermal image of Animal

For a deep convolutional neural network, after multiple convolutions and pooling, its last convolutional layer contains the richest spatial and semantic information, and further down are the fully connected and softmax layers as figure 2.17, which contain information that is difficult for humans to understand and visualize. Therefore, to make the convolutional neural network's give a reasonable explanation of its classification results, it is necessary to make full use of the last convolutional layer.

CAM [15] borrows the idea from the famous paper Network in Network [16] and uses GAP Global Average Pooling to replace the fully connected layer. GAP can be considered as a special average pooling layer, except that its pool size is as large as the whole feature map, which is actually the average value of all pixels in each feature map 2.18.

The advantages of GAP are clearly stated in NIN's paper [16] : without

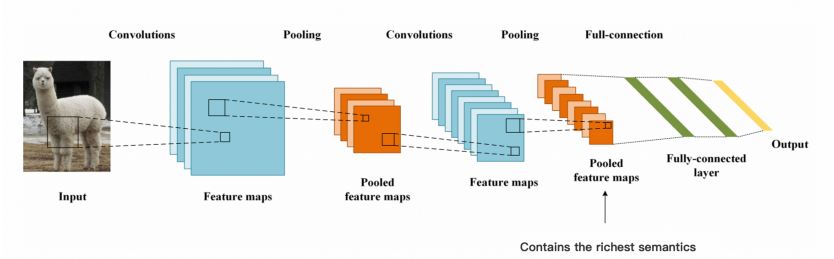


Figure 2.17 Model Structure Diagram of CAM Full-collection

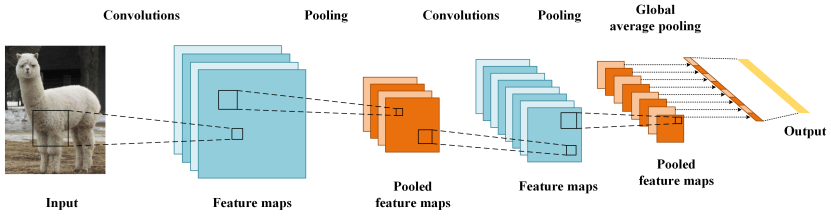


Figure 2.18 Model Structure Diagram of CAM Global average pooling

the fully connected layer, the input does not have to be fixed in size, so it can support any size of input. In addition, the introduction of GAP makes full use of the spatial information. Without the various parameters of the fully connected layer, it is robust and less prone to overfitting. Another important point is that the final mlpconv layer (also known as the last convolutional layer) forces the generation of feature maps that are consistent with the number of target categories.

After GAP, we can get the mean of each feature map in the last convolutional layer, and the output is obtained by weighting and summing (in practice, it is the input of the softmax layer). Note that for each category C, the mean of each feature map k has a corresponding w. This is the basic structure of the CAM, and the following is the same as the ordinary CNN model training can be. After the training is complete is the main event: how do we get a heat map for interpreting the classification results? For example, if we want to explain why the classification result is alpaca, we take out all

the values corresponding to the category of alpaca and find the weighted sum of them and their corresponding feature maps. Since the size of this result is the same as the feature map, we need to upsample it and superimpose it onto the original map as figure 2.19.

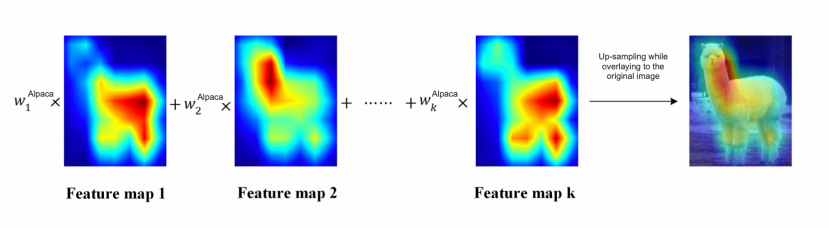


Figure 2.19 Feautre map of Alpaca

In this way, CAM tells us, in the form of a heat map, which pixels the model is focusing on to determine that the image is an alpaca.

Chapter 3. Method

3.1 Training Model

DResUnet model The main method I used and improved for model training is the DResUnet model, and I will then go through the main structure to describe how I used it to implement image classification for COVID-19.

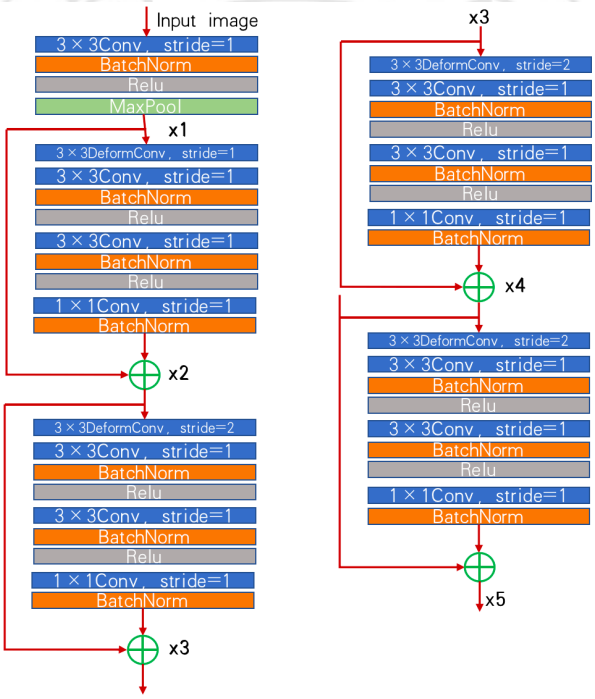


Figure 3.1 Model Structure Diagram of Feature extraction module

Firstly I used five feature extraction modules x1,x2,x3,x4,x5.

x1: The input image is first processed by a 3*3 convolution with a stride of 1 to make a linear change, then a regularization is done by a batchnnorm layer to prevent overfitting of the data before the activation function of relu

is used to make the negative pixel values become 0 in order to increase the nonlinearity of the model, finally a downsampling of the image is done by Maxpool.

x2: Unlike x1, we first pass a Deform convolution of 3*3, stride 1, mainly to make the learned offset more focused on the top edge of the CT map, that is, to extract more focused features to help classification. Next, two more convolutions of 3*3, stride of 1 are superimposed on the input image for processing, regularization and relu activation, respectively, and another convolution of stride of 1, 1*1 is superimposed, followed by a regularization to reduce the order of magnitude of the feature variables and a residual with the x1 feature extraction module. Without residuals, we will find that as the depth of the network deepens, the training error will first decrease and then increase. From theoretical analysis, the deeper the network depth, the better. However, in practice, without a residual network, for a normal network, deeper depth means harder to train with an optimization algorithm. In fact, as the depth of the network increases, the training error will become more and more.

x3 x4: The structure of x3 and x4 is basically similar to that of x2. The only difference is that at the beginning with the 3*3 deformconv, the stide is set to 2, because CNN is a fixed convolution structure for a given step size, and the deformconv is provided to improve the modeling ability of the deformation, so that the convolution kernel is shifted at the sampling point of the input sampling map to focus on our target of interest.

x5: The difference in x5 is that after the second 3*3 convolution, I added an ECAse module as the figure 3.2.

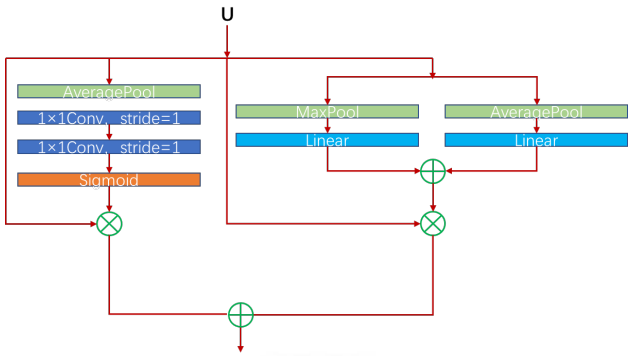


Figure 3.2 Model Structure Diagram of ECAse

It consists of sSE module and ECA module. Firstly, the sSE module uses 1×1×1 convolution on the feature map from [C, H, W] to [1, H, W], and then activates with sigmoid to get the spatial attention map, while ECA performs maximum pooling and average pooling on the feature map respectively, and downsamples the image by 2*2, stride is 2 convolution, so that the image is reduced to 1/2 of the original, which reduces the computation and increases the nonlinearity.

Immediately after the up-sampling process, five up-samplings are performed as figure 3.3.

x44, x33, x22: Firstly, upsampling is performed, scale is 2. After extracting features by convolution, the output size tends to become smaller, and we need to restore the image to its original size for further computation, using expanded image size to realize the mapping of the image from small resolution to large resolution. Then padding is added, mainly to make up for the difference in the input size of the image, making the input image size consistent.

Immediately afterwards, by concat, the input of each convolutional layer

is the splicing of the output of the previous convolutional layers, so that i.e., each time the features obtained earlier are combined to obtain the subsequent features. The fusion of features is enhanced. In this way, the process of convolution, regularization, and activation function is performed twice for 3*3.

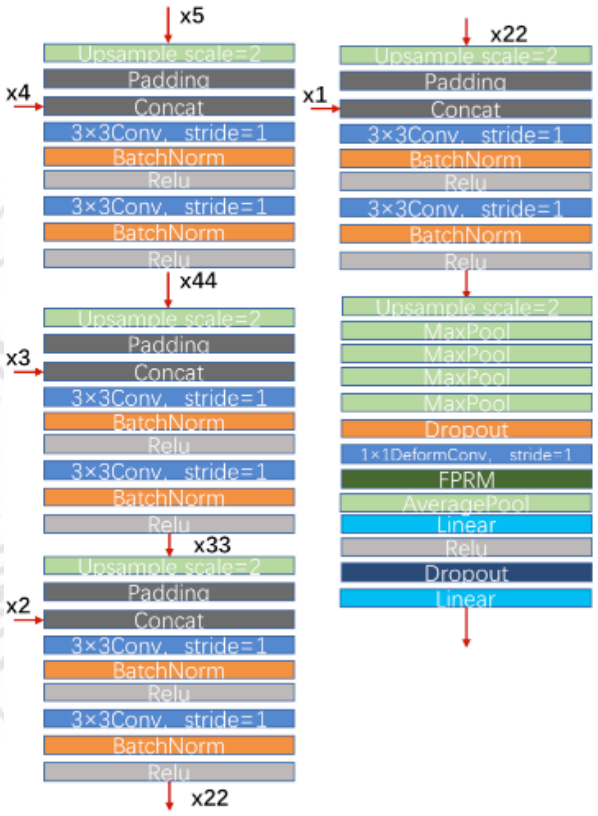


Figure 3.3 Model Structure Diagram of Upsample

After the last upsampling, a feature of size 1*1 is output by four consecutive maximum pooling, and then a dropout layer is passed to output the final probability. Overfitting can be significantly reduced by ignoring half of the feature detectors in each training batch (leaving half of the hidden layer nodes with a value of 0). This approach reduces the interactions between feature detectors (hidden layer nodes). It makes the model more generalizable. The

features are then focused by a 1*1 deform convolution. Especially, I added a module of fprm here, mainly to facilitate the fusion of information between high-level semantic information and low-level features, so that it can have rich semantic information and also preserve more information of low-level. An averaging pooling layer was used immediately afterwards to output the output results through two consecutive fully linked layers to output the CT probability of COVID-19 we need in order for us to determine whether the suspected case has infected COVID-19.

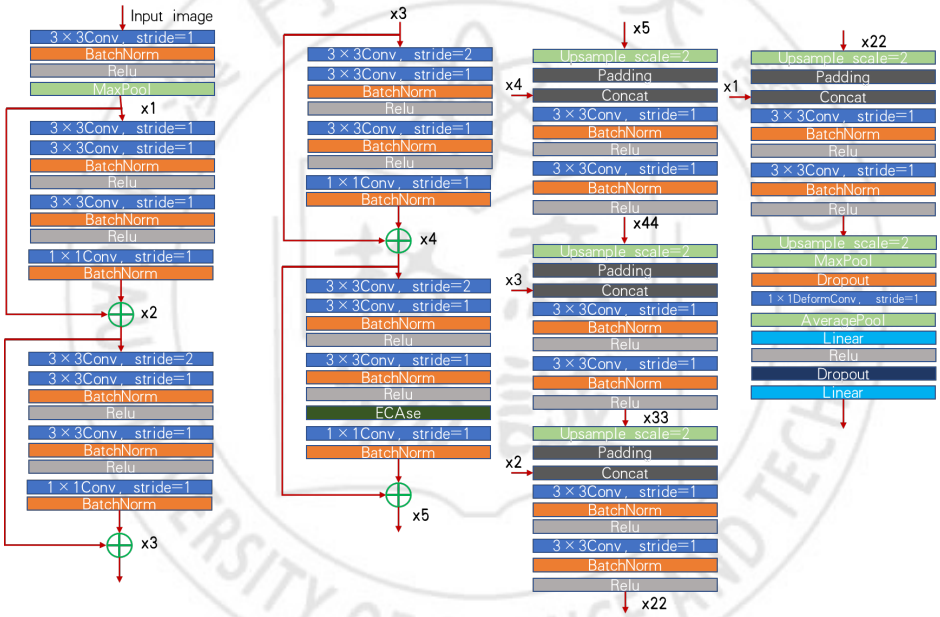


Figure 3.4 Model Structure Diagram of DResUNet

3.2 Data enhancement

The method I use here for data augmentation is mixup which belongs to class augmentation was introduced in section 2.4.2. I will go through the related code as figure 3.5 to describe the implementation.

```python
def mixup_data(x, y, alpha=1.0, use_cuda=True):
    '''Returns mixed inputs, pairs of targets, and lambda
    the larger the alpha, the more it can suppress overfitting.
    recommend alpha=0.4
    '''
    if alpha > 0:
        lam = np.random.beta(alpha, alpha)
    else:
        lam = 1

    batch_size = x.size()[0]
    if use_cuda:
        index = torch.randperm(batch_size).cuda()
    else:
        index = torch.randperm(batch_size)

    mixed_x = lam * x + (1 - lam) * x[index, :]
    y_a, y_b = y, y[index]
    return mixed_x, y_a, y_b, lam

def mixup_loss_fn(loss_fn, pred, y_a, y_b, lam):
    return lam * loss_fn(pred, y_a) + (1 - lam) * loss_fn(pred, y_b)
```

Figure 3.5 The implementation code of Mixup

Firstly calculate the value of lambda, then we can obtain the number of samples in the current batch and perform random sorting to generate the corresponding labels based on a random mixture of x samples. Finally, we calculate the loss by the formula and then call our mixup in the training file.

Mixup is trained in a virtual sample (interpolation of two random samples and labels to build the sample). It is worth trying to integrate this method into an existing model with only a few lines of code, without affecting the model recognition speed at all, and with almost no computational overhead.

3.3 Grad-CAM

In section 2.5 I introduced CAM, and in my experiments I mainly used the Grad-CAM method, because cam has the disadvantage that it requires modifying the structure of the original model, resulting in the need to retrain the model, which greatly limits its usage scenarios. If the model is already online or the training cost is very high, it is almost impossible for us to retrain it. So I choose to use grad-cam here to increase the interpretability of the

images.

The basic idea of Grad-CAM is the same as that of CAM, which is to obtain the weights of each pair of feature maps and finally find a weighted sum. However, the main difference between CAM and Grad-CAM is the process of finding the weights. CAM obtains the weights by replacing the fully connected layers with GAP layers and retraining them, while Grad-CAM takes a different approach and uses the global average of gradients to calculate the weights.

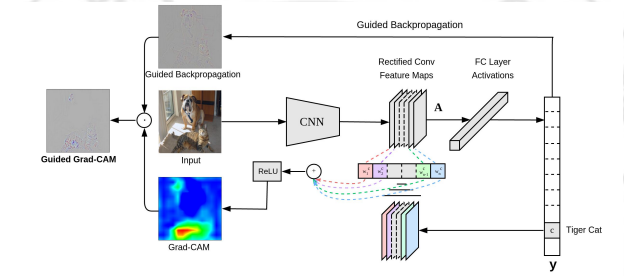


Figure 3.6 Model Structure Diagram of Grad-CAM

Note that another difference between Grad-CAM and CAM is that Grad-CAM adds a ReLU to the final weighted sum. The reason for adding such a ReLU layer is that we only care about those pixels that have a positive effect on category c. If we do not add the ReLU layer, we may end up bringing in some pixels that belong to other categories, thus affecting the interpretation.

The Grad-CAM makes each model in our experiments highlight in the focal region, which indicates that the judgment of our model is well-founded and reasonable, and enhances the interpretability, which will be verified in section 4.4.

Chapter 4. Experiment

4.1 Dataset

In this paper, I use one publicly availavle dataset: COVID-CT-DATASET [17], the authors collected 760 pre prints on COVID-19. Many of these preprints reported patient cases of COVID-19, and some of these reports showed CT images. For each CT image, the authors also collected meta-information extracted from the papers, such as the patient's age, gender, location, medical history, and scan results. For any sub-image containing multiple CT images, they manually segmented them into individual CTs as shown in Figure 4.1 (left).

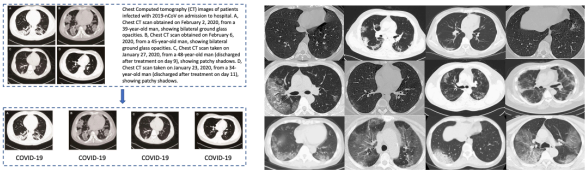


Figure 4.1 (left)For any figure containing multiple CT images as subgraphs, they manually segmented them into individual CTs.(Right) Example of COVID-19-positive CT images.

A total of 349 CT images were obtained that were labeled as positive for COVID-19. These CT images were available in different sizes. These images are from 216 patient cases. Figure 4.1 (right) shows some examples of COVID19 CT images. For the patients marked positive, 169 of them have age information and 137 have gender information. Figure 4.2 (left) shows the age distribution of COVID-19 patients. Figure 4.2 (right) shows the gender ratio of COVID-19 patients. There were more male patients than female

patients, with numbers of 86 and 51, respectively.

Non-COVID-19 CT images were collected as a negative training set as table 4.1 to diagnose the binary classification model of COVID-19, we collected a set of non-COVID-19 CT images as negative training examples in addition to 349 COVID-19 CT images.

The sources of these images include: - MedPix6 database, which is an open online database of medical images, teaching cases and clinical topics. It contains more than 9,000 topics and 59,000 images from 12,000 patient cases.

- The LUNA7 dataset, which contains CT scans of lung cancer from 888 patients.

- The Encyclopedia of Radiology website , containing radiology images from 36,559 patient cases.

- PubMed Central (PMC) This is a free full-text archive of biomedical and life sciences journal literature. Some papers contain CT images.

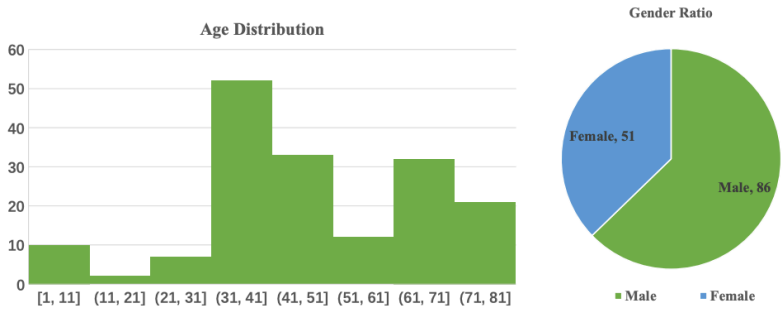


Figure 4.2 (Left) Age distribution of COVID-19 patients. (Right) The gender ratio of COVID-19 patients. The ratio of male:female is 86:51.

Collection of test and validation images To evaluate the trained model, the authors collected a validation set and a test set. In both sets, all images

Table 4.1 Statistics of the negative training set

	Class	LUNA	MedPix	PMC	Radiopaedia	Total
patients	Non-COVID	17	55	55	2	55
images	Non-COVID	36	195	202	30	463

Table 4.2 Statistics of the test set

	Class	LUNA	COVID-Seg	Radiopaedia	Total
patients	COVID	0	4	0	4
	Non-COVID	19	0	1	20
images	COVID	0	173	0	173
	Non-COVID	164	0	4	168

are original CTs donated by the hospital.

Table 4.2 shows the composition of the test set. It has 173 COVID-19 CT images from 4 patients in the COVID-Seg dataset. It contains 168 non-COVID-19 CT images. 164 of them are from 19 patients in the LUNA dataset and the remaining 4 are from 1 patient in Radiopaedia.

Table 4.3 shows the composition of the validation set. It has 88 COVID-19 CT images from 4 patients in the COVID-Seg dataset. It contains 64 non-COVID-19 CT images. 48 of them are from 38 patients in the LUNA dataset and the remaining 16 are from 1 patient in Radiopedia.

Table 4.3 Statistics of the validation set

	Class	LUNA	COVID-Seg	Radiopaedia	Total
patients	COVID	0	4	0	4
	Non-COVID	38	0	1	39
images	COVID	0	88	0	88
	Non-COVID	48	0	16	64

4.2 Experimental setup

Input images: Adjusted to 480×480.

Data enhancement: We perform data enhancement on the training set. Each training image was randomly cropped with a scale of 0.5, horizontally flipped, random contrast, and random luminance with a factor of 0.2.

Weight parameters:The weight parameters in the network were optimized using Adam (Kingma and Ba, 2014) with an initial learning rate of 0.0001 and a mini-batch size of 16. Throughout the training process, the learning rate was adjusted using cosine scheduling with a period of 10.

We implemented the network using PyTorch 1.0, cuda 9.0, and trained on a GTX 1080Ti GPU.

Training Models: SimpleCNN, VGG, UNet, ResUNet, ResUNet-ECSA, ResUNet-FPRM, DResUNet.

4.3 Training model

I wrote a model list as figure 4.3 and a ScseResUnet in the train file so that I could select the model to use.

```
model_list = ["UNet","ScseResUNet","SimpleCNN","VGG"]
ScseResUNet_model_list = ["ResUNet_ECSA",'ResUNet_FPRM',"DResUNet"]
ScseResUNet_model_name = ScseResUNet_model_list[0]
model_name = model_list[3]
```

Figure 4.3 Model list

Take the example of VGG:

Making predictions by the obtained probabilities, greater than 0.5 to determine the infection of new crowns and less than 0.5 for healthy cases (figure 4.4) so that I can make a judgement to know which suspected case has infected the COVID-19 according to the CT diagram.

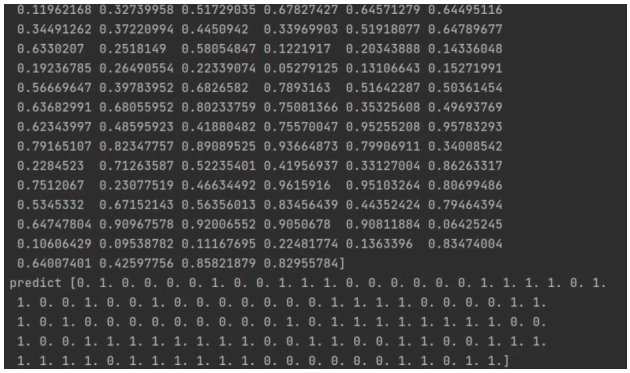


Figure 4.4 The progress of training VGG

Tensorboard For most people, deep neural network is like a black box, its internal organization, structure, and its training process are hard to understand, which brings a great challenge to the understanding and engineering of deep neural network principles. To solve this problem, tensorboard, a visualization tool built into tensorflow, makes it easier and more efficient to understand, debug, and optimize tensorflow programs by visualizing the information in the output log files of tensorflow programs.

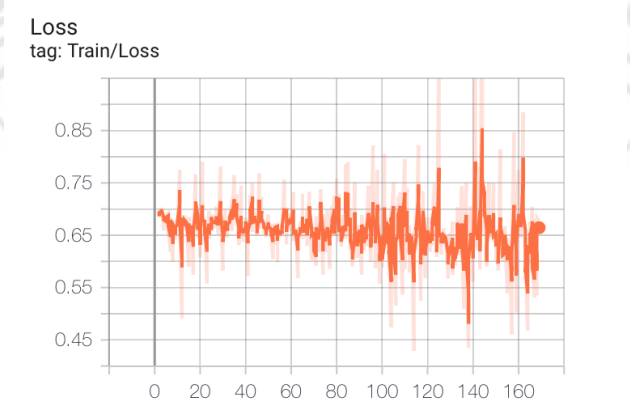


Figure 4.5 Loss fluctuation diagram of the DResUnet

Through the tensorboard I drew the fluctuation diagram of the loss during the training of the model DResUnet (figure 4.5), we can see that it fluctuates a lot up and down and does not converge, at this time we need to adjust our

hyperparameters in order to improve the accuracy.

First, I adjusted the learning rate from large to small, suggesting that each time divided by 5. When the learning rate is adjusted, go to adjust the size of the batchsize, if the batchsize is adjusted 2 times larger, the learning rate will be adjusted correspondingly larger, and vice versa, the learning rate is adjusted correspondingly smaller.

4.4 Use Grad-CAM to generate heat maps

When the model is trained, we can use his weights to generate the heat map of the target image.(figure 4.4)

Generate heat map method

python cam.py –image-path ./... –model-path ./... –method ...

Take the VGG model as an example:

python cam.py –image-path ./Images-processed/CT-COVID/2020.01.24.919183-p27-135.png –model-path ./model-backup/medical-transfer/VGG.pth –method gradcam

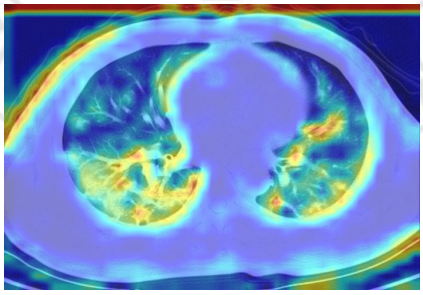


Figure 4.6 Using Grad-CAM to draw the heat map of VGG

Chapter 5. Results and Analysis

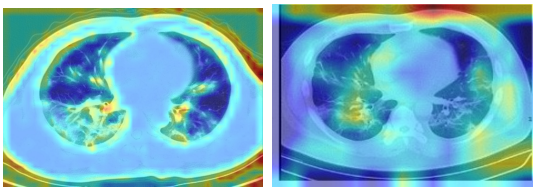
5.1 Results

Evaluation Metrics In the experiment, I used the following metrics: Accuracy, Precision, AUC, Recall, F1-score to evaluate the experimental results in table 5.1. From the table, we can easily see that the main improvement of DResUnet achieved an accuracy of 85.54, which is a good result. But it did not reach my expectation. There are several possible reasons, one is the relatively small amount of data, the training volume is not enough, and maybe is the difficulty of the dataset itself image. Finally, the generalization is not strong and the parameters have not been adjusted to the optimal, these possible reasons all lead to the poor results of the model training.

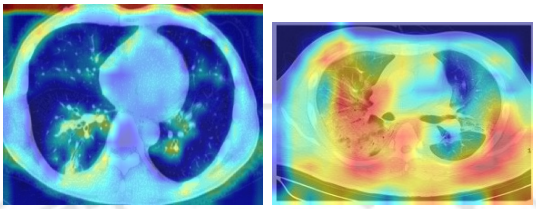
Table 5.1 The experimental results of different deep learning models

Model	Input size	Accuracy %	Precision %	AUC %	Recall %	F1-score
SimpleCNN	256×256	63.56	66.67	72.61	51.72	0.5825
VGG	256×256	70.34	67.69	73.02	75.86	0.7154
UNet	256×256	75.22	78.42	81.23	49.23	0.7821
ResUNet	256×256	80.69	82.09	75.52	43.45	0.7139
ResUNet_ECSA	256×256	84.78	**90.48**	84.08	35.87	**0.8811**
ResUNet_FPRM	256×256	82.67	87.35	83.89	36.23	0.8035
DResUNet	256×256	**85.54**	89.62	**95.02**	**30.29**	0.8292

Interpretability Although our model achieved good results in detecting COVID-19 in CT images, the classification results by themselves are not good enough to further help physicians in clinical diagnosis without justifying the intrinsic mechanism that led to the final decision. To examine



(a) Test-COVID1 (b) Test-COVID2



(c) Test-COVID3 (d) Test-COVID4

Figure 5.1 Effectiveness of GRAD-CAM for testing four different CT maps of COVID-19

the intrinsic mechanism of the DResunet model, we applied Grd-Cam to it ([18]). Grd-CAM is an algorithm that visualizes the discriminative lesion regions of interest to the model. In Figure 5.1, I applied CAM on the CT-COVID-DATASET of COVID-19 images which shows that the red areas and brighter regions have a greater impact on the model's determination of COVID-19 classes, while the blue and darker regions have less impact. The CAM makes the DResUnet model pre-determinable and helps physicians to quickly locate distinguishable lesion areas.

Chapter 6. Conclusion

6.1 Conclusion

In this work, I introduced and used a variety of deep learning models to predict ct maps of neocoronary pneumonia and achieved binary classification of neocoronary pneumonia images, especially the dresunet model with modules such as ecase and fprm to promote the fusion of high level semantics with low level semantics, in addition to the mixup data augmentation method, which eventually led to the model achieving 85.54 accuracy of the model. At the same time, I also used grad-cam to enhance the interpretability of the model to help physicians better diagnose the clinical symptoms.

6.2 Future work

Now the open source dataset of the COVID-19 is still relatively small, some datasets in which the variability of data is still relatively large. It hard to find the suitable dataset to do related researches. At the same, different data formats lead to researchers want to merge data, data fusion is more difficult, whether the future can improve and have a unified format of datasets dedicated to researchers to do research. As well as most of the COVID-19 work is now focused on 2D data sets including the work in this paper, I hope we can focus on the work in 3D data sets afterwards, because the real CT scan is a 3D image to get a high quality model.

So far, the epidemic in China has been better controlled, but the epidemic in foreign countries is still very serious, especially in India, Iran and other

countries do not have enough resources and medical technology to solve the new crown pneumonia, there are many suspected cases of new crown pneumonia every day waiting for assistance because of the lack of doctors so that they fail to get timely diagnosis. COVID-19 is a global problem we also need to consider collecting data from patients in countries such as India and Iran to help them control the local epidemic in the future.

Bibliography

[1] 2020 Ai et al. A Deep Learning Interpretable Model for Novel Coronavirus Disease (COVID-19) Screening with Chest CT Images. 2020.

[2] Lit jens et al. Feature Isolation for Hypothesis Testing in Retinal Imaging: An Ischemic Stroke Prediction Case Study. 2017.

[3] Morri-son Cohen and Dao. iEEE8023 Coivd-chestxray-dataset. 2020.

[4] Yang et al. COVID-CT-Dataset: A CT Scan Dataset about COVID-19. 2020.

[5] Zhang et al. Clinically Applicable AI System for Accurate Diagnosis, Quantitative Measurements, and Prognosis of COVID-19 Pneumonia Using Computed Tomography. 2020b.

[6] Xiaowen Chu Shaohuai Shi Jiangping Tang Xin Liu Xin He, Shihao Wang. Automated Model Design and Benchmarking of 3D Deep Learning Models for COVID-19 Detection with Chest CT Scans. *AAAI 2021*, 2021.

[7] Ali Abbasian Ardakani, U Rajendra Acharya, Sina Habibollahi, and Afshin Mohammadi. Covidiag: A clinical cad system to diagnose covid-19 pneumonia based on ct findings. *European radiology*, 31(1):121–130, 2021.

[8] Yu Shi, Gang Wang, Xiao-peng Cai, Jing-wen Deng, Lin Zheng, Haihong Zhu, Min Zheng, Bo Yang, and Zhi Chen. An overview of covid-19. *Journal of Zhejiang University. Science. B*, page 1, 2020.

[9] Ali Narin, Ceren Kaya, and Ziynet Pamuk. Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks. *arXiv preprint arXiv:2003.10849*, 2020.

[10] Juanjuan Zhang, Maria Litvinova, Yuxia Liang, Yan Wang, Wei Wang, Shanlu Zhao, Qianhui Wu, Stefano Merler, Cécile Viboud, Alessandro Vespignani, et al. Changes in contact patterns shape the dynamics of the covid-19 outbreak in china. *Science*, 368(6498):1481–1486, 2020.

[11] Biraja Ghoshal and Allan Tucker. Estimating uncertainty and interpretability in deep learning for coronavirus (covid-19) detection. *arXiv preprint arXiv:2003.10769*, 2020.

[12] Md Zahangir Alom, MM Rahman, Mst Shamima Nasrin, Tarek M Taha, and Vijayan K Asari. Covid_mtnet: Covid-19 detection with multi-task deep learning approaches. *arXiv preprint arXiv:2004.03747*, 2020.

[13] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Learning a discriminative feature network for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1857–1866, 2018.

[14] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[15] Agata Lapedriza Aude Oliva Bolei Zhou, Aditya Khosla. Learning Deep Features for Discriminative Localization. In *arXiv:1512.04150*, 2016.

[16] Qiang Chen2 Shuicheng Yan2 Min Lin1, 2. Network In Network. In *Neural and Evolutionary Computing (cs.NE)*, 2013.

[17] Xuehai He Xingyi Yang. COVID-CT-Dataset: A CT Image Dataset about COVID-19. In *arXiv:2003.13865v3 [cs.LG]*, 2020.

[18] Abhishek Das Ramakrishna Vedantam Devi Parikh Dhruv Batra Ramprasaath R. Selvaraju, Michael Cogswell. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. In *International Journal of Computer Vision (IJCV) in 2019*, 2019.

Appendix

Code source can be found in the link:

https://pan.baidu.com/s/1b3lp3W8jQq8lcizfs72-qw

Password: wcss

List of appendix:

1. deform-conv-v2.py
2. scse-resunet.py
3. scse.py
4. SimpleCNN.py
5. unet-model.py
6. unet-parts.py
7. VGG.py
8. cam.py
9. utils.py
10. train.py

Note: All Appendix files are under "COVID" directory.

Curriculum Vitae

Personal Data:

Name:	Tianhao Xu
Birthday:	24/01/1999
Email:	573152970@qq.com
Research Area:	Computer vision

Educational Background:

09/2014 - 06/2017	High School, NO.1 High School Of Zhenjiang, Jiangsu Province
09/2017 - 06/2021	Bachelor of Science, Bachelor of Science, Macau University of Science and Technology

Work Experience:

07/2020 - 08/2020	Genzon Investment Group, Shenzhen
04/2020 - 06/2020	SAIC Motor Corporation Limited (SAIC Motor), Shanghai
07/2019 - 08/2019	Jiangsu Science and Trade Information Technology Co., Ltd.,Jiangsu
06/2019 - 07/2019	China Telecom, Jiangsu

Publications

[1] Stock Price Prediction Based on Artificial Neural Network, MLBDBI 2020

Acknowledgements

As my thesis is approaching, I would like to take this opportunity to thank my advisor, Professor Subrota Kumar Monda, for not only giving us regular meetings once a week, but also answering our questions in his free time, giving us patient and professional guidance on topic selection, problem analysis, thesis format, etc.

At the same time, the successful completion of the thesis could not have been achieved without the care and help of other teachers, students and friends. Throughout the process of writing the dissertation, they helped me check the information and provided suggestions and opinions that were beneficial to the writing of the dissertation. With their help, the thesis was continuously improved and eventually helped me to complete the whole thesis.

In addition, I would like to thank my university and all the teachers who taught me during my college years. It is your careful teaching that enabled me to have good professional knowledge and complete the whole graduation design and thesis.