Analysis of Deep Learning Models For Visual Saliency With Different Noises

Author:

Yukun Ge (C21107895)

MSc Computing with Placement

Supervisor:

Dr Hantao Liu

School of Computer Science and Informatics,

Cardiff University

September 2022

Abstract

The human brain's visual attention system can quickly select where to gaze in complex scenes. Saliency prediction methods simulate the visual system of the human brain with specific algorithms to determine the probability of each pixel in an image being seen by the human eye. These have achieved good results. However, due to imperfections in imaging systems and equipment, as well as external factors such as illumination, digital images can introduce different types of noise during formation, transmission and storage, resulting in loss of image information. Therefore, in this project, various types and levels of noise were applied to the CAT2000 dataset and three models were analysed several times to explore the performance of the models in the presence of noise.

Acknowledgment

Over the course of my researching and writing this paper, I would like to express my thanks to all those who have helped me.

Firstly, I would like to express my heartfelt thanks to my supervisor, Dr Hantao Liu, for the opportunity to explore the world of machine learning, and for his generosity, encouragement and strong support .

Furthermore, Sincere gratitude should also go to all my learned Professors and warm-hearted teachers who have greatly helped me in my study as well as in my life.

Finally, I also express my appreciation to my family and friends who love and care me and whom I love and care.

Table of Contents

Abstract i

Acknowledgment ii

List of Figures v

List of Tables vii

Chapter I. Introduction 1

1.1 Problems 1

1.2 Aims and Objectives 2

1.3 Scope 2

1.4 Contributions 3

1.5 Project Flow Diagram 3

1.6 Writing System 4

Chapter II.Background Material 4

2.1 Algorithms Literature Review 5

2.2 Models Literature Review 7

2.2.1 Bottom-up gaze point detection model 8

2.2.2 Top-down Visual Detection model 9

2.4 Convolutional Neural Networks 12

2.5 Visual attention mechanism 15

2.6 Joint Attention 16

2.7 Visual saliency prediction 18

Chapter III. Methods. 22

3.1 Model Selection 22

3.1.1 VGGNet 22

3.2 ResNet Network Model 23

3.3 ML-NET 26
3.4 Loss Function - KLLoss 26
3.5 Noise 27
3.5.1 Gaussian noise 27
3.5.2 Impulse noise 28
3.5.3 Poisson Noise 29
3.5.4 Speckle noise 30
3.6 Datasets 30
3.7 Matrices 31
3.7.1 The Area under the ROC curve (AUC) 31
3.7.2 Normalized Scanpath Saliency 31
3.7.3 Earth Movers Distance 32
3.7.4 Linear Correlation Coefficient (CC) 32
Results and Analysis 36
Conclusion 42
Reflection 44

List of Figures

Fig.1. Overall Project flow. The circle and the capsule respectively indicate the start and the end.....7

Fig.2.1. Examples of eye fixation, the red dots denote the eye fixations.....10

Fig.2.2. Illustration of learning-based fixation model. (a) Training stage. (b) Testing stage.. 14

Fig. 2.2.1 Examples of the situation that contrast information important than semantic information, the first line shows images, the second line shows the result maps of the neural network method 15

Fig. 2.3 Basic structure of neural network.....16

Fig 2.4 Convolution operation.....17

Fig 2.4.2 Activation function.....18

Fig. 2.4.3 Max pooling with a 2x2 filter and stride =2.....19

Fig.2.5 Visual attention mechanism.....20

Fig 2.6 Joint Attention Example.....21

Fig 2.7 Significance prediction Example.....23

Fig 2.8.1 RGB Colour spectrum.....24

Fig 2.8.2 HIS Colour spectrum.....24

Fig 2.8.3 LAB Colour spectrum..... 25

Fig 3.1 VGG16 Structure Diagram.....25

Fig 3.2 Residual block.....26

Fig 3.2.1 ResNet-50 Architecture.....27

Fig 3.2.2. ML-Net Architecture.....28

Fig 3.5.1 The probability density curve of Gaussian noise.....30

Fig 3.5.2 Probability density curve of the impulse noise.....31

Fig 3.5.3 Probability density curve of the possion noise.....31

Fig 4.1 Complexity of Model code.....38

Fig 4.1.2 Number of Parameters of ResNet(left), VGG(middle),ML-NET(right).39
Fig 4.2 The line plot for the results.....40
Fig 4.3 Different noises.....40
Fig 4.3.2 The line plot for the overall result with all noises.....41

List of Tables

Table 3.2 ResNet network structure with different depths.....30

Table 4.2 Overall results for 3 models.....39

Table 4.3.1 Gaussian noise results for 3 models.....41

Table 4.3.2 Poisson noise results for 3 models.....42

Table 4.3.3 Speckle noise results for 3 models.....42

Chapter I.

Introduction

1.1 Problems

With the rapid development of information and communication technology, multimedia technology and the increasing popularity of the Internet, the speed and scale of information collection and dissemination have reached an unprecedented level. Humanity is confronted with profound changes in information: firstly, the volume of data is increasing; secondly, the types of data received are becoming more and more varied. The increasing expansion of information makes it difficult, if not impossible, to process this data manually, which requires the technical means of computers to process large amounts of data quickly, and so artificial intelligence is born. Artificial intelligence uses computers to simulate specific human thought processes and intelligent behaviour: learning, reasoning, thinking, planning, etc. It is considered, along with genetic engineering and nanoscience, to be a frontier technology with great potential for development. After more than three decades of rapid development, artificial intelligence has gradually become a discipline in its own right. Today, AI is widely used in many fields and has achieved excellent results. The main application areas include image recognition, natural language processing, robotics, etc. Computer vision is an important branch of artificial intelligence, which focuses on the use of computers and related devices to simulate biological vision and process collected images or videos to achieve an understanding of scenes. Computer vision includes image processing, pattern recognition and image understanding. Visual saliency analysis, a popular research area in computer vision, has recently been sought after by researchers.

Visual saliency analysis originated from the study of the human visual system and its goal is to simulate the working mechanism of the human visual system using computer vision-related algorithms. The human visual system receives several hundred megabytes of visual information per second, but its information is processed

at a rate of only 40 bits per second. Through visual attention mechanisms, humans can quickly find areas of interest in complex scenes. Early stages produce a unique subjective perceptual quality - saliency - for each location in a visual scene. The human brain has evolved to automatically and in real-time calculate saliency for each location throughout a visual scene. Visual attention is drawn to the salient objects or regions in the visual scene. The visual attention mechanism directs the human eye to salient regions in a large amount of data. It allocates resources to prioritise the salient regions, thus effectively reducing the computational load on the human visual system. Visual saliency analysis simulates the human visual attention mechanism. Through the computer's analysis of prominent locations in the visual scene, the computer can allocate resources to prioritise the processing of prominent areas. This increases the efficiency of image processing and reduces the time consumed.

Drawing on human visual attention mechanisms and the excellent performance of neural networks, the researchers have investigated an attention point prediction model suitable for computer simulation. By using the point-of-gaze prediction model as a icritical component of information filtering and prioritisation of computational resources in machine vision systems, they enhance the ability of computer vision systems to handle large amounts of digital media and improve the utilisation of digital media resources. In this project, I will analyse the performance of different models and apply noise to the same models to explore how the models perform under different noise.

1.2 Aims and Objectives

The aim of this project is:

1. Adding the noises to the data.
2. Performing experiments on deep learning saliency prediction models.
3. Find which models have good generalization ability to noises.

1.3 Scope

The scope of this project includes:

1. Datasets training and evaluation on CAT2000 [1].
2. Noise applying to datasets.
3. The deep learning tool was built in python 3.7.5 using TensorFlow as the main library.
4. Evaluating metrics implemented were AUC, CC, and NSS.

1.4 Contributions

To summarise, the contributions of this work include:

1. A comparison of performance among ResNet-based, multi-level VGG-based, and machine learning-based models.
2. An analysis of the effect of using three different noise methods.
3. An analysis of model performance over different noise-adding methods.
4. Findings of limitations with different models.

1.5 Project Flow Diagram

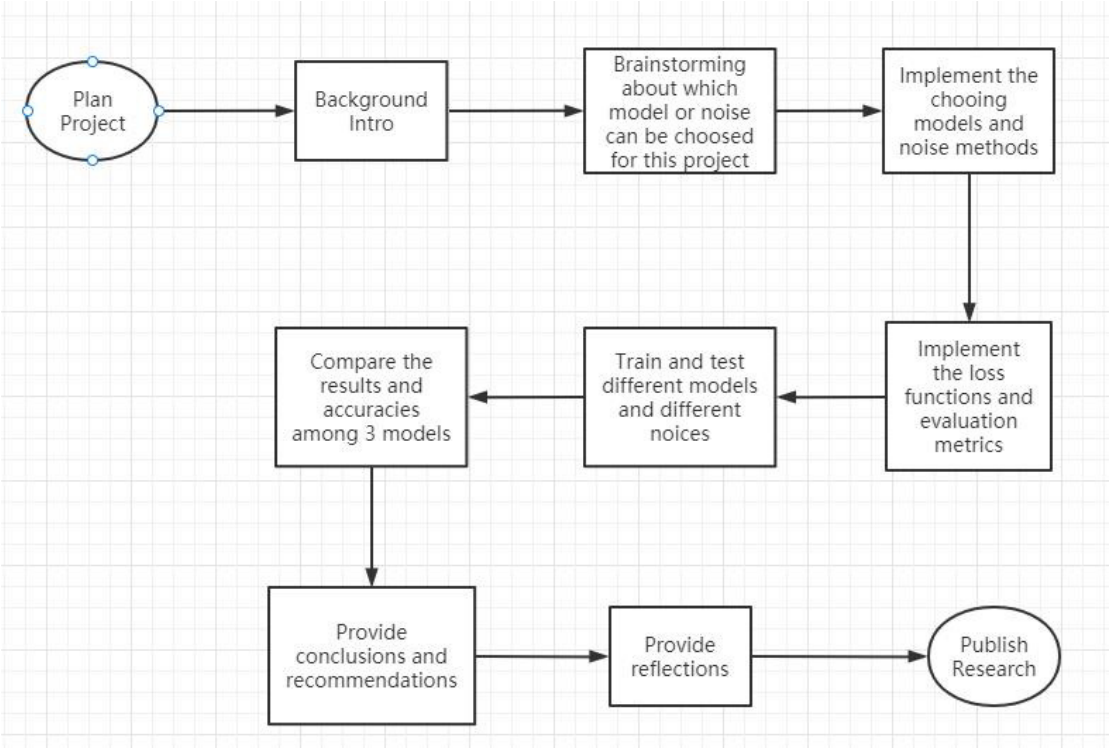

Fig.1.

Overall Project flow. The circle and the capsule respectively indicate the start and the end.

1.6 Writing System

This dissertation is organized into six chapters, which are:

- **Chapter I. Introduction**

  Goal, scope, contributions, low diagram, and writing system of project.

- **Chapter Ⅱ. Background Material**

  Summary of background techniques.

- **Chapter III. Methods**

  Experiment methods and implementation details.

- **Chapter IV.Results and Analysis**

  Analysis and visualization of experiment results.

- **Chapter V. Conclusion**

  Conclusion.

- **Chapter VI.Reflection**

  Descriptions of knowledge and skills gained by the student during this work.

Chapter Ⅱ.

Background Material

According to neuropsychology, saliency detection studies can be divided into two types in terms of what attracts attention: image data-driven bottom-up models and task-driven top-down models. The bottom-up model is a low-level cognitive process. The earliest models of saliency detection were bottom-up models, where biological principles inspired them that regions different from the surrounding environment were more likely to attract human attention. Most traditional methods typically model the attention point detection problem at this stage through feature extraction, feature comparison inference and multi-feature graph fusion. The features referred to here are low-level or manually calibrated features such as colour, texture, luminance and colour histograms. The classical algorithm for saliency detection was first proposed by Itti [3]. They extract three features of the image: luminance, colour and orientation, and compute an initial map by means of the around-centre contrast principle. Finally, multiple initial maps were fused to obtain the final saliency map. Inspired by Itti, many people have improved this framework [4-5]. They either used different features, different methods of computing saliency maps, or different methods of fusing initial maps.

2.1 Algorithms Literature Review

Over time, researchers have come to recognise the importance of semantic information. When people look at an image, they first look at familiar areas, such as faces and text. Therefore, M. Cerf and A. Borji et al. detected faces, text, cars, etc. in images as semantic features [6-7]. However, a portion of manually selected objects cannot represent all semantic information. The application of deep learning in computer vision has made it possible to extract semantic information in a wide range of ways. deep features were first used for saliency detection by Qi Zhao et al. [8], who connected deep features with underlying features as a new feature. Subsequently,

deep neural networks were explicitly designed for eye-dot detection. To date, most of the best-performing models for eye-movement point detection have been based on deep neural network structures. A new direction for eye-point detection has become adapting the network structure to make it more suitable for eye-point detection and fusing a priori cues.

Figure 2.1 Examples of eye fixation, the red dots denote the eye fixations.

E Vig et al. proposed the eDN model [9], the first saliency detection model to apply deep learning. The eDN model was trained on a small-scale database with a linear combination of three different depth features, and the model used the saliency features. Insignificant image chunks are used as training samples. Another model that uses small blocks as the main training unit is the multi-scale convolutional neural network proposed by Han et al. which proposes a multi-scale convolutional neural network (MS-CNN) [10]. This model consists of three convolutional neural networks, each taking blocks of images of different sizes as input and outputting predictions representing saliency. The final regression layer fuses these three networks. The final regression layer does the fusion of the three networks. This multi-scale convolutional network prediction method can effectively extract both the underlying and the higher level information of an image. However, as the model uses image blocks as input, it leads to high computational complexity and also makes the model. It also prevents the model from capturing global information.

Kruthiventi et al. proposed the DeepFix model in 2015 [11], using a fully convolutional neural network to accomplish end-to-end gaze-point prediction. For the

first time, the diffusion of convolutional layers was applied to improve the resolution of images and they took different approaches to fuse the central prior information. A different approach was taken to fuse the central prior information. It is worth mentioning that the last convolutional layer of the network with a larger scale perceptual field can capture the global information to some extent. However, the zero-padding operation limits the sensitivity of extracting global information. The larger the perceptual field, the more significant the perceptual field and the more pronounced the effect. Another reason for DeepFix's success is its pre-training of the Silicon database, an extensive salinity database [12], whose publication has contributed significantly to the development of salinity models.

Existing models for gaze point detection have made significant progress in terms of predictive effectiveness, but two technical challenges remain. The first is the real-time nature of gaze detection. The first is the real-time nature of gaze point detection, which is often nested within more complex computer vision systems and acts as pre-processing. The first is the real-time nature of gaze point detection, which is often nested within more complex computer vision systems and acts as pre-processing, so its real-time nature is highly demanding. The time complexity of the point-of-gaze prediction algorithm is assumed to be too high. In this case, the pre-processing process will take up a significant amount of time, directly affecting the performance of the overall computer vision system. The second difficulty is the accuracy requirement. Again, due to the pre-processing nature of gaze point detection, high accuracy is a requirement. Otherwise, the results of gaze point detection will affect the accuracy of the whole system.

2.2 Models Literature Review

Visual attention detection algorithms can be divided into a data-driven bottom-up model and a task-driven top-down model. This section begins by describing both types of detection algorithms. Bottom-up models typically use heuristic features such as contrast, position and texture. These heuristic features are known as priors in

viewpoint detection, with the contrast prior being the most commonly used one. The contrast prior includes both local and global contrast. Depending on the contrast utilised, viewpoint detection algorithms can be divided into local and global contrasts by feature. Graphical computational methods can be divided into local and global attention point detection models. The local detection model is sensitive to the image. Local detection models are sensitive to high-frequency information, such as edges and noise.

Conversely, only the edges of important targets are often detected and the edges of important targets are ignored due to the lack of global information. Local detection models are sensitive to high-frequency information, such as edges and noise. The opposite is true for the corresponding global detection models.

In contrast, task-driven top-down models typically use external cues to make predictions, including the actual value of the point of view and similar images. The so-called external cues include the actual value of the gaze point and similar pictures, etc. Top-down gaze detection algorithms based on convolutional neural networks are the mainstay of recent developments. Therefore, this section describes the neural network-based gaze detection algorithm.

2.2.1 Bottom-up gaze point detection model

Typically, a bottom-up model usually includes the following components:

(1) Extracting features: Common features include contrast, texture, brightness, colour, etc. Image processing techniques can be used to enhance or transform the underlying features. The basic units of feature extraction include pixel-based, block-based, and region-based.

(2) Calculation of feature maps in specific feature dimensions to measure saliency: feature maps are usually calculated using Gabor filters or more

sophisticated methods. For example, Itti and Baldi assumed that information-theoretic concepts are central to saliency and adopted Bayesian statistical theory to compute the saliency map [15]. Gao et al. used discriminative pericentric assumption to measure significance [16]. Raj et al. used an entropy minimization algorithm to select the attention points [17]. Bruce et al. proposed a self-information model based on an independent component analysis decomposed self-information model [18], which is consistent with the principle of the sparsity of cortical cell responses to visual input.

(3) Fusion of various feature maps to obtain the final saliency map: Early psychological and physiological studies supported fusion approaches such as linear summation and maximization in the field of saliency detection. The former linear summation approach was widely used in early gaze point detection models. The former linear summation approach was widely used in early point-of-gaze detection models. Later, Itti and Koch proposed different methods to normalize the feature maps according to their distribution [19]. The composite significance index combines spatial compactness and significance density.

Bottom-up gaze point detection methods usually run slowly, while manually calibrated features make gaze point detection systems incapable of detecting semantic information, which is crucial for gaze point detection.

2.2.2 Top-down Visual Detection model

Task-driven top-down models often use external cues to make predictions, so-called external cues include the actual value of the point of view and similar pictures. Learning-based point-of-view detection models are often modelled as a classification or regression problem. A mapping function is learnt to map high-dimensional feature vectors and significant scalar values. A typical top-down approach to viewpoint detection learning involves a learning and testing phase, as

shown in the figure below.

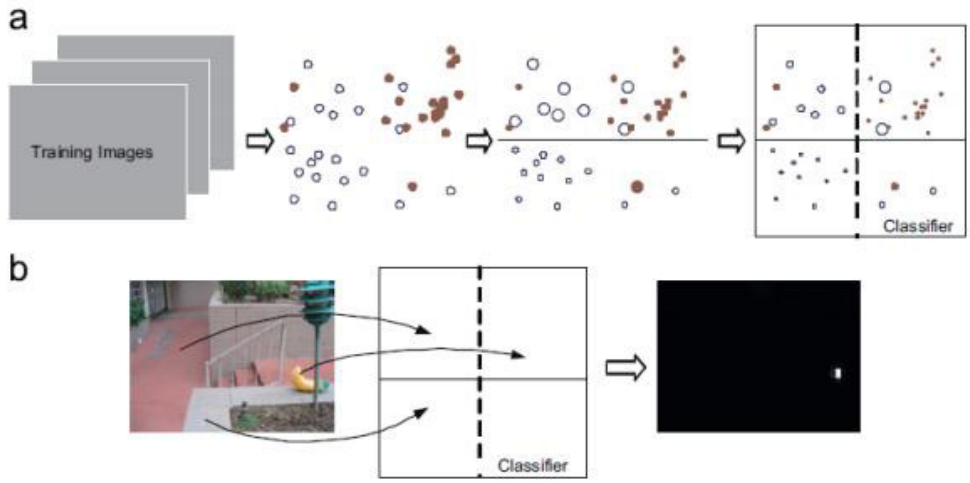

Fig. 2.2 Illustration of learning-based fixation model. (a) Training stage. (b) Testing stage.

The neural network-based learning model is also a top-down model that can effectively break through the limitations of traditional top-down learning. It can effectively break through the limitations of traditional top-down learning methods and achieve better attention point detection.

E Vig et al. proposed the first application of deep learning for the eDN model [9]. eDN model is trained on a small-scale database, its features include three different linear combinations of depth features, and the model uses significant and insignificant image blocks as training samples. Another model that uses small blocks as the basic unit of training is the multi-scale convolutional neural network (Mr-CNN) proposed by Han et al. [10]. This model consists of three convolutional neural networks, each of which uses image blocks of different sizes as input, and the output prediction values represent the significance. A final regression layer does the fusion of the three networks. This multi-scale convolutional network method for predicting the point of attention can effectively extract the image's underlying and higher-level information. However, since the model uses image blocks as input, it leads to high computational complexity and, at the same time, prevents the model from capturing global

information.

Kruthiventi et al. proposed the DeepFix model [11] in 2015, where they used a fully convolutional neural network to accomplish end-to-end gaze-point prediction. The diffusion of convolutional layers was applied for the first time to increase the resolution of images, and they took a different approach to fuse the central prior information. Cornia et al. proposed ML-NET [13], which fuses features extracted from different layers in a convolutional neural network. The model consists of three modules: a feature extraction convolutional neural network, a feature encoding network, and an a priori learning network. Kruthiventi proposed a deep neural network for both salient target detection and gaze point detection [14], with both tasks sharing the initial network part.

Although convolutional neural networks show great ability in extracting semantic information, for images with bottom-up contrast information attracting attention, the neural networks are not as effective as traditional gaze point detection algorithms. For images where the underlying contrast information attracts attention, the neural network is not as effective as the bottom-up traditional gaze point detection algorithm, as shown in Figure 2.2.1

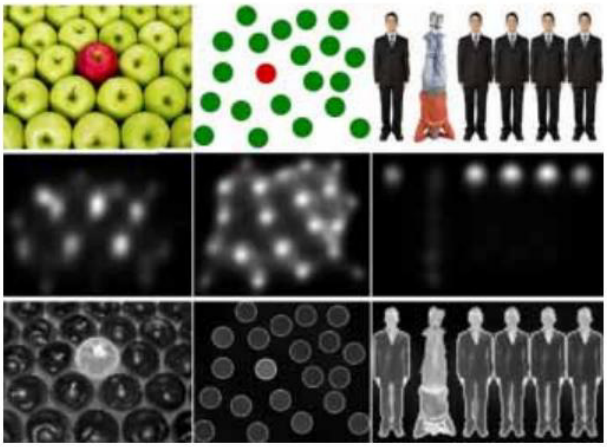

Fig. 2.2.1 Examples of the situation that contrast information important than semantic information, the first line shows images, the second line shows the result maps of the neural network method

2.3 The basic structure of neural networks

The two-layer neural network structure is the interconnection pattern of early neural network models, and this interconnection pattern is the simplest hierarchical

structure. And the neural network structure with three layers and more than three layers is called a multi-layer neural network structure. All neurons are divided into several layers according to their functions. Generally, there are the input, hidden, and output layers. The neurons on the nodes of the input layer receive input patterns from the external environment and pass them from it to the individual neurons on the connected hidden layer. The hidden layer is the internal processing layer of the neural network, and they have no direct connection with the external input and output, so they are called hidden layers. The pattern transformation capability of the artificial neural network is mainly reflected in the hidden layer's neurons. The output layer is used to generate the output of the neural network.

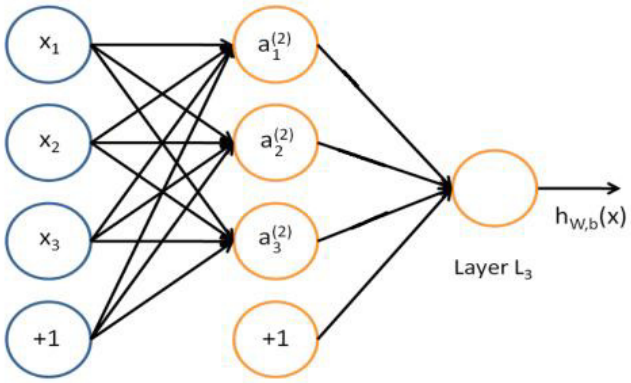

Fig. 2.3 Basic structure of neural network

In figure 2.3, the blue circles represent the inputs to the network, and the circles marked with "+1" represent the bias cells, which correspond to the intercept terms. The leftmost layer in the Figure is the input layer, the rightmost layer is the output layer, and the middle layer is the hidden layer.

2.4 Convolutional Neural Networks

A convolutional neural network (CNN, or ConvNet) is a feed-forward artificial neural network in which the connections between neurons are inspired by the organization of the animal visual cortex. The fixed area where a single cortical neuron responds to a stimulus is called the receptive field, and the receptive fields of different

neurons partially overlap. The response of a single neuron to a stimulus within the receptive field can be mathematically simulated using convolutional operations. Convolutional neural networks are also translation invariant or spatially invariant artificial neural networks.

Convolutional neural networks consist of a series of different layers and several commonly features:

(1) Convolution layer: convolution layer is the core part of CNN, and its primary role is to extract the basic features of the input image. The convolution operation is the process of multiplying and summing the corresponding elements by sliding a convolution kernel of appropriate step size between small blocks of input images with the exact dimensions during the operation. The convolution operation process is shown in the following Figure, which requires the input 3D shapes [20] to have the same width, height, and depth. The image is scanned, and the feature map is obtained using the convolution operation [21]. The hyperparameters of the convolution layer include the size of the convolution kernel, the number of kernels, the step size, and whether to fill the complementary zeros.

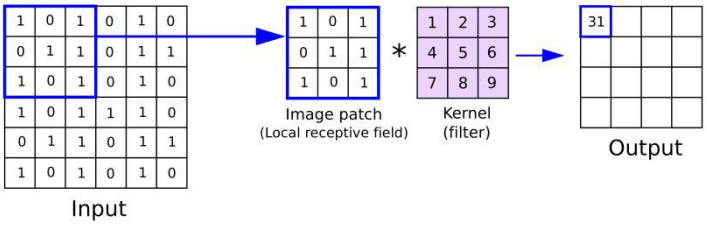

Fig 2.4 Convolution operation

(2) Activation function: The role of the activation function is to increase the expressiveness of the linear model by introducing nonlinear factors. In neural networks, a linear transformation is still obtained after superimposing each layer on top of each other. There would be no point in using a deep neural network model if an activation function is not included in this process [22]. Including an activation function introduces nonlinearity and enables the network to model more complex functions. The commonly used activation functions, along with

their function images and mathematical expressions, are shown in the following graph.

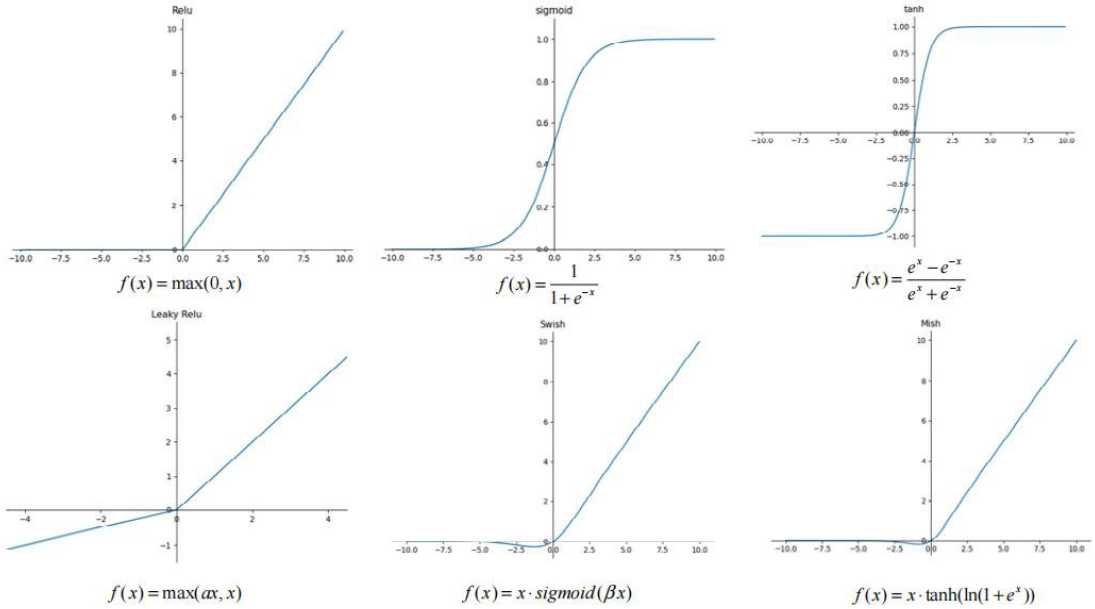

Fig 2.4.2 Activation function

(3) Pooling layer: The pooling layer is usually found in the middle of two convolutional layers, mainly to compress the image. The pooling operation, which can also be called subsampling, can extract local feature information of the image and reduce the number of parameters of the next convolutional layer while reducing the size of the feature map, which in turn can reduce the number of parameters of the whole network, which is very helpful to speed up the network training [23]. In addition, pooling layers can suppress the overfitting of the network to a certain extent and accelerate the convergence of the network parameters.

The everyday pooling operations are max-pooling (maximum pooling), vanpooling (average pooling), and global pooling (global pooling). Among them, max-pooling is to extract the feature value of the most prominent feature in the window, i.e., the maximum value, discarding the others. Vanpooling is to extract the average value of the features present in the window as the sliding window keeps

moving. Global pooling is to obtain the global relationship and then output it as a unit of each feature map.

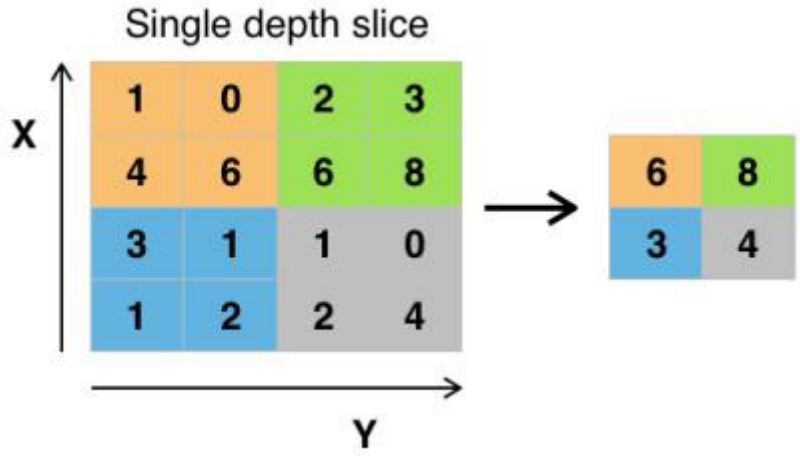

Fig. 2.4.3 Max pooling with a 2x2 filter and stride =2

(4) Fully connected layer: The fully connected layer acts as a "classifier" in the whole CNN and is usually applied in the later layers of the CNN. The fully connected layer is able to fit the image feature information extracted from the convolutional layer. It is suitable for optimizing the function during training, thus allowing the whole model to approximate the target to be trained. However, its parameters are substantial, occupying about 80% of the total network parameters. In general, the fully connected layer's length, width, and activation function affect the fully connected layer's performance for the whole network model.

2.5 Visual attention mechanism

The most fundamental problem of visual attention is addressing the selection of attention, which depends on selecting the "visual object" in the environmental scene. In the process of target tracking to locate the target position and search to detect the border size, adding visual attention can improve the focus on the target and thus obtain more accurate information about the target's position. Visual attention can be

divided into feature-based, spatial location-based, object-based, and other-based visual attention. Traditional visual attention is mainly based on spatial location. Bahdanau[24] first introduced an attention mechanism in natural language processing to reduce the source sequence length by collecting information over time. The attention mechanism has thus been widely applied in various aspects, including computer vision.

Invoking the attention mechanism in computer vision can help the system learn attention effectively [25] and reduce the possibility of extracting distracting information from video scenes, which not only saves resources but also improves the training speed and reduces time consumption. The use of the attention mechanism can obtain the target of the system's attention region, reduce the interference of background information to the model, and achieve accurate localization of the target location. The attention mechanism diagram is shown in Figure 2.5.

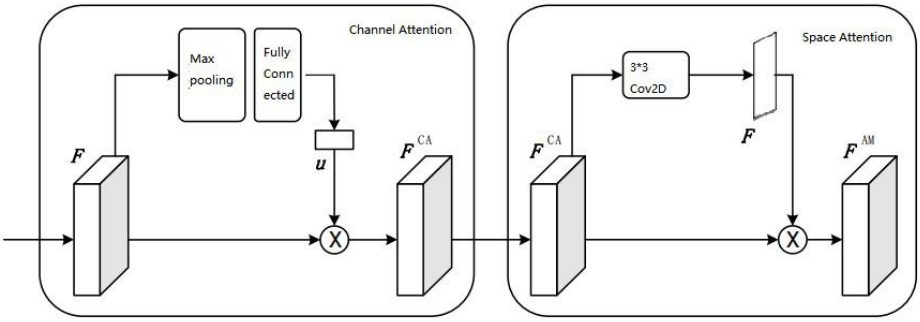

Fig.2.5 Visual attention mechanism

2.6 Joint Attention

Joint Attention (JA) is the process of shared attention in which a person makes visual contact with another person in a three-way interaction by following the direction of the other person's gaze or gazing at an object and shifting their gaze to focus on that direction or object. In early communication, infants are able to distinguish between the interactions of mutual and joint attention by three months of

age [26]. Joint attention is crucial for language and imitation learning, and observers can understand joint attention using environmental cues. Grover [27]. proposed a method to construct a 3D social salience field and locate multiple gaze behaviors in social scenes from videos captured by head-mounted cameras. In addition, they used joint attention as a constraint to predict social behaviors in first-person videos, such as future actions and future gaze directions of individuals in social groups. The predicted behaviors reflect the individual's physical space that can take the next action by engaging in joint attention while conforming to social behavior. These works explore well the detection and application of joint attention in social activities. However, they focus only on first-person videos and do not extend to ordinary third-person videos. An example diagram of joint attention is shown in Figure 2.6.

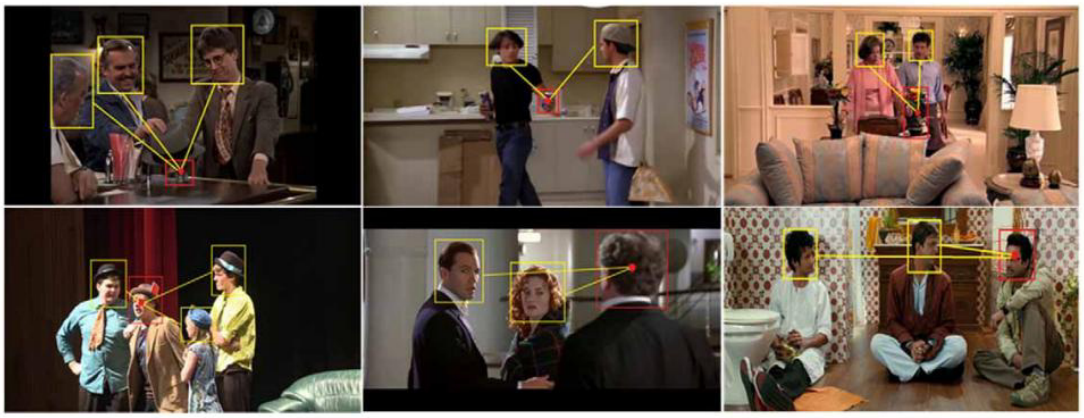

Fig 2.6 Joint Attention Example

The field of Human-Computer Interaction (HCI) enables simple interaction between humans and machines using natural communication. In HCI communication, many techniques, as well as unsolved problems, require the introduction of joint attention for reference, thus becoming a key challenge in achieving the task of joint human-machine attention and visual tracking with or without external evaluation. The critical point of this work is how to infer the direction of the human visual gaze and then control the robot's head turning to reach the interactive communication between human and machine, forming the final joint.

The critical point of this work is how to infer the direction of the human visual gaze and then control the robot's head turning to achieve the interaction between human and machine to form the final joint attention system. It is interesting to improve the joint attention in HCI because not only can the robot be controlled to complete the corresponding commands, but also the robot can learn and detect and join the continuous joint attention in the environment by itself.

2.7 Visual saliency prediction

The purpose of visual saliency for prediction is to obtain the location of the object of attention in an image, which depends not only on low-level features in the environment, such as luminance, colour, texture, etc. but also on high-level features in the scene information and task demands, such as task drivers and center bias phenomena. Greenberg [28] proposed a new saliency-based visual attention algorithm for object acquisition. They automatically extracted visual attention points (PVAs) in the scene based on saliency attention maps with different features, where each saliency attention map represents a specific feature domain. A feature selection based on detection probability, false alarm rate, and repeatability criteria is also used to select the saliency map's most practical combination of features. Assuming that the extracted PVAs represent the most visually salient regions of the image, object acquisition using the visual attention approach has a better performance compared to other detection algorithms.

Deep learning methods are excellent for learning visual saliency based on their strong learning ability and ability to incorporate both global and local scenes into their predictions. Saliency prediction generally includes bottom-up approaches and more goal-oriented top-down approaches. Bottom-up approaches can be understood as drawing the target's attention to salient regions of an image due to data-driven influences, usually using low-level features of the image in contrast to the scene to calculate the saliency of the region, and thus are primarily used for high-contrast scenes. The top-down approach is influenced by the subjective consciousness of

humans and uses specific features of the image part to calculate the saliency under the control of human consciousness. However, it is difficult to understand the human brain structure well, and there are always computational shortcomings, so this method has very few practical applications in vision. An example of saliency prediction is shown in Figure 2.7.

Fig 2.7 Significance prediction Example

2.8 Colour Space Theory

Colour is the most intuitive visual feature of an image. People can easily identify the meaning of an image based on its colour characteristics. The colour space is also called the colour channel. Colours are expressed in different ways in different colour spaces. Three common colour spaces relevant to this thesis are described below.

2.8.1 RGB Coluor spectrum

The RGB colour space, also known as the three primary colours, is the one that is most relevant to us in our lives. In the human visual system, all colours presented on the retina can be represented by a combination of red, green and blue. Currently, most video display systems use this colour model. The three RGB electron guns emit different electrons to the phosphor screen, which contains phosphors that sense each of the three different RGB colours, and the electrons excite the phosphors and produce different colours of light in linear combinations, thus mixing to produce different colours. Figure 2.8.1 shows the RGB colour space: at the origin of the

coordinate system, the component values of RGB are all 0, forming black, and at the maximum value of RGB is 255, forming white.

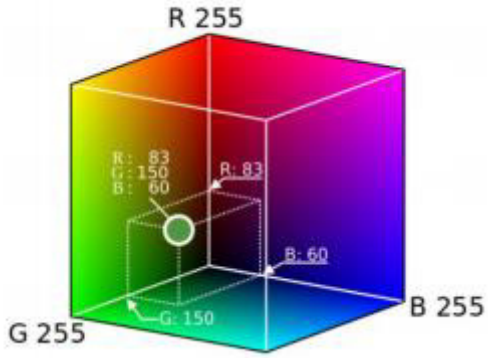

Fig 2.8.1 RGB Colour spectrum

2.8.2 HIS Colour Space

The HIS (hue-intensity-saturation) colour space is also a common colour space in computer technology, which stands for hue, saturation and luminance, respectively. The HIS colour space is shown in Figure 2.8.2. Hue represents the wavelength reflected from an object, and generally refers to the colour, which is represented in the figure by the angle between each axis and the central axis, ranging from 0 to 360 degrees, with different angles representing different colours. Saturation represents the intensity of the colour and can be expressed as the length of the radius from the central axis to the coloured point in the diagram. It ranges from 0% to 100% and represents the ratio of gray to hue. Brightness is the relative lightness or darkness of a colour, and in the diagram is the height on the vertical axis, with black at the lowest brightness and white at the highest. The human visual system is more sensitive to luminance than to colour intensity. Therefore, although the cone model is more complex, it can better represent the hue, brightness and saturation clearly, that is, different colours can be accurately displayed with the HIV colour model. Therefore, the HIV colour space is often used in colour processing.

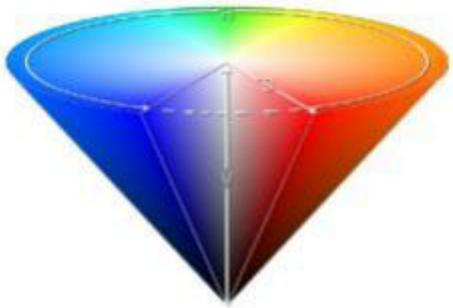

Fig 2.8.2 HIS Colour spectrum

2.8.3 LAB Colour Space

LAB is called CIELAB, and any colour in nature can be expressed in LAB space, where L represents luminance, A represents green to red component, and B represents blue to yellow component. Compared with RGB and HIS space, LAB space has the feature of uniformity of visual perception, and the magnitude of change of LAB space components is basically the same as that of human visual perception, and has the feature of device-independence. Therefore, LAB has a larger colour range than other colour spaces, and there is no need to care about overflow when converting colours from RGB to LAB colour space. Figure 2.3 shows the LAB colour space: the vertical axis is the luminance axis L, and the horizontal coordinate system represents the green to red component and the blue to yellow component.

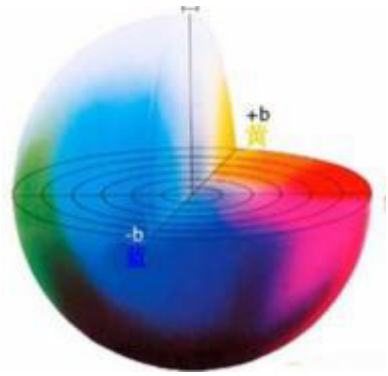

Fig 2.8.3 LAB Colour spectrum

Chapter III.

Methods

3.1 Model Selection

3.1.1 VGGNet

Due to the widespread use of convolutional neural networks in the field of computer vision, many researchers started to try to improve the structure of the network to enhance the performance of neural networks. 2014, Simonyan [29], a research group at the University of Oxford, proposed a deep neural network series model VGGNet network architecture (including VGG11, VGG13, VGG16, and VGG19) and won second place in the ImageNet competition for classification and first place for localization.

VGGNet has the advantage of using convolutional kernels with smaller fields of perception (3 × 3 convolutional kernels) instead of more extensive fields of perception (5 × 5 convolutional kernels or 7 × 7 convolutional kernels), which reduces the number of parameters while increasing the nonlinearity of the network. VGGNet also introduces a 1×1 convolutional layer, which can resize the convolutional kernels to augment or reduce the data, add additional activation functions to introduce more nonlinearities without changing the input size, and finally, it can perform up- and down-dimensioning of the feature map. It is also able to perform dimension raising and lowering operations on the feature map.

The structure of VGG16 is shown in Figure 3.1. It has 16 layers, including 13 convolutional layers and 3 fully connected layers. In the training process of the network model, the same padding is used, i.e., the size of the output layer is equal to the size of the input layer, the max-pooling operation is used after each convolutional layer, and the number of channels of the last three fully connected layers is 4096,

4096, and 1000 respectively, and then SoftMax is used to output the classification.

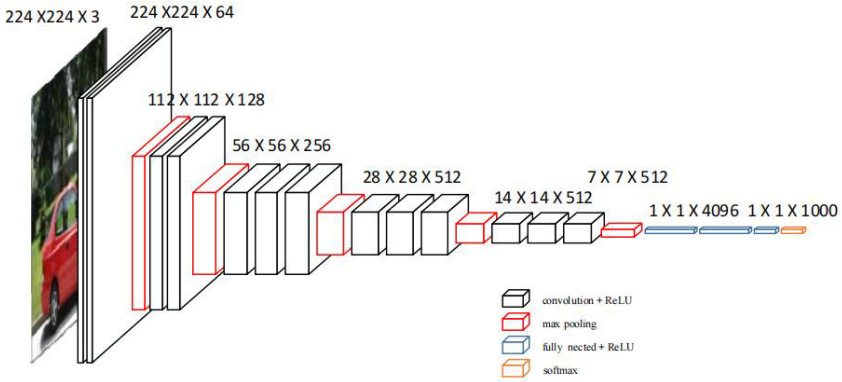

Fig 3.1 VGG16 Structure Diagram

3.2 ResNet Network Model

With the continuous research of neural networks, the number of layers of the network is increasing. The network model is able to extract more complex feature information (lower, middle, and higher layers). The performance of the network will be relatively more superior in theory. However, in practice, after the network reaches a certain depth, the performance of the network shows a decreasing trend instead. One of the reasons for this is that there is gradient disappearance or gradient explosion, and the parameters of the network layer cannot be updated because the deeper network has difficulty in effectively transferring the later gradients to the front network layer during backpropagation. On the other hand, it is because of the network degradation problem brought by blindly increasing the network depth.

To solve this problem, Kai-Ming He of Microsoft Research [30] proposed the residual learning framework ResNet (Deep Residual Network), which solves the network degradation problem by introducing a residual block, which not only enables the number of layers of the network can reach more profound, but also makes the network less challenging to train, and won the championship in the 2015 IamgeNet The network was awarded the first prize in the 2015 IamgeNet competition. The structure of the residual block is shown in Figure 3.2

ResNet introduces the concept of identity mapping because it is challenging to fit a potential expectation mapping H(x) directly, so we can use the residual function F

(x) to design the network as H(x) = F(x) + x. This can be converted to learn the residual function F(x) = H(x) - x, and the difficulty will be reduced. The final desired mapping H(x) can be fitted with F(x) + x. The results show a significant improvement, and the network is easier to optimize because of the increased depth. When F (x) = 0, then H(x) = x constitutes a constant mapping. Since this approach reuses the intermediate feature layers, it can effectively solve the problem of network degradation.

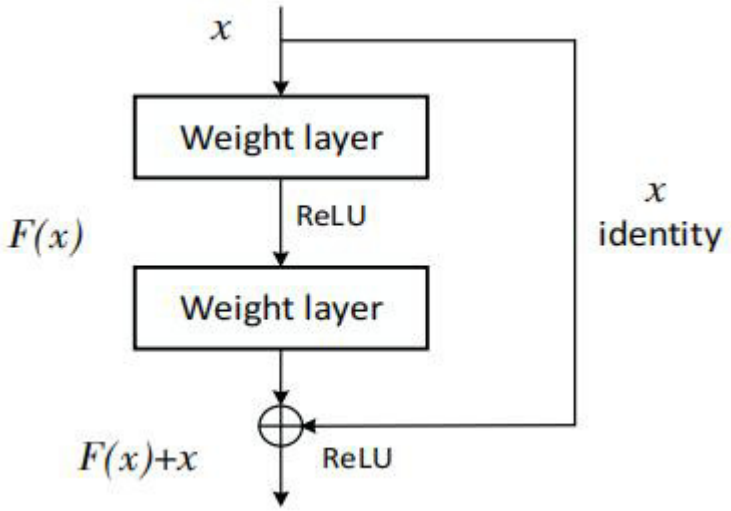

Fig 3.2 Residual block

Assuming that the layers behind the deep network are continuous mappings, the deep network can be transformed into a shallow one. ResNet can stack many residual blocks together at the same time to obtain a deeper neural network structure (e.g., 34, 50, or 152 layers, etc.), and the network structure of ResNet with different depths is shown in table 3.2.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112 X 112	7 X 7 , 64 , stride 2				
conv2_x	56 X 56	3 X 3 max pool , stride 2				
		[3 X 3 , 64; 3 X 3 , 64] X 2	[3 X 3 , 64; 3 X 3 , 64] X 3	[1 X 1 , 64; 3 X 3 , 64; 1 X 1 , 256] X 3	[1 X 1 , 64; 3 X 3 , 64; 1 X 1 , 256] X 3	[1 X 1 , 64; 3 X 3 , 64; 1 X 1 , 256] X 3
conv3_x	28 X 28	[3 X 3 , 128; 3 X 3 , 128] X 2	[3 X 3 , 128; 3 X 3 , 128] X 4	[1 X 1 , 128; 3 X 3 , 128; 1 X 1 , 512] X 4	[1 X 1 , 128; 3 X 3 , 128; 1 X 1 , 512] X 4	[1 X 1 , 128; 3 X 3 , 128; 1 X 1 , 512] X 8
conv4_x	14 X 14	[3 X 3 , 256; 3 X 3 , 256] X 2	[3 X 3 , 256; 3 X 3 , 256] X 6	[1 X 1 , 256; 3 X 3 , 256; 1 X 1 , 1024] X 6	[1 X 1 , 256; 3 X 3 , 256; 1 X 1 , 1024] X 23	[1 X 1 , 256; 3 X 3 , 256; 1 X 1 , 1024] X 36
conv5_x	7 X 7	[3 X 3 , 512; 3 X 3 , 512] X 2	[3 X 3 , 512; 3 X 3 , 512] X 3	[1 X 1 , 512; 3 X 3 , 512; 1 X 1 , 2048] X 3	[1 X 1 , 512; 3 X 3 , 512; 1 X 1 , 2048] X 3	[1 X 1 , 512; 3 X 3 , 512; 1 X 1 , 2048] X 3
	1 X 1	average pool , 1000-d fc , softmax				
FLOPs		$1.8 X10^9$	$3.6 X10^9$	$3.8 X10^9$	$7.6 X10^9$	$11.3 X10^9$

Table 3.2 ResNet network structure with different depths

Considering that in the process of network degradation, the external layer network is trained better than the deep layer network, at this time, passing the features from the lower layer to the higher layer gives a better network than if only the shallow layer is used for training.ResNet adopts precisely this strategy and is able to learn even 152 layers of network structure, about 8 times VGG19, but with low complexity and easy optimization. The structure diagram of ResNet-50 is shown in Figure. The structure of ResNet-50 is shown in Figure 3.3.

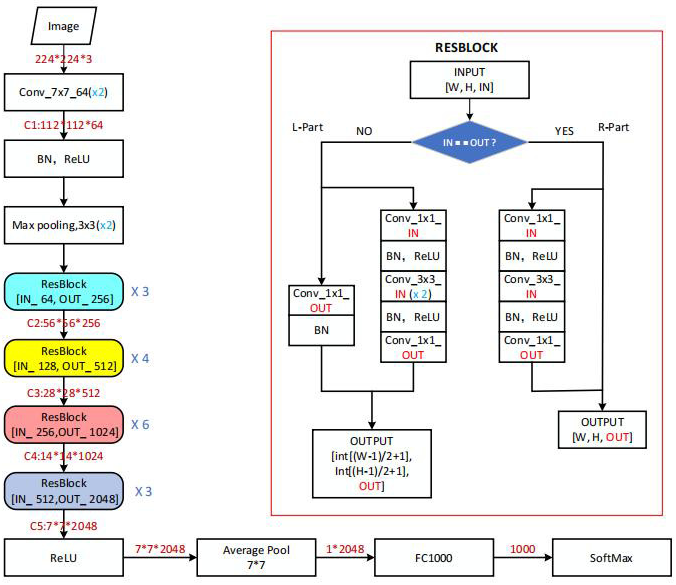

Fig 3.2.1 ResNet-50 Architecture

3.3 ML-NET

ML-NET is a model of neural networks with fully convolutional layers, using the features of different layers in order to efficiently predict saliency maps. This model also uses the concept of prior, which is a way to define regularities in visual perception. In their model, it is fully learned and integrated, at the last stage, with the features of the image extracted. The basic framework for ML-NET is shown in figure 3.3.

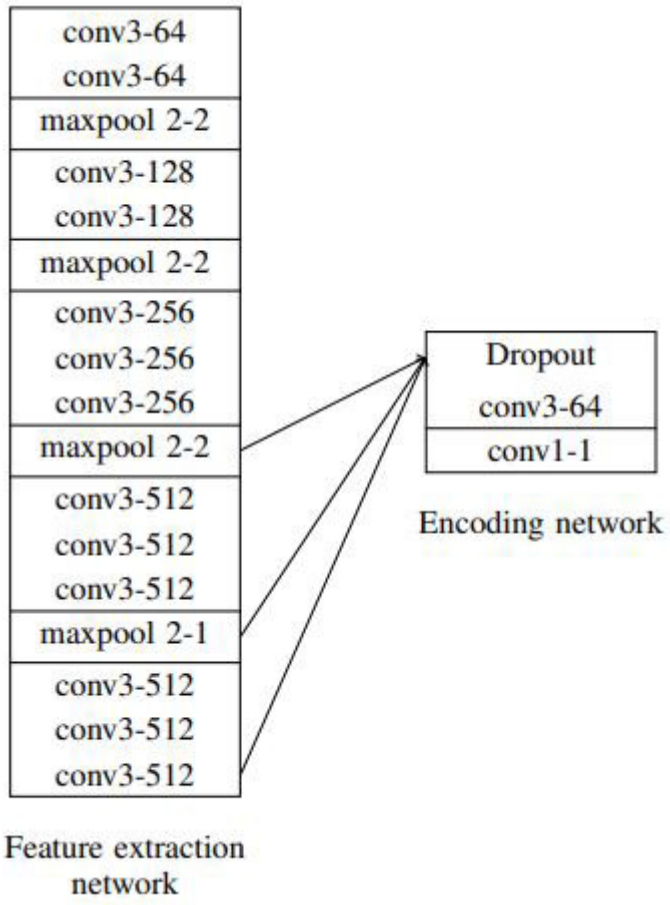

Figure 3.2.1 ML-Net Architecture

3.4 Loss Function - KLLoss

KL scatter, also called relative entropy, is used to measure the distance between two distributions (discrete and continuous). Let p(x) and q(x) be two probability

distributions of a discrete random variable X; then the KL scatter of p to q is:

$$D_{KL}(p\|q) = E_{p(x)} \log \frac{p(x)}{q(x)} = \sum_{i=1}^{N} p(x_i) \cdot (\log p(x_i) - \log q(x_i))$$

For the batch data D(x, y) containing N samples, x is the output of the neural network and is normalized and logarithmic; y is the actual label (default is a probability), and x is in the same dimension as y. The loss value L for the nth sample is calculated as follows:

$$l_n = y_n \cdot (\log y_n - x_n)$$

3.5 Noise

3.5.1 Gaussian noise

Gaussian noise is the most commonly used image noise model and is mathematically tractable in both the spatial and frequency domains. The expression of the probability density function of Gaussian noise is:

$$p(z) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(z-\mu)^2}{2\sigma^2}} \quad (1)$$

μ is the expected value of z, σ and σ^2 denote the standard deviation and variance of z, respectively. The probability density function curve of Gaussian noise is shown in figure 3.9.1

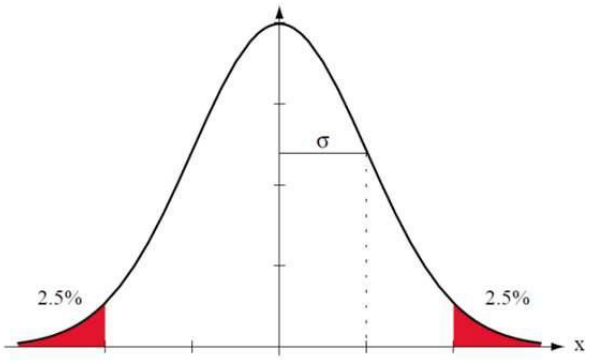

Fig 3.5.1 The probability density curve of Gaussian noise

We can obtain that *z* obeys the Gaussian distribution. There is a 70% probability that its value will fall in $[(\mu - \sigma),(\mu + \sigma)]$ and have 95% probability that its value will fall in $[(\mu - 2\sigma),(\mu + 2\sigma)]$.

3.5.2 Impulse noise

There are many kinds of impulse noise, one of them is Salt And Pepper Noise, which uses minimum or maximum intensity, assuming 8 bits per pixel, and the noise pixels can only be 0 or 255; the visual effect of this noise is similar to sprinkling white and black dots on the image, and its probability density function is expressed as follows:

$$p(z) = p_a \delta(z - a) + p_b \delta(z - b) \qquad (2)$$

The probability density curve of the impulse noise is shown in Figure 3.5.2

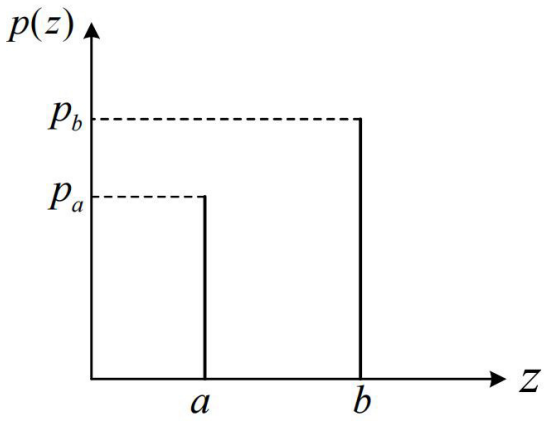

Fig 3.5.2 Probability density curve of the impulse noise

If b > a the gray value b will appear as a bright spot in the image, and the gray value a will appear as a dark spot in the image; if p oor PB is zero, the impulse noise is called a unipolar pulse; if p_a = 0, only bright spot noise exists, which is called salt noise (positive pulse); if PB = 0, only dark spot noise exists, which is pepper noise (negative pulse).

3.5.3 Poisson Noise

Since light has quantum effects, there is a statistical rise and fall in the number of quanta reaching the surface of the photodetector. Therefore, image monitoring has a granularity, which causes the image contrast to become smaller and the image detail information to be obscured; we call this measurement uncertainty due to light quanta as Poisson noise of the image. Like gaussian noise, the probability density function curve of Poisson noise is shown in Figure:

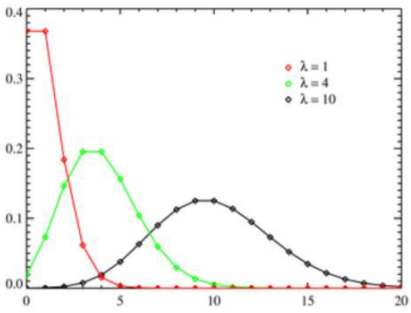

Fig 3.5.3 Probability density curve of the possion noise

3.5.4 Speckle noise

Unlike the gaussian noise, the speckle noise does not follow a normal distribution, and it frequently occurs in medical imaging. The imaging which affected by speckle noise can be represented by:

$$I_n = I\gamma_m \quad (2)$$

3.6 Datasets

SILICON database: this is the most extensive database available for gaze point detection [12], which contains a total of 10,000 training images and 5,000 validation set images, and 5,000 test images for gaze point detection, all selected from the Microsoft CoCo database [36]. The authors of the SILICON database propose a mouse. The authors of the SILICON database propose random mouse tracking on multi-resolution images as an alternative to eye-tracking so that the gaze points in this database are not recorded by eye-tracking but are simulated by mouse clicks. The database authors demonstrate a high degree of similarity between the results of mouse-based gaze point annotation and oculomotor recordings.

CAT2000 database: This database contains 4000 images in 20 categories, including cartoons, artistic satellite images, low-resolution images, interiors, exteriors, sketches, etc. There are 200 images in each category. The test set and the training set each have 2000 images in the database, and the actual values of the gaze points corresponding to the 2000 images in the test set are also not publicly available. The results need to be uploaded to the MIT saliency benchmark for evaluation.

The MIT saliency Benchmark, mentioned above, is a platform for comparison of

annotation point detection results maintained by Ali borji, Jack Brown, and others. The platform aims to provide an up-to-date online saliency model comparison and database. They evaluate and report the latest saliency detection models, and saliency truth values are kept confidential to prevent training and fitting to specific databases. They also provide other gaze-point datasets and acquisition procedures, as well as all relevant links, so that users get a one-stop resource for easy comparison. The comparison platform uses the following evaluation criteria: Earth Movers Distance (EMD), Normalized Scanpath Saliency (NSS), Similarity (SIM), Linear Correlation Coefficient (CC), The Area under the ROC curve (AUC). According to Bylinskii [39], the evaluation methods can be classified as location-based and distribution-based, depending on whether their actual value is represented by the gaze point location or by a continuous gaze point density map. A brief description of these evaluation methods follows.

3.7 Matrices

3.7.1 The Area under the ROC curve (AUC)

AUC is the most widely used criterion for the significance map. The area under the ROC curve (AUC) is the most widely used criterion for the significance map. When calculating the AUC, the significance map is used as a binary classifier to separate positive and negative samples under different thresholds. Samples at different thresholds. The ROC curve is then plotted using the positive and negative favorable rates at different thresholds, and the AUC is the area under the curve. Many researchers have designed various AUCs, including AUC-borji, AUC-Judd, and Stuffed AUC. The difference between AUC-borji and AUC-Judd lies in the way of calculating positive and negative favorable rates. Stuffed AUC aims to eliminate the effect of center bias [40].

3.7.2 Normalized Scanpath Saliency

Normalized Scanpath Saliency (NSS): the NSS is a simple way to calculate the correlation evaluation of the saliency map with the actual value [41]. It calculates the average normalized saliency of the gaze points. Given a significant graph
saliency map P and a binary true value map QB labeled with the attention points.

$$NSS(P,Q^B)=\frac{1}{N}\sum_i \overline{P}_i \times Q_i^B$$

$$N=\sum_i Q_i^B, \overline{P}=\frac{P-\mu(P)}{\sigma(P)}$$

3.7.3 Earth Movers Distance

Earth Movers Distance (EMD): None of the evaluation criteria we have discussed so far evaluates how far the predicted values are from the actual values in terms of spatial distance. EMD evaluates the distribution distance between the predicted and actual images by calculating the minimum cost required for one distribution to match the other. The EMD evaluates the distribution distance between the predicted image and the actual image by calculating the minimum cost required for one distribution to match the other. The calculation is as follows：

$$EMD(P,Q^D)=(\min_{\{f_{ij}\}}\sum_{i,j} f_{ij}d_{ij})+|\sum_i P_i-\sum_j Q_j^D|\max_{i,j} d_{ij}$$
$$s.t.\ f_{ij}\geq 0, \sum_j f_{ij}\leq P_i, \sum_i f_{ij}\leq Q_j^D, \sum_{i,j} f_{ij}=\min(\sum_i P_i, \sum_j Q_j^D)$$

3.7.4 Linear Correlation Coefficient (CC)

Linear Correlation Coefficient (CC): The linear correlation coefficient is also known as the Pearson linear correlation. It measures the linear correlation coefficient between the prediction and actual value plots. Using P and QD to represent the two plots, respectively. Then CC can be calculated using the following equation.

$$CC(P,Q^D)=\frac{\sigma(P,Q^D)}{\sigma(P)\times\sigma(Q^D)}$$

3.8 Gradient descent

Gradient descent is a common first-order optimization method, which is one of the simplest and most classical methods for solving unconstrained optimization problems.

3.8.1 Batch gradient descent

Batch gradient descent is a common form of gradient descent that uses the entire training sample set to compute the gradient of the cost function $l(\theta)$ with respect to parameter θ and then updates the parameters.

$$\theta=\theta-\eta\cdot\nabla_\theta\ell(\theta)$$

where η is the learning rate and $\nabla_\theta\ell(\theta)$ denotes the gradient of the function $\ell(\theta)$ with respect to the parameter θ . Batch gradient descent uses the entire training set in each iteration. Therefore, it can be updated in the right direction, and eventually convergence to the extreme value point is guaranteed. However, the same F is slow to update iterations due to the large amount of data used, which puts a certain pressure on memory and computation.

3.8.2 Stochastic gradient descent

Stochastic gradient descent considers a randomly selected training sample *xi* and label *yi* from the training sample set in each iteration to perform the parameter update.

$$\theta=\theta-\eta\cdot\nabla_\theta\ell(\theta;x_i;y_i)$$

Batch gradient descent and random gradient descent are two extremes: one uses all training samples; the other uses one sample for gradient descent. Naturally, their advantages and disadvantages are very prominent. In terms of training speed, stochastic gradient descent is very fast, while batch gradient descent is unsatisfactory when the training sample set is large. In terms of accuracy, stochastic gradient descent

uses only one sample to determine the direction of the gradient, which may not be the optimal direction of descent. In terms of convergence speed, since stochastic gradient descent considers only one sample per iteration, the gradient direction is highly variable and does not converge to the local optimal solution quickly. However, in terms of computational speed, there is no doubt that stochastic gradient descent is faster.

3.8.3 Mini-batch gradient descent

Mini-batch gradient descent is a compromise between batch gradient descent and random gradient descent, which performs updates using small batches of N randomly sampled training samples.

$$\theta = \theta - \eta \cdot \nabla_\theta \ell(\theta; x_{(i:i+N)}; y_{(i:i+N)})$$

where N is the number of batches. Small batch gradient descent has a more accurate update direction, i.e., a more stable convergence. In addition the highly optimized matrix optimization algorithm, which exists in the advanced deep learning library, can be used to efficiently compute the gradient of small batches.

3.8.4 Adaptive Momentum Estimation

The Adam (Adaptive Momentum Estimation) algorithm [32] is one of the most mainstream algorithmic optimizers available. It takes into account both the momentum variable Vt and the exponentially weighted moving average variable Sy of small batch random gradients squared by elements. The momentum Vt update formula is shown below:

$$v_t = \beta_1 v_{t-1} + (1-\beta_1)\nabla_\theta \ell(\theta; x_{(i:i+N)}; y_{(i:i+N)})_t$$

The variable St update formula is shown below:

$$s_t = \beta_2 s_{t-1} + (1-\beta_2)\nabla_\theta \ell(\theta; x_{(i:i+N)}; y_{(i:i+N)})_t \otimes \nabla_\theta \ell(\theta; x_{(i:i+N)}; y_{(i:i+N)})_t$$

The update formula of θ is shown below:

$$\theta_t = \theta_{t-1} - \frac{\eta \frac{v_t}{1-\beta_1^t}}{\sqrt{\frac{s_t}{1-\beta_2^t}} + \tau}$$

where τ is the parameter added to maintain the stability of the coefficients and keep the denominator from being zero, usually 10^{-8}. θ is the parameter at the *t-th* update. β_1 and β_2 are hyperparameters with values in the range [0,1), usually β_1 is set to 0.9 and β_2 is set to 0.999. the Adam algorithm incorporates the ideas of gradient descent, momentum (SGDm) [33], Adagrad [34], and RMSProp with minor improvements. It has the advantages of simplicity, small memory requirement, insensitivity to gradient isometric scaling, large data size, sparse data handling, and easy hyperparameter tuning.

Chapter IV.

Results and Analysis

In this chapter, I will perform different experiments with different models to see the performance of CAT2000 datasets. Then I will apply the three different noises to the datasets, followed by the experiments with noised data.

4.1 Complexity of Model

There exist two complexity for deep learning models. The time complexity determines the training/prediction time of the model. If the complexity is too high, it results in a time-consuming model training and prediction, which neither allows for fast idea validation and model improvement, nor for fast prediction. The spatial complexity determines the number of parameters of the model. Due to the limitation of the curse of dimensionality, the more parameters a model has, the larger the amount of data required to train the model, and real-life data sets are usually not too large, which can lead to more overfitting of the model training. In this section, the complexity of all three Models will be given by using PyTorch library.

```
from torchsummary import summary
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
model = ResNet50().to(device)
summary(model, input_size=(3, 540, 960))
```

Fig 4.1 Complexity of Model code

```
Total params: 16,710,019
Trainable params: 16,710,019
Non-trainable params: 0
```

```
Total params: 15,029,939
Trainable params: 15,029,939
Non-trainable params: 0
```

```
Total params: 15,452,097
Trainable params: 3,097,217
Non-trainable params: 12,354,880
```

Fig 4.1.2 Number of Parameters of ResNet(left), VGG(middle),ML-NET(right)

As we can see, the ResNet have the highest parameters. In general, the more complex the model is, the better the results perform. This is because a complex model with multiple layers can extract deeper features in the image and optimize the results.

Therefore, in terms of model complexity, the results of resnet should be the best for the same conditions.

4.2 Model Performance with Original Data

In this section, I applied different loss functions in different models. The result is shown in the following tables.

Model	Noise	KLD	NSS	SIM
VGG16	None	0.77	0.71	0.67
ResNet50	None	1.54	0.7	0.59
ML-NET	None	0.43	0.23	0.71

Table 4.2 Overall results for 3 models

We can see for the KLD value, the ML-NET outperform others, since KLD measures a given arbitrary distribution is away from the true distribution, the reason that ML-NET performs the best, is may because the model is more complex than any others. The similarity performance measured by metrics are nearly the same. However, the NSS shows the completely different result. The NSS result of ML-NET shows that the ML-NET performance is behind from the frequency of evaluation, while the VGG and ResNet did not have significant difference. The following figure shows the visualization of the result.

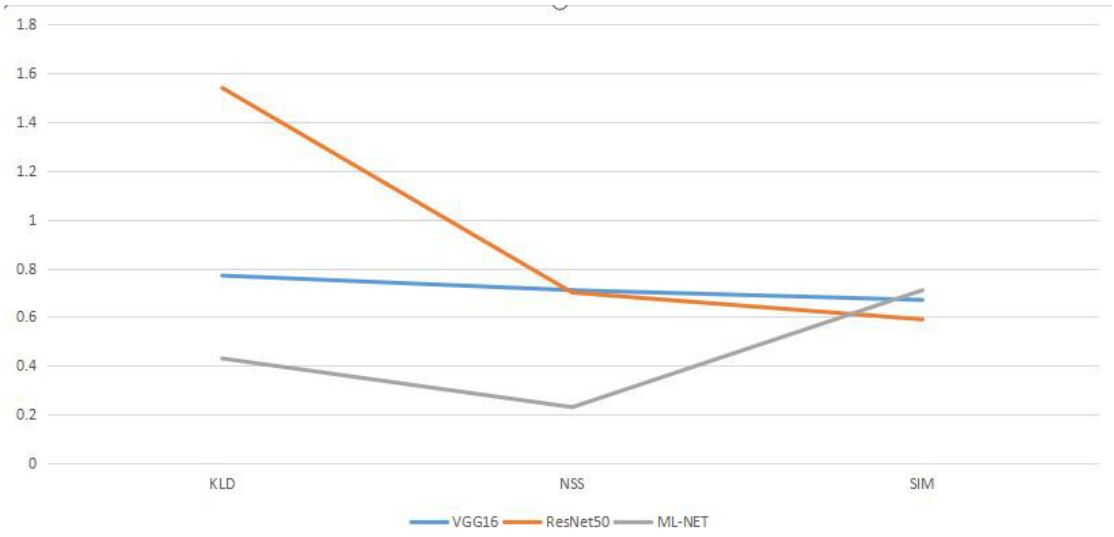

Fig 4.2 The line plot for the results.

In more detail, it can be seen here that ML-NET results are the best overall. Starting with the model structure, first comparing ResNet and VGG, it can be seen that the results of VGG are better than those of ResNet. One perspective is that VGG naturally performs better than ResNet in the saliency detection task, but overfitting due to the more complex structure of ResNet cannot be ruled out. Then compare VGG and ML-NET, ML-NET is based on VGG with an added layer of priors for sampling. It can be seen that the results are significantly improved after adding a layer of priors. Therefore, it can be considered that the prior layer can play a role in saliency detection to promote better results.

4.3 Model Performance with Noised Data

Image noise is the random signal disturbance that an image is subjected to when it is ingested or transmitted, manifested as random variations in image information or pixel brightness. In this section, we use the gaussian noise, poisson noise and speckle noise will be applied to all models. After the all the three noises applied to all data. We can see the photo became unclear as seen in following figure.

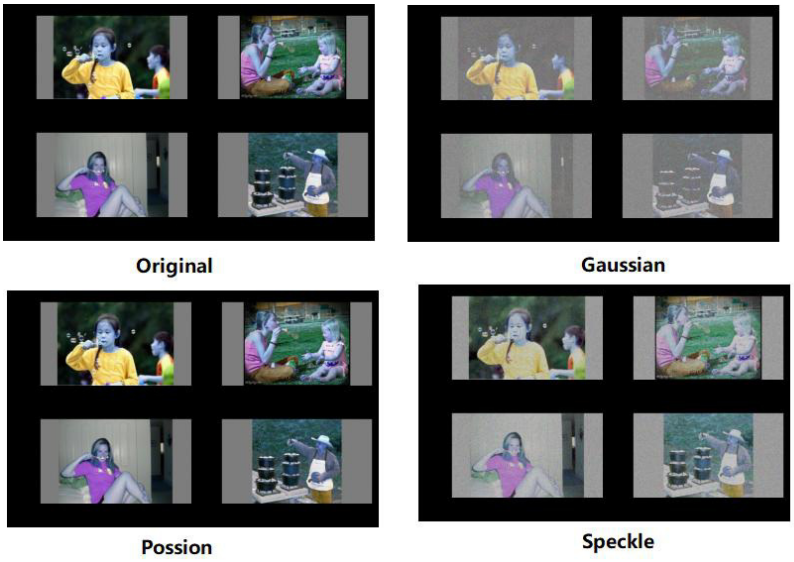

Fig 4.3 Different noises

The result for Gaussian noise is shown in the following table

Model	Noise	KLD	SSIM	NSS
ResNet50	Gaussian	1.54	0.31	0.59
VGG16	Gaussian	1.62	0.28	0.61
ML-NET	Gaussian	0.96	0.33	0.21

Table 4.3.1 Gaussian noise results for 3 models

It can be seen that the performance of all three models drops significantly after adding Gaussian noise. Overall, the performance results ML-NET are consistent with the results without adding noise points.

Model	Noise	KLD	SSIM	NSS
ResNet50	Poisson	0.73	0.55	0.73
VGG16	Poisson	0.82	0.57	0.70
ML-NET	Poisson	0.42	0.54	0.24

Table 4.3.2 Poisson noise results for 3 models

After adding Poisson, the performance of all three models decreases, but not as much as guassian. The performanc shows that VGG16 and ResNet50 have better resolution of poission noise than ML-NET. There are few reasons my cause the difference between possion noise and Gaussian noise. Firstly, In raw image, the main noise is two kinds, Gaussian noise and scattered noise, among which, Gaussian noise is the noise that has no relationship with light intensity, and the average level of noise (generally 0) remains the same regardless of the pixel value. The other is scattered noise, because it conforms to Poisson distribution, also known as Poisson noise, Poisson noise increases with light intensity, the average noise also increases. While ML-NET adds sampling at the last layer of the model structure, this may cause ML-NET to be more sensitive to the noise generated by light intensity, resulting in a decrease in accuracy. Secondly, by deepening the network structure, the network is able to extract more abstract higher-order features, and VGG16 finally extracts a 512-channel feature map, which means it have strong ability to deal with possion

noises.

Model	Noise	KLD	SSIM	NSS
ResNet50	Speckle	2.46	0.21	0.42
VGG16	Speckle	2.23	0.21	0.65
ML-NET	Speckle	0.65	0.42	0.22

Table 4.3.3 Speckle noise results for 3 models

After adding speckle noise, the performance of the three models also dropped significantly. The decrease of ResNet and VGG16 is especially obvious, while the decrease of ML-NET is not particularly large. This indicates that ML-NET has an advantage in the resolution of speckle noise.

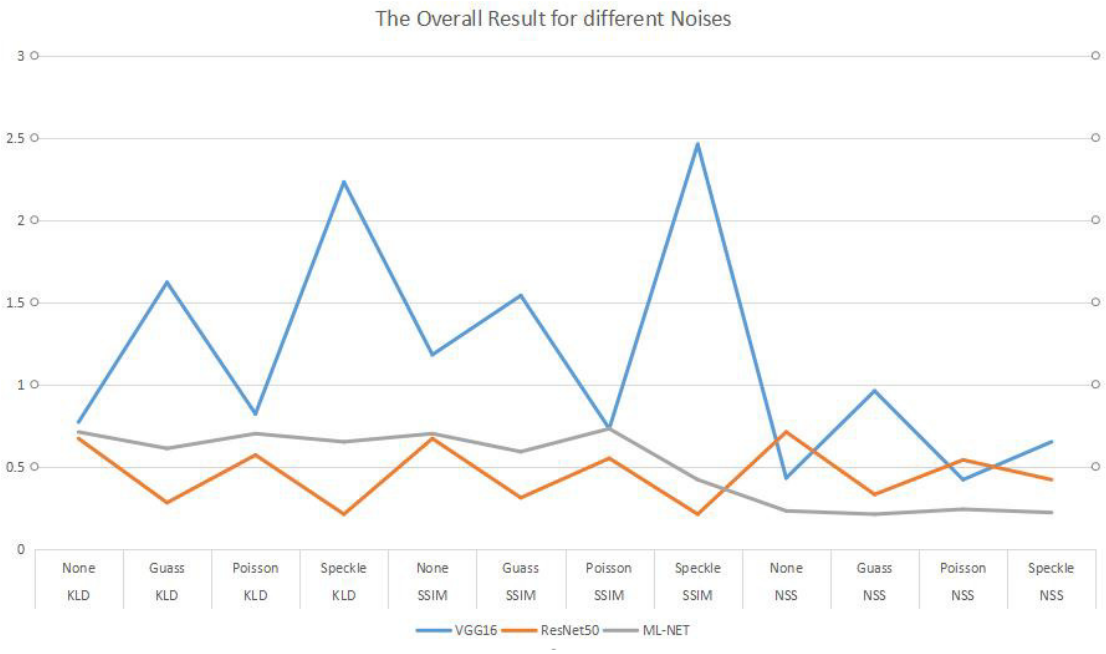

Fig 4.3.2 The line plot for the overall result with all noises.

As we can see, the ResNet50 model have strong generalization ability since we can see there is no significant difference on matrices with different noises. The VGG16 may performs very well in specific metrics, but the noises have huge impacts to the performance of VGG16. The ML-NET also shows good generalization ability. In

summary, firstly, the performance of each of the three models decreases after adding noise, which shows that adding noise does not improve the performance of the saliency detection task for these three models. Secondly, the ability of impact among 3 noises adding method are poisson>gaussian>speckle. Figure 4.3.2 also shows that poisson noise has the least impact on the image.

Chapter V.

Conclusion

The advantages of visual target tracking technology in handling computer vision tasks are becoming increasingly apparent, not only in a variety of It has a wide range of applications not only in various fields, but can also be used to analyse some advanced semantic information. Although visual tracking techniques have contributed greatly to the advancement of computer vision technology, the current tracking techniques are not perfect and there are still many problems. One of these problems is the generalisation of the model. In this task, we use different methods to test the generalisation ability of the model.

By applying different noises to different deep learning models to recognise salient objects, the results showed a decrease in overall performance. Specifically, ResNet was not affected by noise, while its performance was relatively poor compared to the other models. Therefore, we can conclude that if a model has better generalisation capabilities, its performance will be sacrificed.

From the final experimental as well as detection results, the noise model generalisation capability test method proposed in this paper has good results in tracking target attention as well as detecting joint attention, but there are still some issues that need to be optimised as follows:

(1) ResNet used in this paper has some advantages in dealing with blurred targets and image background occlusion, but the ML-NET results are lacking irreverently.

(2) When using the noise addition method, the intensity is not taken into account and will be bu'z in the subsequent work

(3) The practical applications are lacking. This paper mainly focuses on algorithm research in the field of visual target tracking, and when deployed to servers and some edge devices, it is currently only applied to smaller facilities due to the lack

of performance of the used devices. In addition, due to the hardware computing power, the recognition effect is poor for large-scale scenarios.

In view of the above problems, the main future research direction work of this paper are:

(1) In terms of visual target tracking speed, the algorithm needs to be further optimized, and some improvements should also be made for fast-moving targets to ensure that the target location can be accurately located and the visual attention of the target can be focused.

(2) Adjustment of different intensity of noise for testing.

(3) In terms of practical applications, experiments on servers and edge devices need to be strengthened to solve the problem of visual tracking detection in large-scale scenarios.

Chapter VI.

Reflection

During my studies in the field of deep learning, I gained a lot of knowledge that is different from machine learning and computer vision. As the basic concepts are based on mathematics, at the beginning of this project, I studied the basic concepts of MLP (multi-layer perceptron) and I figured out what is forward and backward propagation. I also bought a book called "Deep Learning" by Ian Goodfellow. the most important advantage of this book is that it explains some of the processing methods of deep learning from the principles section, with an extensive list of some standard formulas and specific derivations. The first author of the book, Ian Goodfellow, was a former student of Andrew Ng, and in an interview with Ng, Goodfellow said that it was his taking a class from Ng that sparked his interest in deep learning.

After I understood the basic concepts of deep learning, I read the deep learning code and tried to run some basic models myself. By debugging and running the code, I gained a better understanding of what I had learnt and familiarised myself with how I should apply the basics in practice. In addition, I read some classic papers such as LeNet, AlexNet, VGG, etc. in order to get a more accurate understanding of how the relevant model or method is implemented. All the previous steps were in preparation for this project. What I have learnt has been fully used during the time I have been working on this project. Last but not least, I also learnt how to handle images with different noises and gained experience that I had never had before.

Reference

[1] BYLINSKII Z, JUDD T, BORJI A, et al. MIT saliency benchmark[EB/OL]. 2015, http://saliency.mit.edu/

[3] ITTI L, KOCH C, NIEBUR E. A model of saliency-based visual attention for rapid scene analysis[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 1998, 20(11):1254-1259.

[4] ITTI L, BORJI A. Exploiting local and global patch rarities for saliency detection[C]. Computer Vision and Pattern Recognition. IEEE, 2012:478-485.

[5] HOU X, ZHANG L. Dynamic visual attention: searching for coding length increments[C]. Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December. DBLP, 2008:681-688.

[6] CERF M, HAREL J, EINHÄUSER W, et al. Predicting human gaze using low-level saliency combined with face detection[J]. Advances in Neural Information Processing Systems, 2008, 20:241-248.

[7] BORJI A. Boosting bottom-up and top-down visual features for saliency estimation[C]. IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2012:438-445.

[8] SHEN C, HUANG X, ZHAO Q. Predicting eye fixations on webpage with an ensemble of early features and high-level representations from deep network[J]. Multimedia IEEE Transactions on, 2015, 17(11):2084-2093.

[9] VIG E, DORR M, COX D. Large-scale optimization of hierarchical features for saliency prediction in natural images[C]. Computer Vision and Pattern Recognition. IEEE, 2014:2798-2805.

[10] LIU N, HAN J, ZHANG D, et al. Predicting eye fixations using convolutional neural networks[C]. IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2015:362-370.

[11] KRUTHIVENTI S S S, AYUSH K, BABU R V. DeepFix: a fully convolutional neural network for predicting human eye fixations[J]. arXiv preprint arXiv:1510.02927, 2015.

[12] JIANG M, HUANG S, DUAN J, et al. SILICON: saliency in context[C]. Computer Vision and Pattern Recognition. IEEE, 2015:1072-1080.

[13] CORNIA M, BARALDI L, SERRA G, et al. A deep multi-Level network for s saliency prediction[J]. arXiv preprint arXiv:1609.01064, 2016.

[14] KRUTHIVENTI S S S, GUDISA V, DHOLAKIYA J H, et al. Saliency unified: a deep architecture for simultaneous eye fixation prediction and salient object segmentation[C]. IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2016:5781-5790.

[15] ITTI L, BALDI P. Bayesian surprise attracts human attention[J]. Vision Research, 2008, 49(10):1295-1306.

[16] GAO D, MAHADEVAN V, VASCONCELOS N. The discriminant center-surround hypothesis for bottom-up saliency[J]. Advances in Neural Information Processing Systems, 2007, 20:497-504.

[17] RAJ R, GEISLER W S, FRAZOR R A, et al. Contrast statistics for foveated visual systems: fixation selection by minimizing contrast entropy[J]. Journal of the Optical Society of America. A, Optics, Image Science, and Vision, 2005, 22(10): 2039.

[18] Bruce N D B. saliency, attention and visual search: an information theoretic approach[J]. Journal of Vision, 2009, 9(3):1-24.

[19] ITTI L, KOCH C. Comparison of feature combination strategies for saliency-based visual attention systems[J]. Proceedings of SPIE - The International Society for Optical Engineering, 1999, 3644(1):473-482.

[20] SRIVASTAVA N, HINTON G, KRIZHEVSKY A, et al. Dropout: a simple way to prevent neural networks from overfitting[J]. Journal of Machine Learning Research, 2014, 15 (1):1929-1958.

[21] He K, Zhang X, Ren S, et al. Spatial Pyramid Pooling in Deep Convolutional

Networks for Visual Recognition [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9): 1904-1916.

[22] Tan M, Pang R, Le Q V. EfficientDet: Scalable and Efficient Object Detection[C]. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020:10778-10787.

[23] Krizhevsky A, Sutskever I, Hinton G E. ImageNet Classification with Deep Convolutional Neural Networks [J]. Communications of the ACM, 2017, 60(6): 84-90.

[24] Graves A. Long Short-Term Memory [M]//Graves A. Supervised Sequence Labelling with Recurrent Neural Networks. Berlin, Heidelberg; Springer Berlin Heidelberg. 2012: 37-45. Conference on Computer Vision and Pattern Recognition (CVPR), 2016:770-778.

[25] Bahdanau D, Chorowski J, Serdyuk D, et al. End-to-end attention-based large vocabulary speech recognition[C]. 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016:4945-4949.

[26] Serhan B, Cangelosi A. Replication of Infant Behaviours with a Babybot: Early Pointing Gesture Comprehension[C]. 2019 Joint IEEE 9th International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob), 2019:157-162.

[27] Grover S, Sidana K, Jain V. Pipeline for 3D reconstruction of the human body from AR/VR headset mounted egocentric cameras[J]. arXiv e-prints, 2021:arXiv:2111.05409.

[28] Pan C, Cao H, Zhang W, et al. Driver activity recognition using spatial-temporal graph convolutional LSTM networks with attention mechanism. IET Intell Transp Syst. 2021; 15: 297– 307.

[29] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. arXiv e-prints, 2014:arXiv:1409.1556.

[30] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[C]. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016:770-778.

[31] Y. Cao et al., "ML-Net: Multi-Channel Lightweight Network for Detecting Myocardial Infarction," in IEEE Journal of Biomedical and Health Informatics, vol. 25, no. 10, pp. 3721-3731, Oct. 2021, DOI: 10.1109/JBHI.2021.3060433.

[32] Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.

[33] Qian N. On the momentum term in gradient descent learning algorithms[J]. Neural networks, 1999, 12(1): 145-151.

[34] Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization[J]. Journal of machine learning research, 2011, 12(7): 2121-2159.

[35] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[36] Zeiler M D , Fergus R . Visualizing and Understanding Convolutional Networks[C] Computer Vision and Pattern Recognition Workshops .2012:234-244.

[37] Qiong Yan, Li Xu, Jianping Shi. Hierarchical Saliency Detection[C]// Computer Vision and Pattern Recognition .2013:1155-1162.