DEGREE PROJECT IN COMPUTER SCIENCE AND ENGINEERING, SECOND CYCLE, 30 CREDITS

STOCKHOLM, SWEDEN 2020

Machine Learning for Automation of Chromosome based Genetic Diagnostics

GONGCHANG CHU

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Machine Learning for Automation of Chromosome based Genetic Diagnostics

GONGCHANG CHU

Master in Computer Science
Date: November 18, 2020
Supervisor: Shatha Jaradat
Examiner: Amir H. Payberah
School of Electrical Engineering and Computer Science
Host company: Arkus AI AB
Swedish title: Maskininlärning för automatisering av kromosombaserad genetisk diagnostik

Abstract

Chromosome based genetic diagnostics, the detection of specific chromosomes, plays an increasingly important role in medicine as the molecular basis of human disease is defined. The current diagnostic process is performed mainly by karyotyping specialists. They first put chromosomes in pairs and generate an image listing all the chromosome pairs in order. This process is called karyotyping, and the generated image is called karyogram. Then they analyze the images based on the shapes, size, and relationships of different image segments and then make diagnostic decisions. Manual inspection is time-consuming, labor-intensive, and error-prone.

This thesis investigates supervised methods for genetic diagnostics on karyograms. Mainly, the theory targets abnormality detection and gives the confidence of the result in the chromosome domain. This thesis aims to divide chromosome pictures into normal and abnormal categories and give the confidence level. The main contributions of this thesis are (1) an empirical study of chromosome and karyotyping; (2) appropriate data preprocessing; (3) neural networks building by using transfer learning; (4) experiments on different systems and conditions and comparison of them; (5) a right choice for our requirement and a way to improve the model; (6) a method to calculate the confidence level of the result by uncertainty estimation.

Empirical research shows that the karyogram is ordered as a whole, so preprocessing such as rotation and folding is not appropriate. It is more reasonable to choose noise or blur. In the experiment, two neural networks based on VGG16 and InceptionV3 were established using transfer learning and compared their effects under different conditions. We hope to minimize the error of assuming normal cases because we cannot accept that abnormal chromosomes are predicted as normal cases. This thesis describes how to use Monte Carlo Dropout to do uncertainty estimation like a non-Bayesian model[1].

Keywords: Genetic Diagnostics, Abnormality Detection, Transfer Learning, Deep Learning, Uncertainty Estimation

Sammanfattning

Kromosombaserad genetisk diagnostik, detektering av specifika kromosomer, kommer att spela en allt viktigare roll inom medicin eftersom den molekylära grunden för mänsklig sjukdom definieras. Den nuvarande diagnostiska processen utförs huvudsakligen av specialister på karyotypning. De sätter först kromosomer i par och genererar en bild som listar alla kromosompar i ordning. Denna process kallas karyotypning, och den genererade bilden kallas karyogram. Därefter analyserar de bilderna baserat på former, storlek och förhållanden för olika bildsegment och fattar sedan diagnostiska beslut.

Denna avhandling undersöker övervakade metoder för genetisk diagnostik på karyogram. Huvudsakligen riktar teorin sig mot onormal detektion och ger förtroendet för resultatet i kromosomdomänen. Manuell inspektion är tidskrävande, arbetskrävande och felbenägen. Denna uppsats syftar till att dela in kromosombilder i normala och onormala kategorier och ge konfidensnivån. Dess huvudsakliga bidrag är (1) en empirisk studie av kromosom och karyotypning; (2) lämplig förbehandling av data; (3) Neurala nätverk byggs med hjälp av transfer learning; (4) experiment på olika system och förhållanden och jämförelse av dem; (5) ett rätt val för vårt krav och ett sätt att förbättra modellen; (6) en metod för att beräkna resultatets konfidensnivå genom osäkerhetsuppskattning.

Empirisk forskning visar att karyogrammet är ordnat som en helhet, så förbehandling som rotation och vikning är inte lämpligt. Det är rimligare att välja brus, oskärpa etc. I experimentet upprättades två neurala nätverk baserade på VGG16 och InceptionV3 med hjälp av transfer learning och jämförde deras effekter under olika förhållanden. När vi väljer utvärderingsindikatorer, eftersom vi inte kan acceptera att onormala kromosomer bedöms förväntas, hoppas vi att minimera felet att anta som vanligt. Denna avhandling beskriver hur man använder Monte Carlo Dropout för att göra osäkerhetsberäkningar som en icke-Bayesisk modell [1].

Nyckelord: Genetisk diagnos, onormal detektion, överföringsinlärning, djupinlärning, osäkerhetsuppskattning

Contents

Chapter 1

Introduction

The thesis presents methods for data processing and implementation in the automation of chromosome-based genetic diagnostics. In this introductory chapter, I motivate the research question, introduce the study's related work, and put forward a further step for the remainder of the thesis.

1.1 Motivation

Visual search and identification of human chromosomes has become an essential clinical procedure for screening and diagnosing genetic disorders and cancers. In this area, the most common steps to achieve the goal are karyotyping and abnormality detection on the karyograms. Karyotyping is a standard technique utilized to classify metaphase chromosomes into 24 types called karyograms. Because manual genetic diagnostics is a labor-intensive and time-consuming task, developing automatic computer-assisted genetic diagnostics systems has attracted significant research interests in the last 30 years.

While there are numerous methods for automated segmentation and classification of chromosomes, karyotyping analysis remains challenging. The work presented in this thesis is part of a larger project for chromosome-based genetic diagnostics. We believe that there is a value in the abnormality detection of karyograms. The current flow chart of the existing method and what automation system Arkus want to build is as follows (Fig. 1.1):

1.2 Problem

Visual search and identification of analyzable chromosomes are a tedious and time-consuming task that is routinely performed in genetic laboratories to de-

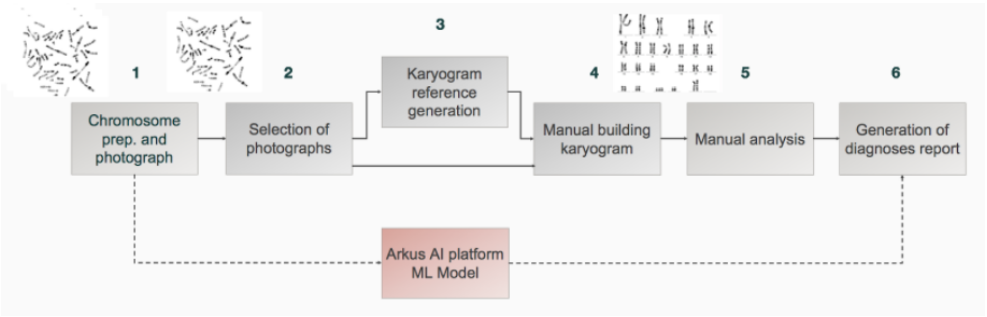


Figure 1.1: Flow Chart of Karyotyping and Abnormality Detection

tect and diagnose cancers and genetic diseases.

This thesis addresses the problem of abnormality detection on karyograms and gives the confidence score of the results. *How can we classify normal and abnormal chromosomes from karyograms and give the confidence score?*

1.3 Goal

The project's objective is to build a prototype of an Artificial Intelligence (AI) powered tool to automate chromosome-based genetic diagnostics. Such a diagnostic process is also called karyotyping. Machine Learning (ML) and image processing techniques are experimented with and developed as the tool's key components.

The input data are the chromosome images after karyotyping processing. There is one label, whether regular or abnormal, for every shot.

The project applies computer vision techniques to classify the input chromosome images into normal cases or various types of abnormal cases.

1.4 Thesis Contributions

The thesis contributions can be summarized as

- **An appropriate pre-processing on the karyogram dataset.** The original dataset is unbalanced and small. In the dataset, about 90% of cases are normal. Thus, it is not easy to train an accurate model to make predictions. We used manual data augmentation to make to dataset balanced and larger. We compared the performance before and after the data preprocessing as well.

- **Building two neural networks and doing the hyper-parameter search.** We built two deep convolutional neural networks by using transfer learning. The basic model we chose VGG-16 and Inception-V3. Before training the models, we were interested in the hyper-parameters, such as learning rate and batch size. We ran a Bayesian optimization for them to get better performance.
- **Evaluating the model performance in different conditions.** We did several experiments on the two neural networks, doing or not doing data augmentation, using or not using pre-trained weights. We compared the performance and chose the best one.
- **Adding Monte-Carlo Dropout into the network and computing the confidence score.** In the application, we gave how confident our predictions. However, the softmax value does not reflect the reliability of the sample classification result. A model can be uncertain in its predictions, even with a high softmax output. We did uncertainty estimation by adding Monte-Carlo Dropout into the network and gave a method to compute the confidence score.

1.5 Ethics and Sustainability

This project supports sustainable development by enabling a chromosome-based abnormality detection system, which benefits both hospitals and laboratories to do genetic diagnostics. Detection of chromosomal abnormalities can be used for prenatal examinations to prevent genetic diseases. More importantly, it can help hospitals and laboratories screen out people with abnormal chromosomes, convenient for diagnosis and treatment.

An ethical prerequisite for processing human chromosome images is that the processing complies with privacy laws and respects users' integrity. The data used to produce results presented in this thesis are non-confidential and processed solely for scientific purposes. The data collection was external to my work and is out of the scope of this thesis.

1.6 Research Methodology

The research methodology consists of quantitative evaluations and comparisons with different solutions. As no established theory exists on the topic of

our work, the conducted research is inductive, intending to provide new approaches and solutions within our field of study.

1.7 Outline

The remainder of this thesis is structured as follows. Chapter 2 introduces the background and related works of this thesis. Chapter 3 describes the methodology and implementation of the automation system and neural networks. Chapter 4 summarizes the performance of each network and makes the comparison. Our group discussions and conclusions are shown in Chapter 5. In the last, Chapter 6 is the future work of this project.

Chapter 2

Background

Nowadays the most popular methodology of chromosome-based genetic diagnosis is karyotyping. A chromosome karyotype is used to detect chromosome abnormalities, diagnose genetic diseases, congenital disabilities, and specific blood or lymphatic disorders.

2.1 Karyotype

Karyotyping is the process by which photographs of chromosomes are taken to determine the chromosome complement of an individual, including the number of chromosomes and any abnormalities. The term is also used for the complete set of chromosomes in a species or an individual organism and for a test that detects this complement or measures the number. It will get a result called karyogram (Fig. 2.1).

Karyograms describe the chromosome count of an organism and what these chromosomes look like under a light microscope. Attention is paid to their length, the centromeres' position, banding pattern, differences between the sex chromosomes, and any other physical characteristics. There are some typical chromosomal abnormalities, such as Down syndrome and Edward's syndrome. In the karyotype analysis, patients with Down syndrome have three chromosomes in the 23rd pair.

2.2 Chest X-ray Abnormality Detection

There are some automatic abnormality detection cases in chest x-rays without medical training using Deep Convolutional Neural Networks. Chest x-rays are widely used for diagnosis in the heart and lung area[2].

Figure 2.1: Human male karyogram

Bar et al.[3] examined the strength of deep learning approaches for pathology detection in chest radiograph data. Most of the research in computer-aided detection and diagnosis in chest radiography has focused on lung nodule detection. Nevertheless, lung nodules are a relatively rare finding in the lungs. The most common findings in chest X-rays include lung abnormalities of the size, infiltrates, catheters, or contour of the heart[4]. Aiming to distinguish the various chest diagnosis, they chose to use deep neural networks, which have recently gained considerable interest due to the development of new variants of CNNs.

They used a CNN that was trained with Image-Net, a well-known large scale non-medical image database. The best performance was achieved using features extracted from the CNN and a set of low-level features. They obtained an area under curve AUC of 0.93 for Right Pleural Effusion detection.

Islam et al.[5] explored deep convolutional network (DCN) based abnormality detection in frontal chest x-rays. They believed that automatically detecting these abnormalities with high accuracy could greatly enhance real-world diagnosis processes. Many research groups have focused on developing computer-aided detection (CAD) tools for X-Ray analysis in the past decade. However, the accuracy of these CAD tools cannot achieve a significantly high level. Thus they aimed to improve the abnormality detection and localization in chest X-Rays.

They studied the performance of various DCN architectures on different abnormalities. They found that the same DCN architecture doesn't perform well across all anomalies. They also found ensemble models to improve classification significantly compared to a single model when only DCN models are used. The summary of the classification is shown in Figure 2.2. Their contributions showed a 17 percentage point improvement in accuracy over the rule-based method for Cardiomegaly detection using deep convolutional networks. They multiplied random train/test data split achieve robust accuracy results when the number of training examples is low. It has a vital reference significance because our chromosome data set is also relatively small. They reported the highest accuracy on several chest x-ray abnormality detection, where comparison could be made.

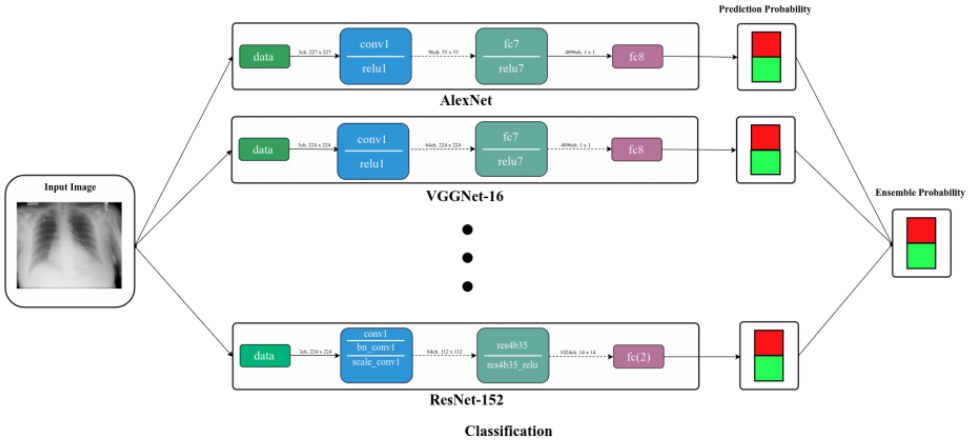


Figure 2.2: Summary of the classification method[5]

2.3 Chromosome classification

Chromosome classification is critical for karyotyping in abnormality diagnosis. Classification is generally done to extract the information classes. Nearest neighbor, decision tree, boosted tree, support vector machine, random forest, linear, neural network are various image classification methods used. Most important among them was the neural network because they process different records at different record times and learn to compare the classification of documents with the actual records.[6] The chromosome classification is done by these procedures:

- The first step is preprocessing, dealing with noise using a bilateral filter and unbalanced datasets such as oversampling using normalization or argumentation. Dimension reduction is made as well.
- The second step is segmentation, where the image gets segmented. Segmentation is done to get a detailed study of the picture.
- The third step is feature extraction, where the features are extracted individually which helps to analyze the image with accuracy, sensitivity, dimension, etc.
- The fourth step is the classification. The images get classified with Artificial Neural Network. With the help of a predefined dataset, the diseases are identified.

There are many mature networks that can classify chromosomes, such as faster-RCNN, Varifocal-Net and so on.

Ding et al. proposed a preprocessing model with object segmentation and feature enhancement[7]. It is essential for ensuring the healthy survival rate of newborns that prenatal screening of abnormalities. However, chromosome karyotype image analysis's complex information and tedious workload is a significant difficulty in medical diagnosis.

Their contributions can be summarized as the preprocessing model and deep learning network architecture. The study of chromosome karyotype images pixel-level segmentation and enhancement of chromosome band features in the preprocessing model are useful for our project. They also developed an automatic classification network for chromosome karyotype images. The result of the system could be the input of ours.

Qin et al. presented a novel method named Varifocal-Net for simultaneous classification of chromosomes type and polarity using deep convolutional networks[8]. Chromosome classification is essential for karyotyping in abnormality diagnosis. It has a deep meaning to expedite the diagnosis.

Their contributions are (1) Inspired by the zoom capability of cameras. (2) They utilized the concatenated features from both global and local scales to predict the type and polar simultaneously. (3) They evaluated the proposed approach on a large dataset. (4) The Varifocal-Net had been put into clinical practice for chromosome classification.

2.4 Uncertainty Estimation

Deep learning tools have gained significant attention in applied machine learning. However, such means for classification and regression do not capture model uncertainty. In comparison, Bayesian models provide a mathematically grounded framework to measure model uncertainty but usually come with a prohibitive computational cost[9]. There are two main non-Bayesian methods for uncertainty estimation: *Monte Carlo Dropout* (MC Dropout)[1] and *Deep Ensembles*[10].

Gal et al.[1] developed a new theoretical framework casting dropout training in a deep neural network as approximate Bayesian inference in deep Gaussian processes. They implemented dropout as a Bayesian approximation, so-called Monte Carlo Dropout.

In their experiments, a direct result gave them tools to model uncertainty with dropout neural networks - extracting information from existing models that have been thrown away so far. This mitigated the problem of measuring tension in deep learning without sacrificing either computational complexity or test accuracy. They performed a comprehensive study of the properties of dropout uncertainty.

Bernoulli dropout is only one example of a regularisation technique in their theory. Other variants of dropout would result in different uncertainty estimations, trading-off uncertainty quality with computational complexity.

Lakshminarayanan et al. proposed an alternative to Bayesian Deep neural networks (NNs) that is simple to implement, readily parallelizable, requires minimal hyperparameter tuning, and yields high-quality predictive uncertainty estimates[10]. Deep neural networks are robust. However, quantifying predictive uncertainty in them is a challenging and yet unsolved problem.

Their contributions in that paper can be summarized as two-fold. First, they described a scalable and straightforward method for uncertainty estimation from NNs. Second, they proposed a series of tasks for evaluating the quality of predictive uncertainty estimates.

In comparison, MC-dropout is relatively simple to implement, leading to its popularity in practice. The ensemble interpretation seems more plausible, specifically in the scenario where the dropout rates are not tuned based on the training data.

Chapter 3

Methodologies and Implementation

3.1 Data Processing

Overall, this system's input is a picture of human cell chromosomes, and the output is a prediction of whether the patient's chromosomes are abnormal. So we need to perform specific processing on the photo so that we can make accurate predictions.

3.1.1 Original Data

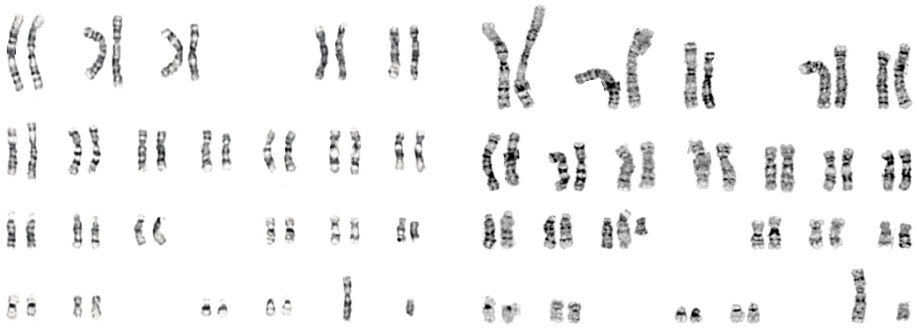
The original data in our project is chromosome photos of human cells (Figure 3.1). Such a cell map is difficult to detect abnormalities because of the folding and covering of chromosomes. Therefore, this project is based on the existing chromosome classification technology to preprocess cell chromosome photos into ordered karyograms (Fig. 3.2). The detailed division process is not in this thesis and will not be described in detail.

3.1.2 Data Augmentation

We got 75 karyograms of standard cases and nine karyograms of abnormal points at the beginning of the thesis. As we all know, the proportion of people with chromosomal abnormalities is not high. It is an unbalanced data set, with anomalies accounting for about 10%.

Because of the COVID-19, we have to search for karyogram data from the Internet. The data set reached 105 regular cases and 41 abnormal cases. It

Figure 3.1: Chromosome Photos of Human Cells



(a) Normal Male (b) Abnormal Male

Figure 3.2: Karyotypes of a Normal Male and an Abnormal Male

solves the imbalance to a certain extent. It is called *alpha-dataset*. Examples of normal and abnormal circumstances are shown in Figure 3.2.

To solve imbalance and too few data sets, we once again made data augmentation and enhancements to this data set manually. Because karyograms are ordered and arranged positions, we cannot add flips or rotations to the images. So we first resized the images into 416×416 pixels in the preprocessing step. Then we added brightness between -25% to +25% and noise up to 5% of pixels in the augmentation step. We got 376 typical images and 316 abnormal images after the data augmentation. These mostly solve the data imbalance.

3.2 Hyperparameter Search

We ran a Bayesian optimization for two different hyper-parameters, learning rate, and batch size. This section presents the results alongside the conclusions to which setup will be used in future experiments. Note that all failed experiments are filtered out.

We have the parameter of interest (learning rate, batch size) on the x-axis, validation loss on the y-axis, and validation accuracy on the color axis for the plots below.

From the first plot (Fig. 3.3) below, it is observed that there is a negative correlation between the learning rate and the validation loss. Learning rates in the range [7e-4,3e-2] seems to achieve validation loss.

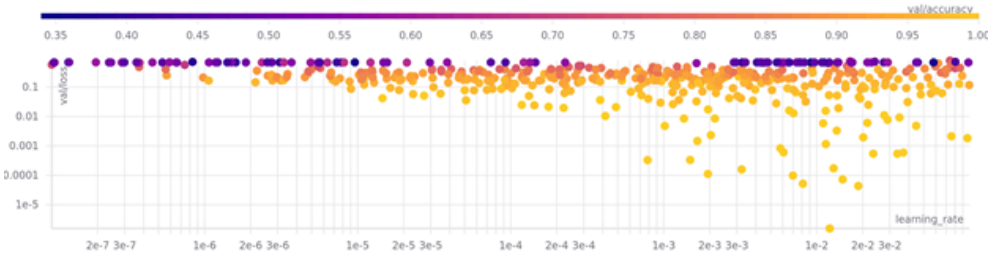


Figure 3.3: Validation Loss and Validation Accuracy vs. Learning Rate

The second plot (Fig. 3.4) demonstrates that we should use a batch size of 16 or 32. Do not use a batch size of 512 or higher. There are many cases where we run out of GPU memory. Note that the same could be right for even smaller batch sizes.

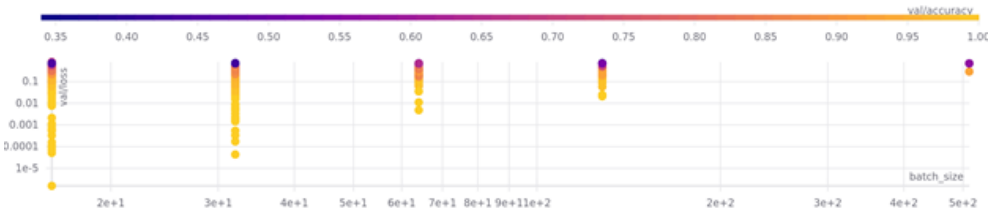


Figure 3.4: Validation Loss and Validation Accuracy vs. Batch Size

We chose 1e-3 as the learning rate and 16 as the batch size in the following experiments.

3.3 Model Architecture

3.3.1 Transfer Learning

Because of the small data set, we chose to use transfer learning to build the network. We chose the well-known neural networks *VGG 16*[11] and *Inception V3*[12] in image recognition as the original network.

Transfer learning is a technique of machine learning. Its purpose is to apply it to new and relevant tasks under the premise of obtaining specific additional data or an existing model[13]. We can divide the data for transfer learning into two categories: source data and target data. Source data refers to other data and is not directly related to the task to be solved, while target data is directly related to the study. In a typical transfer learning, the source data is often massive. The target data is usually relatively small. For example, there are many different people's speech data in speech recognition tasks, but we only want to recognize a specific speech in practical applications accurately. Unlike the voice data of other people, the voice data of a particular person is insignificant. How to make fair use of source data to help improve the model's performance on target data is a problem to be considered in migration learning.[14]

3.3.2 VGG16 Based Model

VGG16 is a convolution neural net (CNN) architecture won ILSVR (Imagenet) competition in 2014[15]. It is considered to be one of the excellent vision model architecture to date. The unique thing about VGG16 is that instead of having a large number of hyper-parameter, they focused on having convolution layers of 3×3 filters with a stride 1 and always used the same padding and max pool layer of 2×2 filters of stride 2.

We removed the fully connected layers of VGG16 and wrote top layers suitable for chromosome binary classification to apply transfer learning. We added two dropout layers, one next to the basic model and another next to the fully connected layer. There are a global average pooling layer and a batch normalization layer, which can deal with overfitting and other problems. The summary of the model is shown in Figure 3.5 and the visualization is shown in Figure 3.7.

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 150, 150, 3)]	0
block1_conv1 (Conv2D)	(None, 150, 150, 64)	1792
block1_conv2 (Conv2D)	(None, 150, 150, 64)	36928
block1_pool (MaxPooling2D)	(None, 75, 75, 64)	0
block2_conv1 (Conv2D)	(None, 75, 75, 128)	73856
block2_conv2 (Conv2D)	(None, 75, 75, 128)	147584
block2_pool (MaxPooling2D)	(None, 37, 37, 128)	0
block3_conv1 (Conv2D)	(None, 37, 37, 256)	295168
block3_conv2 (Conv2D)	(None, 37, 37, 256)	590080
block3_conv3 (Conv2D)	(None, 37, 37, 256)	590080
block3_pool (MaxPooling2D)	(None, 18, 18, 256)	0
block4_conv1 (Conv2D)	(None, 18, 18, 512)	1180160
block4_conv2 (Conv2D)	(None, 18, 18, 512)	2359808
block4_conv3 (Conv2D)	(None, 18, 18, 512)	2359808
block4_pool (MaxPooling2D)	(None, 9, 9, 512)	0
block5_conv1 (Conv2D)	(None, 9, 9, 512)	2359808
block5_conv2 (Conv2D)	(None, 9, 9, 512)	2359808
block5_conv3 (Conv2D)	(None, 9, 9, 512)	2359808
block5_pool (MaxPooling2D)	(None, 4, 4, 512)	0
dropout (Dropout)	(None, 4, 4, 512)	0
global_average_pooling2d (Gl	(None, 512)	0
dense (Dense)	(None, 256)	131328
batch_normalization (BatchNo	(None, 256)	1024
dropout_1 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 1)	257

Total params: 14,847,297
Trainable params: 14,846,785
Non-trainable params: 512

Figure 3.5: Summary of VGG16 Based Network

dropout (Dropout)	(None, 3, 3, 2048)	0	mixed10[0][0]
global_average_pooling2d (Globa	(None, 2048)	0	dropout[0][0]
dense (Dense)	(None, 256)	524544	global_average_pooling2d[0][0]
batch_normalization_94 (BatchNo	(None, 256)	1024	dense[0][0]
dropout_1 (Dropout)	(None, 256)	0	batch_normalization_94[0][0]
dense_1 (Dense)	(None, 1)	257	dropout_1[0][0]

Total params: 22,328,609
Trainable params: 22,293,665
Non-trainable params: 34,944

Figure 3.6: Summary of Top Layers in InceptionV3 Based Network

3.3.3 InceptionV3 Based Model

Inception V3 by Google is the third version in a series of Deep Learning Convolutional Architectures. Inception V3 was trained using a dataset of 1,000 classes from the original ImageNet data set, which was trained with over 1 million training images. Inception V3 was trained for the ImageNet Large Visual Recognition Challenge, where it was the first runner.

Considering that the convolutional layers of VGG16 and InceptionV3 are the same effect, they are both feature extraction. To compare its effect and future scalability, we used the same fully connected layer as the VGG16 based model for the InceptionV3 based model.The Summary of Top Layers in InceptionV3 based model is shown in Figure 3.6 and the visualization is shown in Figure 3.7.

3.4 Confidence Score

At the application level, the performance of the network cannot be easily measured as accuracy. So we introduce a new indicator: confidence score, which can show how confident our prediction. A confidence score is an excellent way to measure uncertainty. In the thesis, our network is a non-Bayesian network. There are two main non-Bayesian methods for uncertainty estimation: *Monte Carlo Dropout* (MC Dropout)[1] and *Deep Ensembles*[16].

The dropout method is one of the most popular approaches to prevent overfitting. Gal et al.[1] demonstrate that we could obtain the model uncertainty when sampling from the Bernoulli distribution with probability like the dropout probability. With MC Dropout, the dropout layer is applied in training process and testing process and then making multiple predictions on one

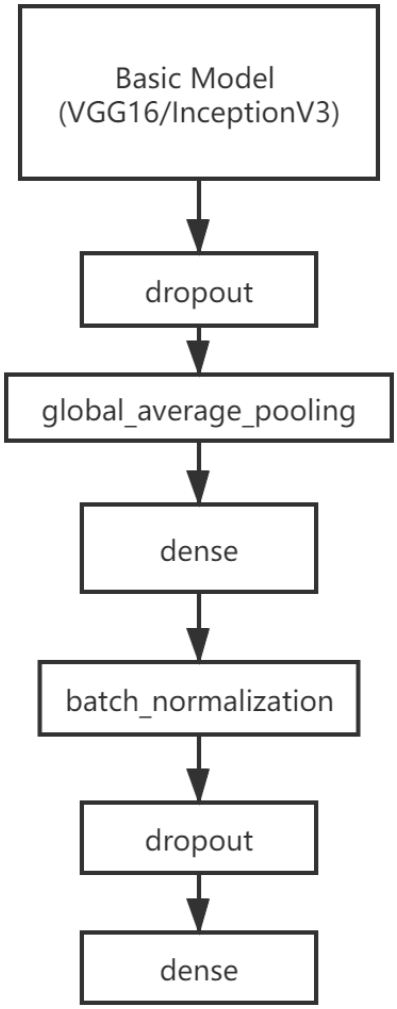


Figure 3.7: The Structure of Model by Transfer Learning

image to measure the uncertainty.

In this project, we chose to use MC Dropout. It costs less computational resources and has fewer hyper-parameters to tune.

In Figure 3.7, the penultimate layer is the dropout layer. Under normal circumstances, during the training process, the dropout layer is enabled to avoid overfitting. In the prediction process, dropout will be automatically closed so that the same picture's prediction result is the same every time. In MC Dropout, we need to turn on the dropout layer in the prediction process so that each prediction's softmax value fluctuates, which affects its classification. In the next step, we make 100 predictions for each target picture and take most of the prediction results as the final prediction classification. The number of predictions will become the percentage of confidence score. If the confidence score value is less than a certain threshold, such as 80%, that is to say, in 100 predictions, none of the positive and negative predictions is greater than 80 times. We regard this situation as impossible to predict and require manual processing accurately.

Chapter 4

Results and Discussions

4.1 Experiments without Data Augmentation

In the first experiment, we use *alpha-dataset* as the karyograms' data set, which has 105 standard cases and 41 abnormal cases. We started with an imbalanced data set. We split the data set into a training set, validation set, and testing set randomly. Their ratio is 64%, 16%, and 20%. So we got 92 training images, 24 validation images, and 30 testing images.

4.1.1 VGG 16

In this experiment, we chose VGG 16 as the base model and used its pre-trained weight. We removed the fully connected layers and implemented one which fits the topic.

We set the training metrics: loss, precision, recall, and area under the curve (AUC). We chose validation AUC as the monitor of the training process. The figure of each metrics in the training process is shown below (Fig. 4.1).

The X-axis is epochs in these charts, and Y-axis is metrics: loss, AUC, precision, and recall. We can see that the precision and recall are not so good on the validation data set. Also, the AUC is low on the validation set. An unbalanced and small data set most likely causes this.

Then we predicted the test set using the trained model. A confusion matrix summarizes the actual vs. predicted labels where the X-axis is the predicted label, and the Y-axis is the actual label. In this thesis, we set abnormal cases as positive, while normal cases as unfavorable. The confusion matrix of this experiment is drawn below (Fig. 4.2).

If the model had predicted everything correctly, this would be a diagonal

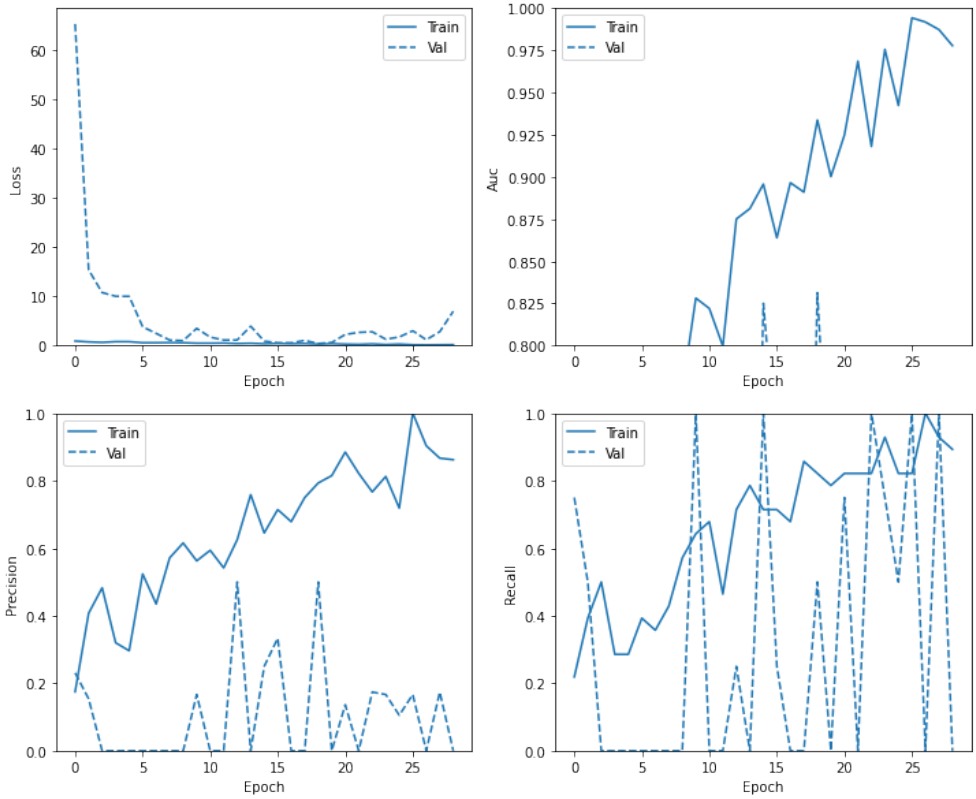


Figure 4.1: Training Metrics of VGG 16 Based Model with Pre-trained Weight on Original Dataset

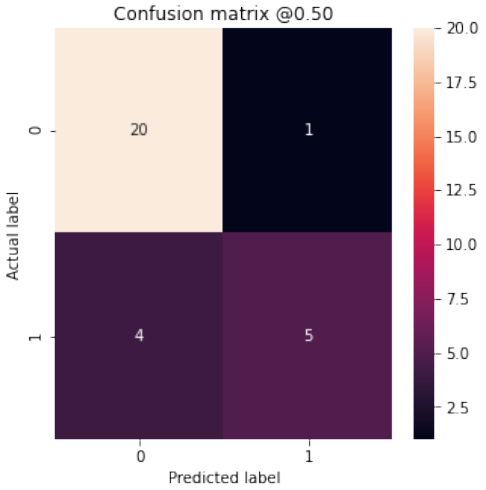


Figure 4.2: The Confusion Matrix of VGG 16 Based Model with Pre-trained Weight on Original Dataset

matrix where values off the main diagonal, indicating incorrect predictions, would be zero. It can be seen from this figure: there are five true-positive cases, one false-positive case, 20 true-negative cases, and four false-negative cases. According to the formula of precision and recall, it can be calculated that

- Precision = 0.83
- Recall = 0.56

We also got the test accuracy is 0.83 and *auc* is 0.79.

The low recall indicates that there are many abnormal cases classified as usual by incorrect predictions.

Finally, we plot the ROC curve (Fig. 4.3). This plot is useful because it shows, at a glance, the range of performance the model can reach just by tuning the output threshold.

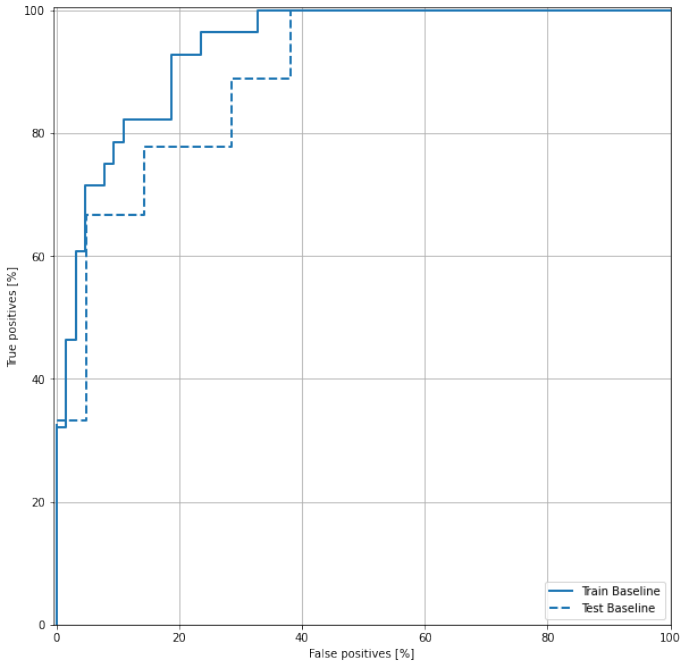


Figure 4.3: The ROC Curve of VGG 16 Based Model with Pre-trained Weight on Original Dataset

It looks like the precision is relatively high, but the recall and the area under the ROC curve (AUC) are not as high as expected. Classifiers often face challenges when maximizing precision and recall, which is especially true when working with imbalanced datasets.

4.1.2 Inception V3

In this experiment, we chose Inception V3 as the base model and used its pre-trained weight in ImageNet. We removed the top layers and put the same top layers on it as the previous model.

The training metrics: loss, precision, recall, and AUC are shown in the figure (Fig. 4.4). In these charts, the X-axis is epochs, and Y-axis is metrics. It is shown that the precision is high while the recall is low. Besides, the AUC is not acceptable in the validation data set.

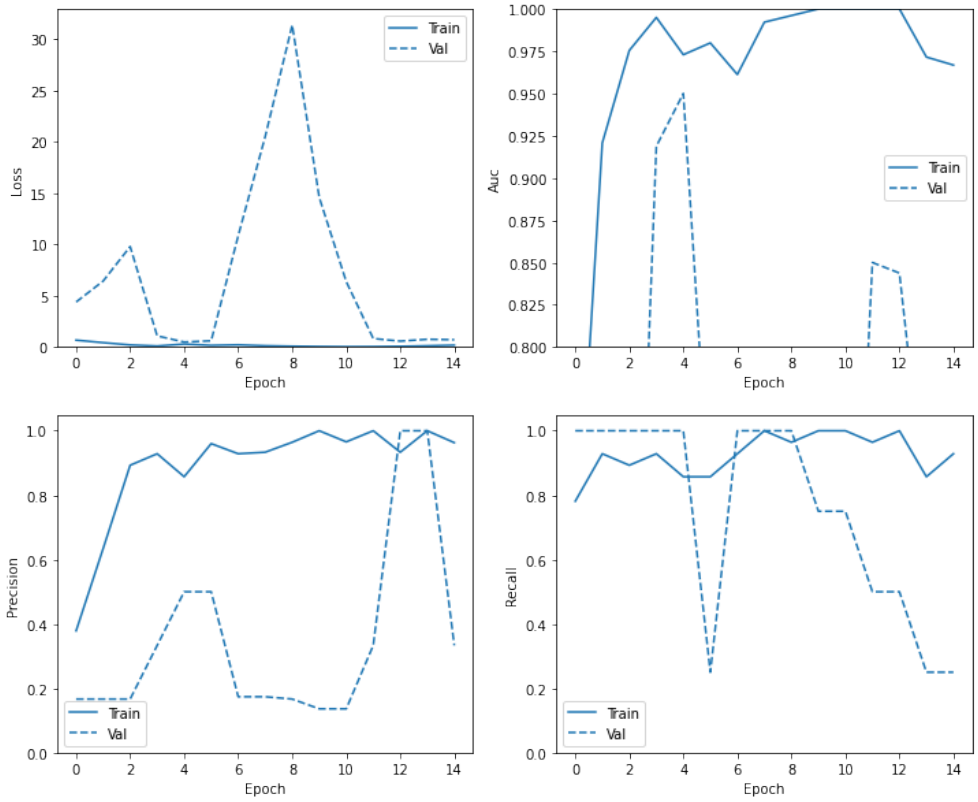


Figure 4.4: Training Metrics of Inception V3 Based Model with Pre-trained Weight on Original Dataset

In the testing process, the prediction's confusion matrix is the plot (Fig. 4.5). It can be seen from this figure: there are seven true-positive cases, five false-positive cases, 16 true-negative cases, and two false-negative cases. According to the formula of precision and recall, it can be calculated that

- Precision = 0.58

- Recall = 0.78

We also got the test accuracy is 0.77 and *auc* is 0.86.

It has a low precision while a high recall. There are lots of standard cases filtered into abnormal cases by incorrect predictions.

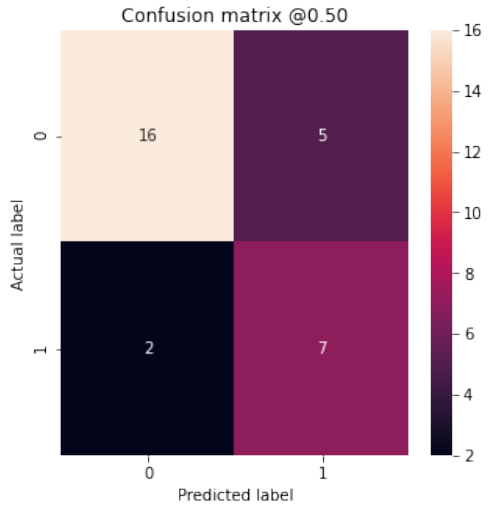


Figure 4.5: The Confusion Matrix of Inception V3 Based Model with Pre-trained Weight on Original Dataset

At last, we plot the ROC curve (Fig. 4.6). In this figure, it is evident that the recall is high, and the precision is low. Although the balancing of precision and recall is not good enough, the area under the curve (AUC) is more extensive than VGG16 based model.

4.1.3 Discussion

In conclusion, in the original data set, the InceptionV3 based model performs better than the VGG16 based model. However, they both have disadvantages in precision or recall. It could be improved by doing the data augmentation.

4.2 Experiments with Data Augmentation

In the previous section, we found the two models do not perform well enough on the original data. Then, we did data augmentation manually and got 376 typical cases and 316 abnormal cases. It is said to be a balanced data set. We split the data set into a training set, validation set, and testing set randomly.

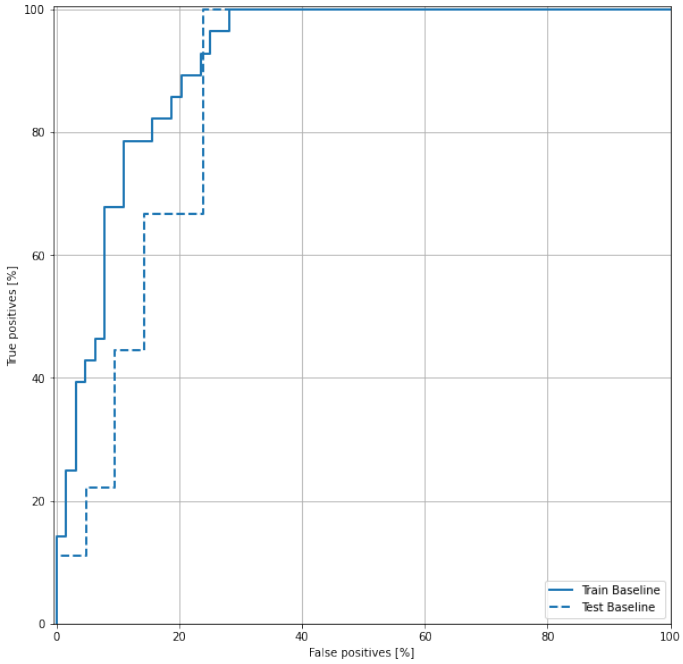


Figure 4.6: The ROC Curve of Inception V3 Based Model with Pre-trained Weight on Original Dataset

Their ratio is 64%, 16%, and 20%. So we got 442 training images, 111 validation images, and 139 testing images.

4.2.1 VGG 16

In this section, we chose VGG-16 as the base model and use its pre-trained weight. We removed the fully-connected layers and replaced them with our top layers.

The training metrics: loss, precision, recall, and AUC are shown in the figure (Fig. 4.7). In these charts, the X-axis is epochs, and Y-axis is metrics. It is shown that the precision is a little bit higher than the recall; however, it has the right balance on them. The high AUC represents this. Also, the recall and AUC curves fluctuate relatively large. It may not be good enough in normal cases predicting.

In the testing process, the prediction's confusion matrix is the plot (Fig. 4.8). It can be seen from this figure: there are 56 true-positive cases, one false-positive case, 56 true-negative cases, and three false-negative cases. According to the formula of precision and recall, it can be calculated that

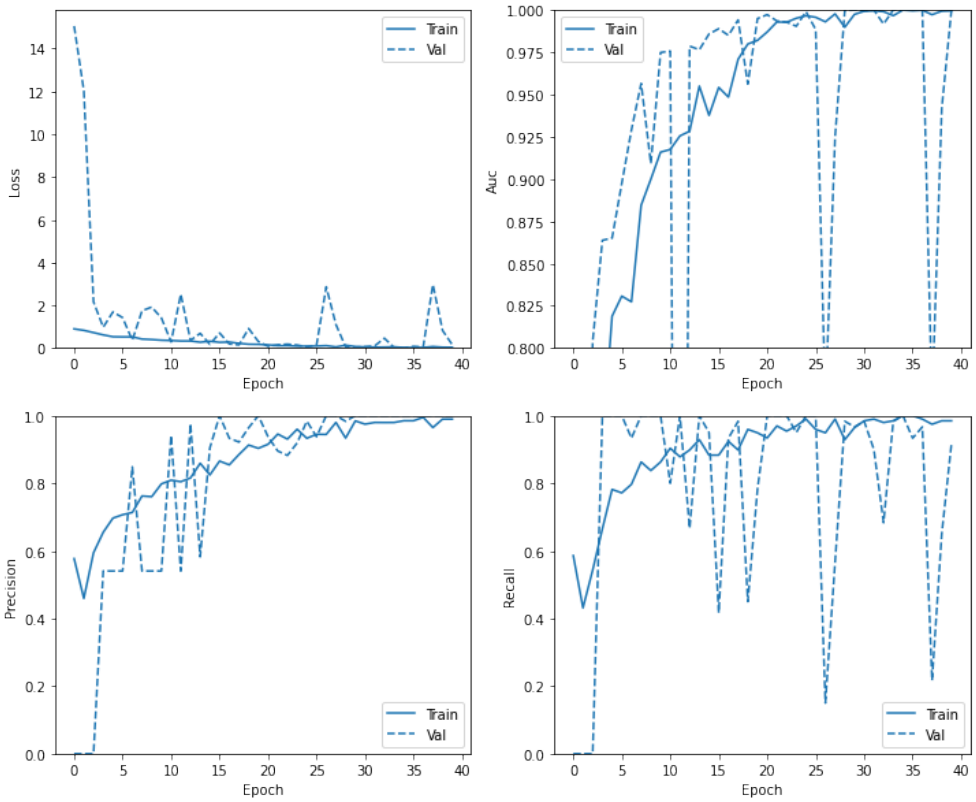


Figure 4.7: Training Metrics of VGG-16 Based Model with Pre-trained Weight on Augmented Dataset

- Precision = 0.98
- Recall = 0.95

We also got the test accuracy is 0.97, and *auc* is 0.99. The precision is a little bit larger than the recall. So this model performs better on abnormal cases.

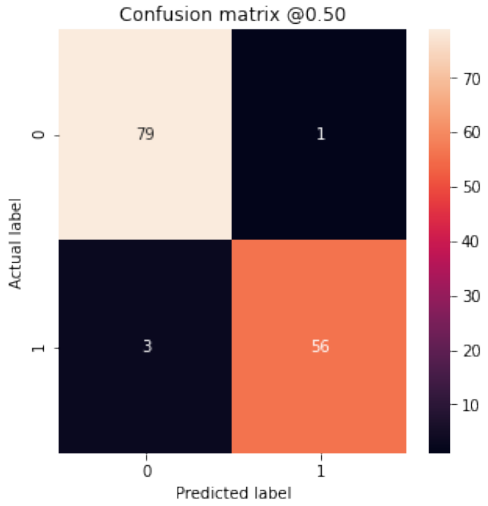


Figure 4.8: The Confusion Matrix of VGG-16 Based Model with Pre-trained Weight on Augmented Dataset

At last, we plot the ROC curve (Fig. 4.9). In this figure, it is evident that the performance of the model is outstanding. The AUC of the training set is almost one, while the AUC of the test set is also larger than 0.95.

On the other hand, data augmentation significantly makes the model perform better. Whether it is precision, recall, or AUC, the model trained with the processed data is much better than the model trained with the original data.

4.2.2 Inception V3

In this section, we use the augmented dataset as the training, validation, and testing data. Input them into the Inception V3 based model and use its pre-trained weight and our top layers.

The training metrics: loss, precision, recall, and AUC are shown in the figure (Fig. 4.10). In these charts, the X-axis is epochs, and Y-axis are different metrics. It is shown that the AUC and recall are relatively bad in 0-15 epochs. Moreover, precision performs better than recall. It means that it performs better on normal cases of the validation set.

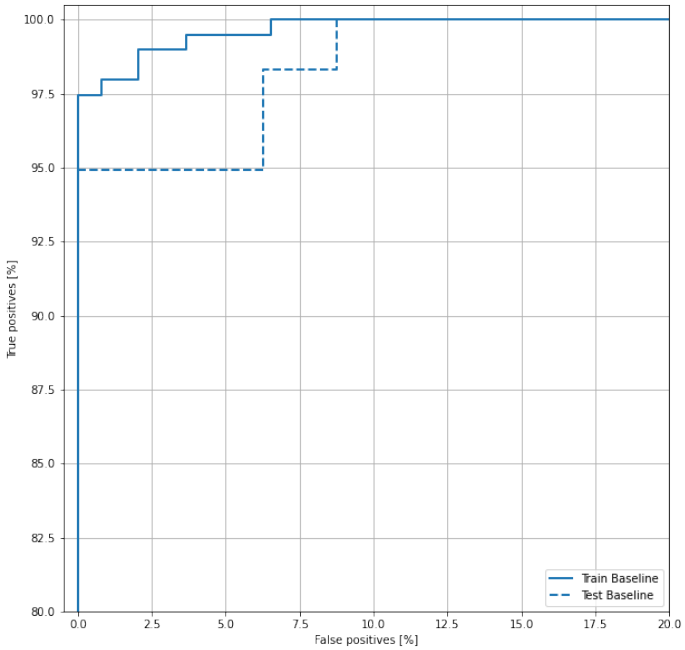


Figure 4.9: The ROC Curve of VGG-16 Based Model with Pre-trained Weight on Augmented Dataset

The confusion matrix of the prediction result is the plot in the figure (Fig. 4.11). The figure shows 51 true-positive predictions, one false-negative prediction, 79 true-negative predictions, and eight false-negative predictions. It can be calculated that

- Precision = 0.98
- Recall = 0.86

We got the test accuracy is 0.94, and *AUC* is 0.99. The precision is a little bit larger than the recall. So this model performs better on abnormal cases.

In the end, we plot the ROC curve (Fig. 4.12). In this figure, it is obvious that the performance of the model is excellent. The AUC of the training set is higher than 0.98, and The AUC of the testing set is higher than 0.95.

4.2.3 Discussion

In this section, we used the augmented dataset to train the two models that are the same as the previous section.

Compared with the small unbalanced data set, the model's effect based on VGG16 or InceptionV3 has made significant progress. For the VGG16-based

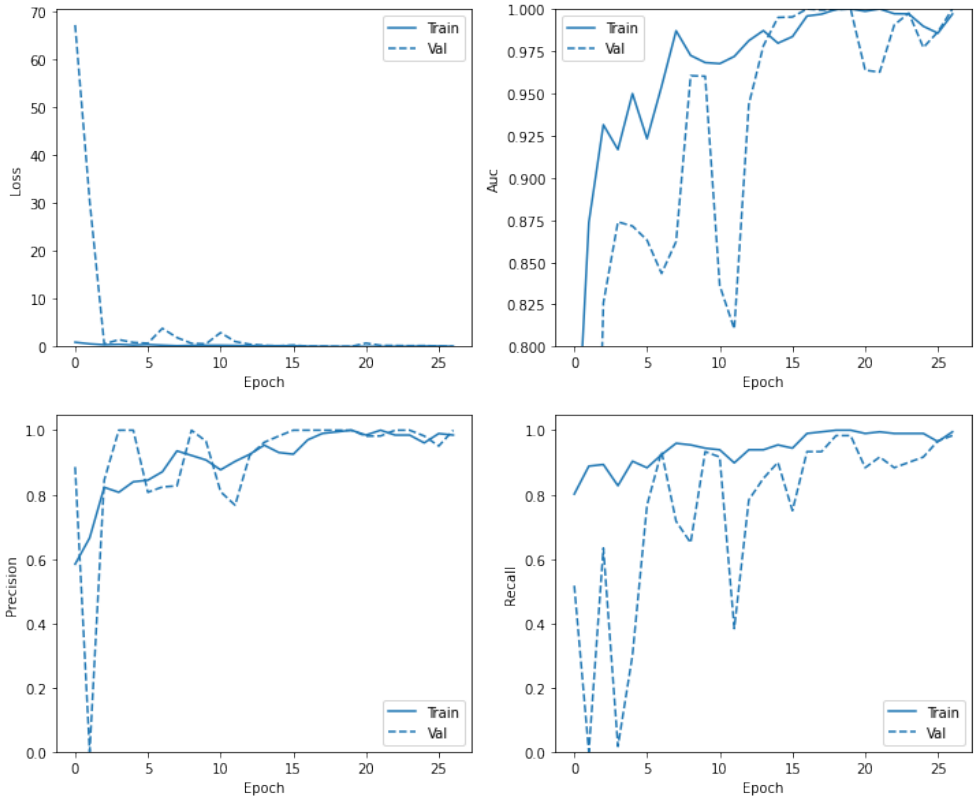


Figure 4.10: Training Metrics of Inception V3 Based Model with Pre-trained Weight on Augmented Dataset

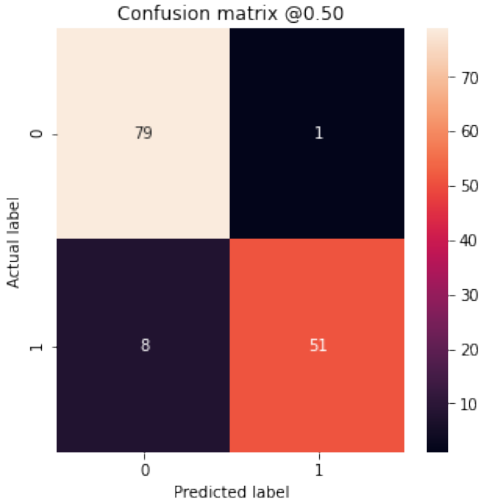


Figure 4.11: The Confusion Matrix of Inception V3 Based Model with Pre-trained Weight on Augmented Dataset

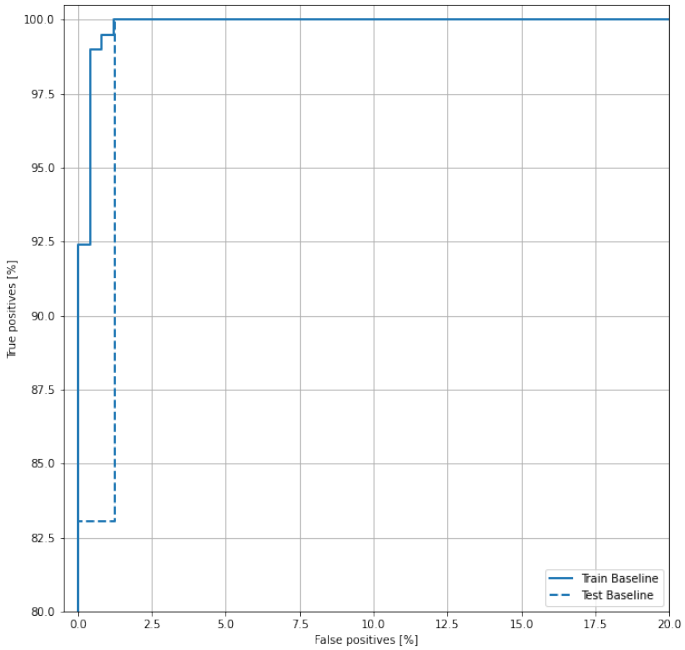


Figure 4.12: The ROC Curve of Inception V3 Based Model with Pre-trained Weight on Augmented Dataset

model, precision increased from 0.83 to 0.98, and recall increased from 0.56 to 0.95. AUC increased from 0.79 to above 0.95. For the InceptionV3-based model, precision has increased from 0.58 to 0.98, and recall has increased from 0.78 to 0.86. At the same time, AUC has increased from 0.86 to more than 0.95.

On the other hand, when comparing these two models, their performance on the data set is more precise than recall, which means that the two models are better than abnormal cases in predicting normal cases. The VGG16-based model predicts abnormal cases significantly better than the InceptionV3-based model.

4.3 Experiments without Pre-trained Weights

To improve the prediction accuracy for chromosome abnormalities, we decided to use the original model's pre-training weight as the initial value and train the two models as a whole, including the weight of the basic model.

In this section, we still use the better-performing processed data set, train the layers other than top layers as a whole or partly, observe how the two mod-

els perform and compare them.

4.3.1 VGG 16

The structure of VGG16 is a loop block (Fig. 4.13). In this section, we trained one or two blocks that are closest to the top layers. Looked for the rules and the possibility of improving model performance.

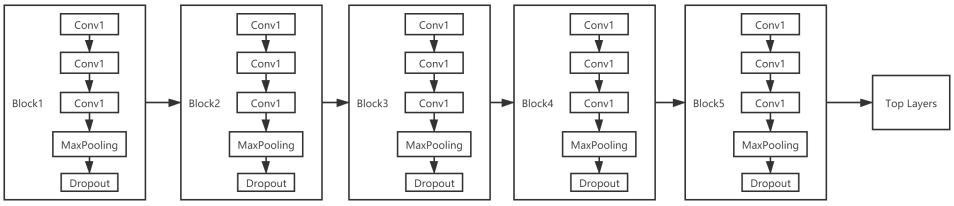


Figure 4.13: The model that built based on VGG-16 model

4.3.2 Setting the Last Block as Trainable

In this section, we unfreeze the last block in the base model. Then use the processed dataset to train and test.

The training metrics: loss, precision, recall, and AUC are shown in the figure (Fig. 4.14). In these charts, we can find that the loss is shallow. Then, the accuracy fluctuates somewhat and is low. Also, the recall rate and AUC were lower in the first ten periods, but both reached a higher level afterward.

Then, we used the same test set as before to make predictions. The generated confusion matrix is shown in Figure 4.15. As can be seen from the figure, there are 57 true-positive cases, one false-positive case, 79 true-negative cases, and two false-negative cases. It is easy to get that

- Precision = 0.98
- Recall = 0.96

The value of recall is a little bit larger than what we got in 4.2.1.

Finally, the ROC curve is plotted in the Figure 4.16. We put the original ROC curve and the ROC curve after training a block in a figure. The blue line represents the original ROC curve, and the orange line represents the ROC curve after training a block. It can be found that the new ROC curve as a whole is to the right of the original ROC curve, which means that the new model is better than the original model in predicting abnormal cases. It is also in line with the broader conclusion of recall in the confusion matrix.

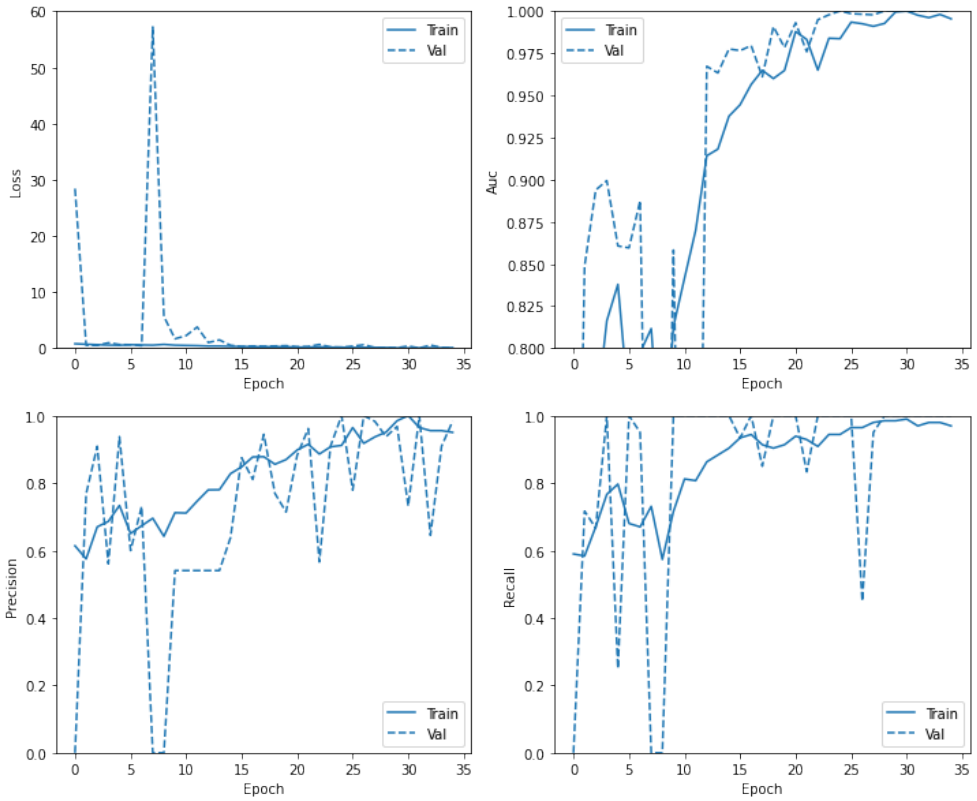


Figure 4.14: Training Metrics of VGG16 Based Model with One Block Layers Trainable on Augmented Dataset

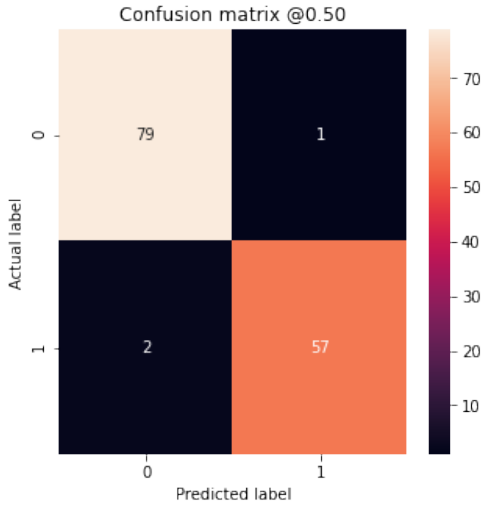


Figure 4.15: The Confusion Matrix of VGG16 Based Model with One Block Layers Trainable on Augmented Dataset

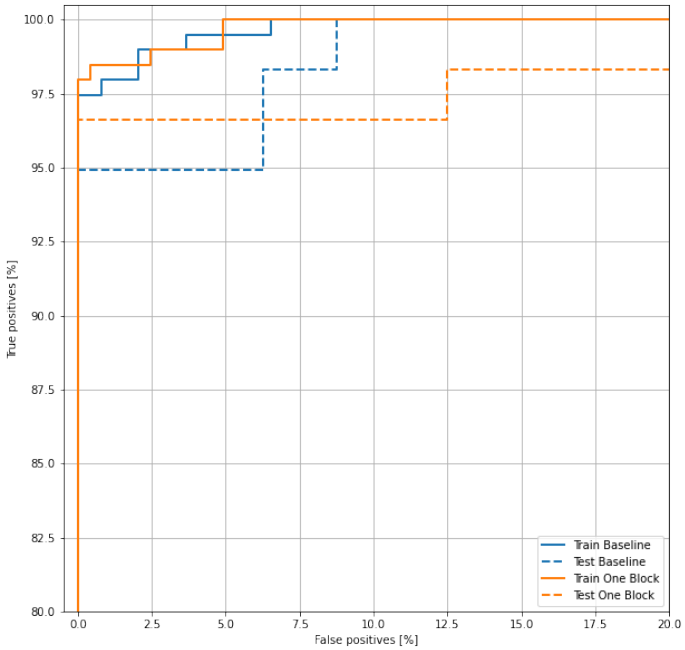


Figure 4.16: The ROC Curve of VGG16 Based Model with One Block Layers Trainable on Augmented Dataset

4.3.3 Setting the Last Two Blocks as Trainable

Because training the last block has improved the recall, we decided to train one more block. We set the last two blocks to be trainable, used the same data set to train and test them, and got the following results.

The training metrics: loss, precision, recall, and AUC are shown in Figure 4.17. The loss of training and validation sets are shallow. The AUC curve fluctuated several times in the range of 20-25 epochs and finally stabilized. The Precision and recall curves have slightly larger fluctuations compared to the previous ones.

Then, we used the same test set as before to make predictions. The generated confusion matrix is shown in Figure 4.18. As can be seen from the figure, there are 57 true-positive cases, four false-positive cases, 76 true-negative cases, and two false-negative cases.

- Precision = 0.93
- Recall = 0.96

The value of precision is slightly lower than the previous one, while recall has not changed. This is because the model incorrectly predicted three normal

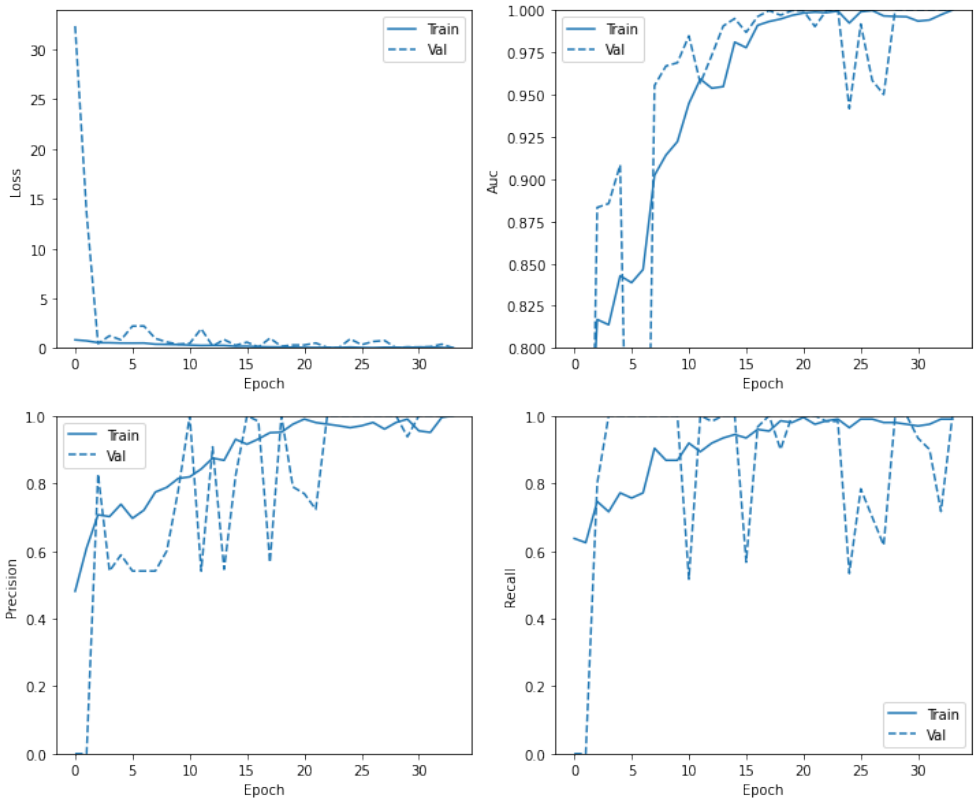


Figure 4.17: Training Metrics of VGG16 Based Model with Two Block Layers Trainable on Augmented Dataset

cases into abnormal categories. Although this reduces the accuracy rate, it may reduce problems such as overfitting. The specific performance depends on the ROC curve.

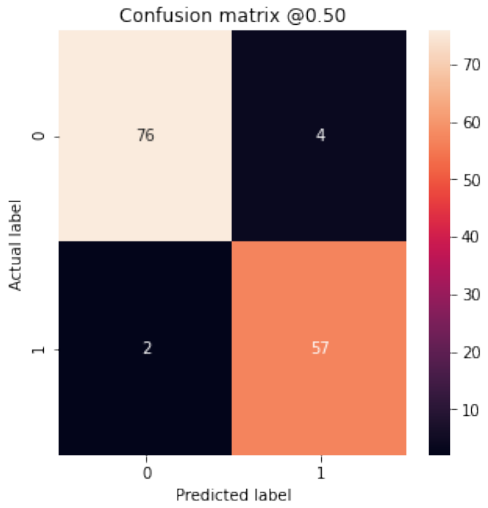


Figure 4.18: The Confusion Matrix of VGG16 Based Model with Two Block Layers Trainable on Augmented Dataset

This section puts the ROC curve of three trials in a graph for comparison (Fig. 4.19). The blue line represents the transfer learning model, the orange line represents the model trained with one block, and the green line represents the model trained with two blocks. We can see that the blue and orange polylines have intersections, and the orange is to the right of the blue as a whole, indicating that the model trained with a block performs better than the pre-trained model to predict abnormal cases. More importantly, the green polyline is above the blue and orange as a whole and has no intersection with them. It shows that the model's overall performance for training two blocks will be better than the previous two models. It also verifies our conjecture that training the basic model's pre-training weight as the initial value will make the model more effective.

4.3.4 Inception V3

InceptionV3 is not a sequential model, so we decided to unfreeze it all and train it. We looked for the probability of improving model performance.

The training metrics: loss, precision, recall, and AUC are shown in Figure 4.20. The validation loss, AUC, precision, and recall fluctuate between

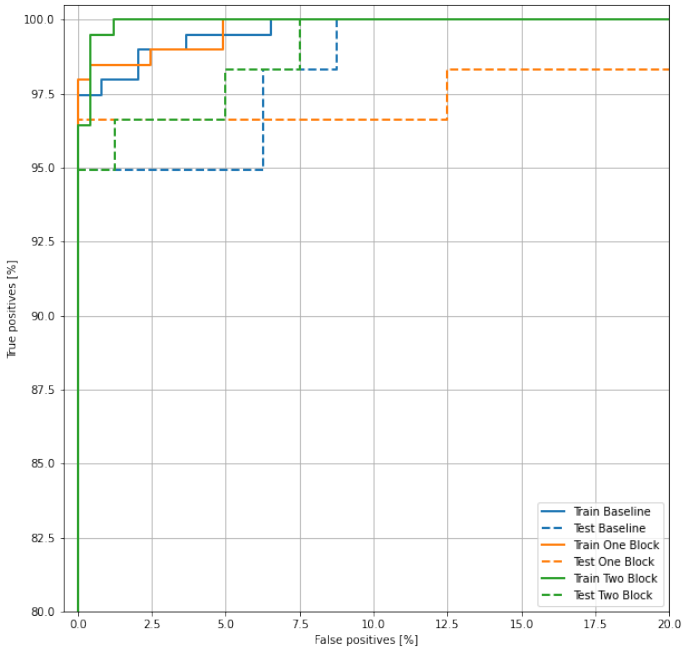


Figure 4.19: The ROC Curve of VGG16 Based Model with Two Block Layers Trainable on Augmented Dataset

15-17 epochs and may not reach convergence at the end of the training. Although precision and recall have reached a high level, it is not stabilized in the end. The training AUC, precision, and recall are also decreased between 15-17 epochs. In total, the AUC has a high level.

Then, we used the same test set as before to make predictions. The generated confusion matrix is shown in Figure 4.21. As can be seen from the figure, there are 53 true-positive cases, 0 false-positive cases, 80 true-negative cases, and six false-negative cases.

- Precision = 1
- Recall = 0.89

Because the model predicts all normal cases in the test set correctly, its precision is very high. In contrast, recall performance is not satisfactory, but it is still better than when it is used with training weights. When both are improved, it can be expected that the AUC will be larger than the previous one in 4.2.2.

We can also use the ROC curve to verify this. In Figure 4.22, We also plot the model's ROC curve using pre-trained weights and the ROC curve of

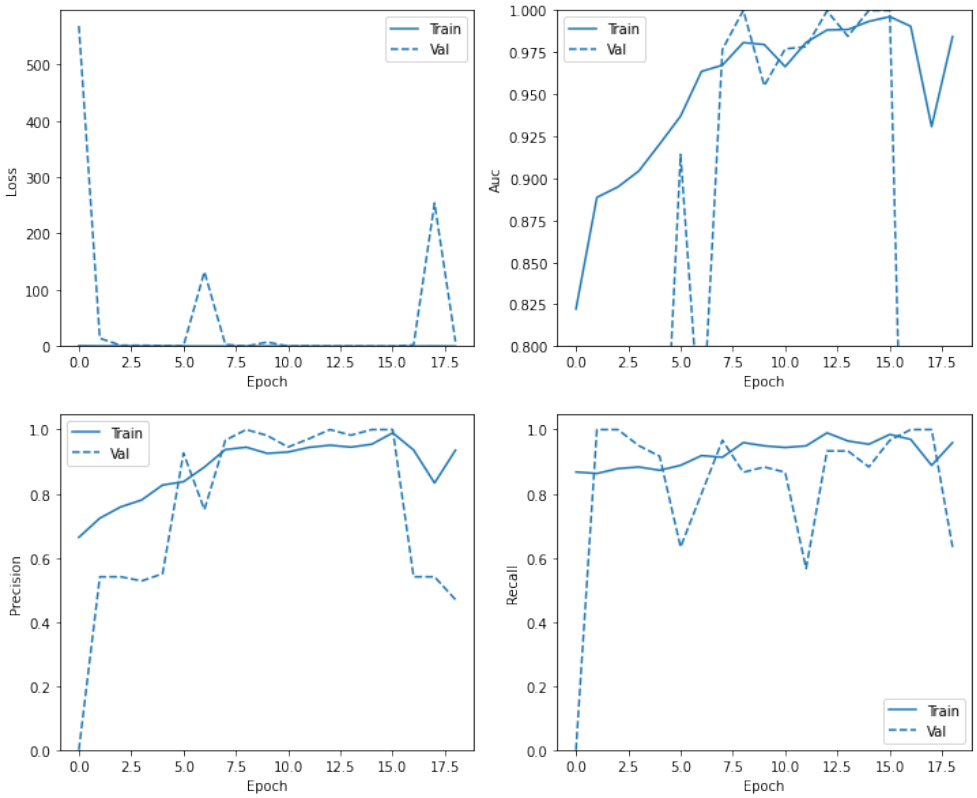


Figure 4.20: Training Metrics of InceptionV3 Based Model with Basic Model Trainable on Augmented Dataset

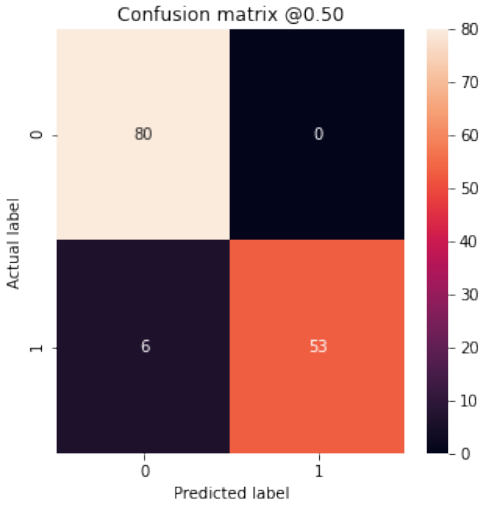


Figure 4.21: The Confusion Matrix of InceptionV3 Based Model with Basic Model Trainable on Augmented Dataset

the full model training. The blue polyline represents the original model, and the orange polyline represents the fully trained model. The orange polyline is above the blue as a whole. It means that training the basic model at the same time is much better than transfer learning.

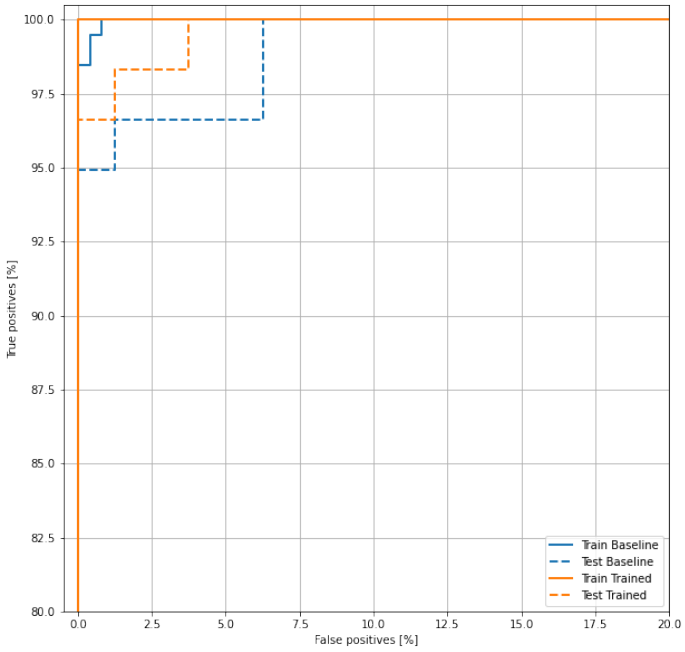


Figure 4.22: The ROC Curve of InceptionV3 Based Model with Basic Model Trainable on Augmented Dataset

4.3.5 Discussion

In comparing this section and the previous section, it is not difficult to find that VGG16 or IncepitoV3, after the same training of its convolutional layer, its precision, recall, and AUC have a certain degree of improvement.

On the other hand, in this section, the two models are compared. The recall of the VGG16 model is greater than that of the InceptionV3 model, but its precision is relatively small. It shows that VGG16 is better at predicting abnormal cases than InceptionV3, and InceptionV3 is better at predicting normal cases. In the comparison of AUC, there is not much difference between the two. Therefore, VGG16 is undoubtedly a better choice for this project that pays more attention to detecting chromosomal abnormalities. It is the same conclusion as the previous two sections.

Table 4.1: Wrong Predicted Cases for One Hundred Times Tests on VGG16 Based Model

Image No	Predicted Label	Actual Label	Mean Softmax	Confidence Score
29	Abnormal	Normal	0.69	99%
35	Abnormal	Normal	0.99	100%
98	Abnormal	Normal	0.99	100%
101	Abnormal	Normal	0.99	100%

4.4 Confidence Score

For this project, if we want to put it into market use, such as hospitals and biological laboratories, another essential function is to tell users how confident they can predict the correct chromosome picture. Such a value, we call it confidence score. For a single image, the softmax value predicted by a model does not reflect its uncertainty very well. Therefore, we introduced Monte Carlo Dropout (MC Dropout) to measure its uncertainty.

MC dropout is easy to use. We do not need to modify the existing neural network model. It only needs a dropout layer in the neural network model. Whether it is standard dropout or its variants, such as drop-connect, it is all right.

There is no difference between MC dropout and dropout during training, following the usual model training method.

During the test, the neural network's dropout cannot be closed during the forward propagation process. It is the only difference between regular use.

The MC dropout is embodied in that we need to perform multiple forward propagation processes on the same input. Under the blessing of dropout, the output of "different network structures" can be obtained. These outputs are averaged, and statistical variances are obtained to obtain the model Forecast results and uncertainty. Moreover, this process can be parallelized to be equal to one forward propagation in time.

We added MC Dropout into the two models and run 100 times predictions for every image. Then, we used the most predicted label as the predicted label and the times predicted in this label as the percentage of confidence score. Meanwhile, we calculated the mean value of softmax during 100 testings.

In the two experiments, most of them correctly predicted cases achieved a confidence score of more than 98%. Here we list some cases of prediction errors (Tab. 4.1 & Tab. 4.2).

Table 4.2: Wrong Predicted Cases for One Hundred Times Tests on InceptionV3 Based Model

Image No	**Predicted Label**	**Actual Label**	**Mean Softmax**	**Confidence Score**
0	Normal	Abnormal	0.49	54%
52	Normal	Abnormal	0.13	100%
54	Normal	Abnormal	0.11	100%
66	Normal	Abnormal	0.002	100%
72	Normal	Abnormal	0.18	100%
103	Normal	Abnormal	0.013	100%
105	Normal	Abnormal	0.15	100%
128	Normal	Abnormal	0.014	100%

Chapter 5

Conclusions

This thesis performed data pre-processing, neural network establishment and training, experiments, and comparisons under different conditions, and confidence score generation. In terms of data pre-processing, we used noising, blur, and brightness to augment data due to its integrity and orderliness. The performance of the processed data is much better than that of the pre-processed data.

We used transfer learning to build two different neural networks and trained and tested them. The results are as follows (Tab. 5.1): From the table, it is obvious that:

1. The processed data is better for neural network training.
2. The VGG16-based network is better than InceptionV3 in recall and weaker in precision.
3. Using the pre-trained weight of the basic model as the initial value, the

Table 5.1: Comparison of Different Networks and Conditions

Network and Conditions	**Accuracy**	**Precision**	**Recall**	**AUC**
original data on VGG16	0.83	0.83	0.56	0.79
original data on InceV3	0.77	0.58	0.78	0.86
processed data on VGG16	0.97	0.98	0.95	0.99
processed data on InceV3	0.94	0.98	0.86	0.99
VGG16 with one block layers trained	0.97	0.98	0.96	0.99
VGG16 with two blocks layers trained	0.98	0.93	0.96	0.99
InceV3 with entire network trained	0.95	1	0.89	0.99

training part or the entire network will get a better solution. The more layers you train, the better the effect.

For the cases that have been mispredicted, some predictions with a softmax value around 0.5 will have a low confidence score. Such results are generally not accepted.

5.1 Applications

Researchers' current method to detect chromosomal abnormalities is to perform karyotyping first manually and manually check the generated karyogram. The number of chromosomal abnormalities in the population is less than 10% of the total, so many normal chromosomes need to be manually checked by researchers. The two-manual operations have high technical requirements, which makes the workload huge, and the error rate dramatically increases. If we can screen out the cases of normal chromosomes with certainty before the investigation, the researchers only need to manually screen the abnormal chromosomes determined by the system and normal cases with low confidence scores. Work efficiency and accuracy will be significantly improved. After completion, this component will enter hospitals and biological laboratories to help the research institute perform preliminary chromosomal abnormalities investigations.

5.2 Discussions

We have analyzed chromosome abnormality, karyotyping, used transfer learning to train two supervised models for abnormality detection of karyograms, and done many different experiments to improve models' performance. This section contains our interpretation of the results.

Why not choose to train and test directly on the original chromosome image? When we first started this project, we thought about this problem. At that time, we thought of two schemes: predicting the chromosome picture directly, and the other is to perform karyotype on the chromosome picture first and then predict the result. After we checked many papers, we found that there are some obstacles in processing cell chromosome pictures, making it impossible to process the chromosome pictures as a whole. For example, many chromosomes will crossover; that is, two chromosomes overlap in the

photo. At this time, it is difficult to determine which centromere is the chromosome. Most of the processing of chromosome photos is to segment them with a bounding box. The chromosomes after karyotype are arranged in an orderly manner, each with a distinct distribution. Using such pictures for training and prediction can significantly improve the application effect. So we decided to divide it into two steps: chromosome classification and karyotyping anomaly detection.

What are the preprocessing of karyogram? We used a flip, 90° rotate, crop, rotation, shear, exposure, blur, noise, and other means when preprocessing the image in the initial experiment. The data set thus obtained performs very well on the model. Nevertheless, when we used this model, there were some problems, and it did not perform well for ordinary test pictures. The reason is actually to start with the characteristics of a karyogram. The karyogram is a picture of 24 pairs of chromosomes arranged in pairs, and the relative position of each pair of chromosomes is fixed. After performing operations such as rotation and folding, the order and relative position are changed, which affects the effect of unprocessed pictures. So later, we only kept the blur, brightness, and noise.

Why recall is a more important indicator than precision? We define abnormal cases as positive cases and normal cases as negative cases. Recall mainly reflects the model's predictive performance for positive cases, while precision reflects the model's predictive performance for negative cases. When an error occurs, it cannot be accepted that people with chromosomal abnormalities are diagnosed as normal. It will cause severe medical malpractice and social problems. Therefore, the false-negative generated by the model should be as small as possible. Therefore, recall is a more critical parameter index than precision.

Can the softmax probability generated by the neural network represent uncertainty? We have taken the probability of softmax to calculate uncertainty many times, such as the least confident, margin, and entropy in active learning query strategies. Under the entropy strategy, the more uniform the probability of the softmax, the greater the entropy. We think that the greater the uncertainty; conversely, when one dimension of the softmax is close to 1, and the others are close to 0, the uncertainty is the smallest.

However, the softmax value does not reflect the reliability of the sample classification result. A model can be uncertain in its predictions, even with a

high softmax output.[1]

Taking the MNIST classification as an example. When the model hurts the validation set, we can still get a high softmax value by inputting a picture into the neural network. The classification result is not reliable; when the model is on the validation set. The above effect is perfect, even on the test set. At this time, we add some noise to a picture, or handwrite a number to take a photo, and input it into the network. At this time, we get a higher softmax value. Do we think the results are reliable? We can understand that, in the known information, the model thinks that it is doing well, and the model itself cannot generalize to all sample spaces. For data it has not seen, its generalization ability maybe not so strong. At this time, the model still has a firm judgment on the data that has not been seen before, based on the known information (softmax has a tremendous value). Sometimes, the judgment is good, but sometimes the judgment may error, and the model cannot give the confidence of this judgment.

Moreover, MC dropout can give a predicted value and give confidence to this predicted value, which is the advantage of Bayesian deep learning.

5.3 Future Work

Better data processing. In this thesis, we only keep the blur, brightness, and noise of the data due to the data's order and integrity. However, for each pair of chromosomes in the karyogram, rotation, folding, mirroring, and other operations are all processing methods that can reasonably improve the model's accuracy. It will be a direction for future work. On the other hand, the gaps between the chromosome pairs in the karyogram are somewhat large, which can be reduced to make meaningful objects more compact and reduce the blank areas' interference on the results.

Use more updated models as base models. In this paper, we only used VGG16 and InceptionV3 as the object of transfer learning. They are all very famous and excellent models in image recognition. There are still many other models, such as AlexNet, Resnet, and so on., and updated versions of each model. We can try to compare which one can better solve our requirements.

A better calculation method of confidence score. In our evaluation, we used one hundred predictions as the percentage of its confidence score. Nevertheless, this also brings many errors. Many softmax values close to 0.5 can also be predicted on the same site multiple times. If we can combine the value

of softmax to create a formula to calculate the confidence score, our results will be more convincing.

Bibliography

[1] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050–1059.

[2] S. Rajaraman, S. Candemir, Z. Xue, P. O. Alderson, M. Kohli, J. Abuya, G. R. Thoma, and S. Antani, "A novel stacked generalization of models for improved tb detection in chest radiographs," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2018, pp. 718–721.

[3] Y. Bar, I. Diamant, L. Wolf, and H. Greenspan, "Deep learning with non-medical training used for chest pathology identification," in *Medical Imaging 2015: Computer-Aided Diagnosis*, vol. 9414. International Society for Optics and Photonics, 2015, p. 94140V.

[4] B. van Ginneken, L. Hogeweg, and M. Prokop, "Computer-aided diagnosis in chest radiography: Beyond nodules," *European Journal of Radiology*, vol. 72, no. 2, pp. 226–230, 2009.

[5] M. T. Islam, M. A. Aowal, A. T. Minhaz, and K. Ashraf, "Abnormality detection and localization in chest x-rays using deep convolutional neural networks," *arXiv preprint arXiv:1705.09850*, 2017.

[6] N. Nimitha, C. Arun, A. Puvaneswari, B. Paninila, V. Pavithra, and B. Pavithra, "Literature survey of chromosomes classification and anomaly detection using machine learning algorithms," in *IOP Conference Series: Materials Science and Engineering*, vol. 402, no. 1. IOP Publishing, 2018, p. 012194.

[7] W. Ding, L. Chang, C. Gu, and K. Wu, "Classification of chromosome karyotype based on faster-rcnn with the segmatation and enhancement preprocessing model," in *2019 12th International Congress on Image*

and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI). IEEE, 2019, pp. 1–5.

[8] Y. Qin, J. Wen, H. Zheng, X. Huang, J. Yang, N. Song, Y.-M. Zhu, L. Wu, and G.-Z. Yang, "Varifocal-net: A chromosome classification approach using deep convolutional networks," *IEEE transactions on medical imaging*, vol. 38, no. 11, pp. 2569–2581, 2019.

[9] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *Advances in neural information processing systems*, 2017, pp. 5574–5584.

[10] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in neural information processing systems*, 2017, pp. 6402–6413.

[11] K. Gopalakrishnan, S. K. Khaitan, A. Choudhary, and A. Agrawal, "Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection," *Construction and Building Materials*, vol. 157, pp. 322–330, 2017.

[12] D. Kim and T. MacKinnon, "Artificial intelligence in fracture detection: transfer learning from deep convolutional neural networks," *Clinical radiology*, vol. 73, no. 5, pp. 439–445, 2018.

[13] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, "A survey on deep transfer learning," in *International conference on artificial neural networks*. Springer, 2018, pp. 270–279.

[14] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, 2014, pp. 3320–3328.

[15] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[16] C. Zhang and Y. Ma, *Ensemble machine learning: methods and applications*. Springer, 2012.

[17] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251–1258.

[18] W. Zhang, S. Song, T. Bai, Y. Zhao, F. Ma, J. Su, and L. Yu, "Chromosome classification with convolutional neural network based deep learning," in *2018 11th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*. IEEE, 2018, pp. 1–5.

[19] X. Wang, B. Zheng, S. Li, J. J. Mulvihill, M. C. Wood, and H. Liu, "Automated classification of metaphase chromosomes: optimization of an adaptive computerized scheme," *Journal of biomedical informatics*, vol. 42, no. 1, pp. 22–31, 2009.