

Master's thesis

**A Study on Interpretability of  
Image Classification Using Deep  
CNNs for Histopathological Images**

**Daisuke Saito**

Division of Electrical and Electronic Engineering  
Graduate School of Engineering  
Mie University

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Background . . . . .	1
1.2	Research Objective . . . . .	2
<b>2</b>	<b>Related Works</b>	<b>3</b>
2.1	Convolutional Neural Networks . . . . .	3
2.2	Segmentation with CNNs . . . . .	6
2.3	CNNs Explanation Technologies . . . . .	8
<b>3</b>	<b>Experiments</b>	<b>13</b>
3.1	Materials and Data set . . . . .	13
3.2	Model Creation with CNNs . . . . .	15
3.2.1	Classification for Glioma Histopathological Images . . . . .	15
3.2.2	Training of Cell Nucleus Segmentation . . . . .	20
3.2.3	CNNs Evaluation . . . . .	26
3.3	Object-Level-Features and Interpretability of CNNs classification . . . . .	29
3.3.1	Mask Image of Large Size . . . . .	29
3.3.2	Visualization . . . . .	30
3.3.3	Object Level Features . . . . .	30
3.3.4	Feature Selection . . . . .	32
<b>4</b>	<b>Result and Discussion</b>	<b>34</b>
4.1	Performance of Classification Model . . . . .	34
4.2	Segmentation . . . . .	34
4.3	Relationship between CNNs Decision and Object-Level-Features . . . . .	43
<b>5</b>	<b>Concluding Remarks</b>	<b>45</b>
5.1	Conclusion . . . . .	45
5.2	Future Works . . . . .	45
	<b>Acknowledgment</b>	<b>47</b>

*CONTENTS*

ii

**Reference**

**48**

**Publication List**

**51**

## List of Figures

1.1	Pipeline of this Work . . . . .	2
2.1	Convolution Process . . . . .	4
2.2	The Data Size in Convolutional Process . . . . .	4
2.3	Feature Map Generation by Pooling . . . . .	5
2.4	Shortcut-Connection . . . . .	6
2.5	Transposed Convolution . . . . .	7
2.6	Segmentation Architectures . . . . .	9
2.7	DeepLab v3 . . . . .	10
2.8	Prediction with ImageNet Weight and Visualization by Grad CAM++ . . . . .	12
3.1	Obtained Histopathological Images from TCGA . . . . .	14
3.2	Patched Pathological Image and Mask Image of Cell Nucleus Annotation . . . . .	15
3.3	An Example of MoNuSeg Data Set . . . . .	16
3.4	Comparative Fully-Connected-Layer vs. GAP . . . . .	19
3.5	The Process of Mask Image Generation . . . . .	29
3.6	Explanation . . . . .	31
3.7	Important Region and Random Region on the Mask Image . . . . .	32
4.1	Learning Curve of VGG16 Training . . . . .	35
4.2	Learning Curves of All Models (Data Set 1) . . . . .	36
4.3	Learning Curves of All Models (Data Set 2) . . . . .	37
4.4	Learning Curves of All Models (Data Set 3) . . . . .	37
4.5	Bar Graph of All Segmentation Models . . . . .	38
4.6	Heatmap by Grad CAM++ and Mask Image . . . . .	43

## List of Tables

3.1	Experimental Environment for Segmentation . . . . .	16
3.2	Meanings of Symbols in Network Architecture Table . . . . .	17
3.3	Our Classification Model . . . . .	18
3.4	Stage 1 (34-layer) . . . . .	21
3.5	Stage 1 (50-layer, 101-layer) . . . . .	21
3.6	Stage 2 (34-layer) . . . . .	21
3.7	Stage 2 (50-layer, 101-layer) . . . . .	22
3.8	Stage 3 (34-layer) . . . . .	22
3.9	Stage 3 (50-layer, 101-layer) . . . . .	22
3.10	Stage 4 (34-layer) . . . . .	23
3.11	Stage 4 (50-layer, 101-layer) . . . . .	23
3.12	U-Net Architecture . . . . .	24
3.13	U-Net++ . . . . .	25
3.14	DeepLab v3+ . . . . .	27
3.15	Parameters for Training Segmentation. . . . .	28
3.16	Significance Level and $C(\alpha)$ . . . . .	33
4.1	Classification Accuracy for Glioma Pathological Image. . . . .	34
4.2	Confusion Matrix for Learning CNNs with the Most Highest Performance. . . . .	35
4.3	Result of All Segmentation Models . . . . .	38
4.4	Prediction Result I . . . . .	39
4.5	Prediction Result II . . . . .	40
4.6	Prediction Result III . . . . .	41
4.7	Prediction Result IV . . . . .	42
4.8	Result of K-S Test . . . . .	44

# Chapter 1

## Introduction

### 1.1 Background

In the field of digital pathology, medical doctors (pathologists) usually make a final decision (diagnosis) with plenty of given HE stained images. To do this, they first investigate all nuclei in the tissue and extract features from them. However, it will provide much burden to the pathologist. Because the size of the given images is large (approximately several GB), and many nuclei have to be extracted from the given images for diagnosis. The skill of nuclei extraction and their analysis heavily depends on a pathologist's skills and experiences. As a result, this work also will cause a non-quantitative result.

From this background, systems related to computer-aided-diagnosis (CAD) interest actively as the solution to the above problem. In particular, thanks to deep learning-based approaches and the era of Big Data, we can now suggest deep learning-based solutions with input a lot of medical images as Big Data. Convolutional Neural Networks (CNNs), which is one of the Deep Learning technique for computer vision, has been growing quickly. Generally, CNNs automatically extract features, and enough training data leads to high performance for classification or segmentation. The diagnosis with CNNs has expected two main contributions. Firstly, pathologists' work will reduce extremely. Secondly, the criteria will be quantitative. Besides, its technologies provide highly accurate solutions to various medical image classification problems as classification or segmentation and many projects, *e.g.*, [1][2][3][4].

However, in the pathological analysis, there are many scenarios where doctors have to explain the detail of their diagnosis. In similar, constructed CNNs model is required the diagnosis reason even if CNNs' classification ability superior. CNNs architecture is complex to realize high prediction performance, and the all process of feature extraction and classification is automatic. In summary, CNNs middle layer is an almost black box. For this reason, despite the success in automatically classifying disease stages, the CNNs model cannot explain how the classification into each stage happened since feature selection is not apparent. This problem is a big wall to introduce CNNs for medical sites.



## Chapter 2

# Related Works

### 2.1 Convolutional Neural Networks

Currently, CNNs is one of the most popular technologies for classification or segmentation. Its architecture mainly consists of a convolutional layer and a max-pooling layer. Figure 2.1 shows the convolutional process, and this layer uses a filter to compress the features. In CNNs, this layer gives the output as feature map to another convolutional layer or pooling layer. Figure 2.2 explains the convolution process when the image whose size is  $(N, H, W, C)$  is used. In this case, output is  $(N, FN, C, OH, OW)$  through the  $(FN, C, FH, FW)$  filter. Moreover, there are some other convolution parameters such as padding size, stride size and so on. Padding is a process for adding some value on each side of the input feature maps, affecting the output feature size. Stride size means how many pixels are slid for filtering; for instance, the stride size of Figure 2.2 is 1.

On the other hand, the pooling layer is the operation that makes the given image smaller, such as height and width. For example, Figure 2.3 shows the max-pooling of  $2 \times 2$  filter with stride 2. Max pooling means selecting a max value in the region of  $2 \times 2$  filter. Thus pooling layer has no parameter for the training and works for just compressing input data. Through the learning, CNNs tune the parameter of the convolutional filter mainly. Generally, the layer near the input layer extracts low-level features *e.g.*, edge and blob, and so on. In contrast, more deep layers extract high-level features, which means more complex and abstract information.

In the last decade, CNNs study became a large scale due to large public image repositories, such as ImageNet [7], and high-performance computer systems, such as GPUs. Karen *et al.* suggested very deep convolutional neural networks with high accuracy for ImageNet classification [8]. Their model's characters were mainly depth, which has 16 to 19 layers and a smaller filter size than previous architectures. They are called VGGNet. Also, this architecture is more simple than other deep CNNs, like ResNet [9] and Inception-vX [10], so that it is one of the most popular CNNs classification models. On the other hand, [10] explored approaches to scale up GoogLeNet [11] to utilize the

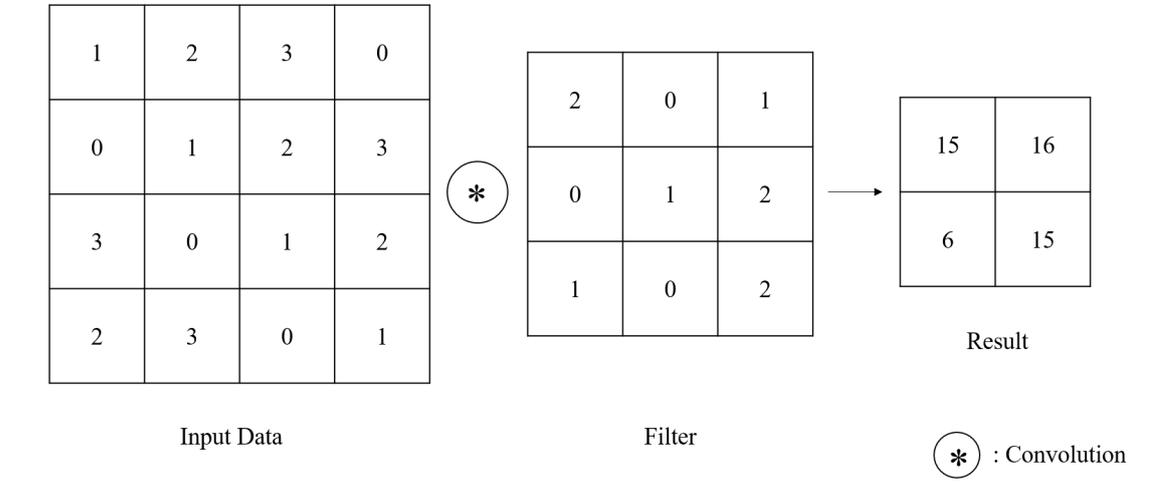


Figure 2.1: Convolution Process

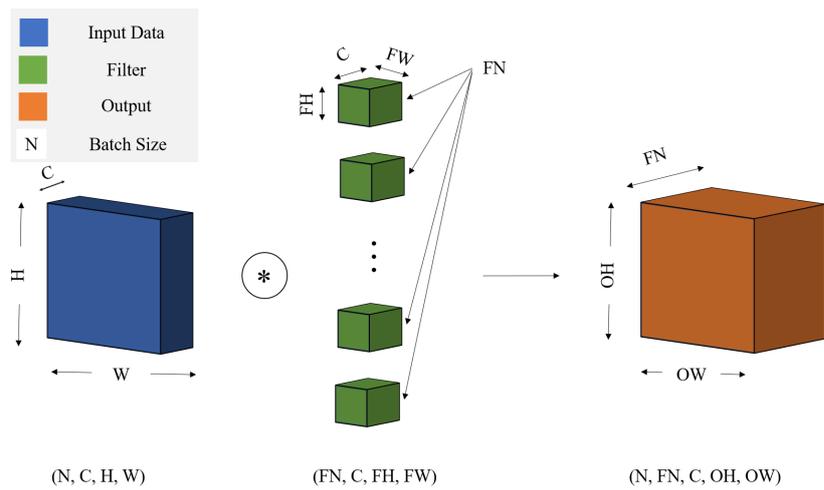
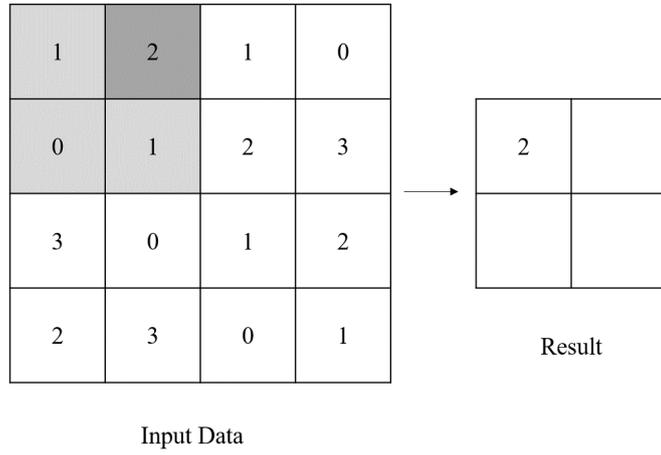
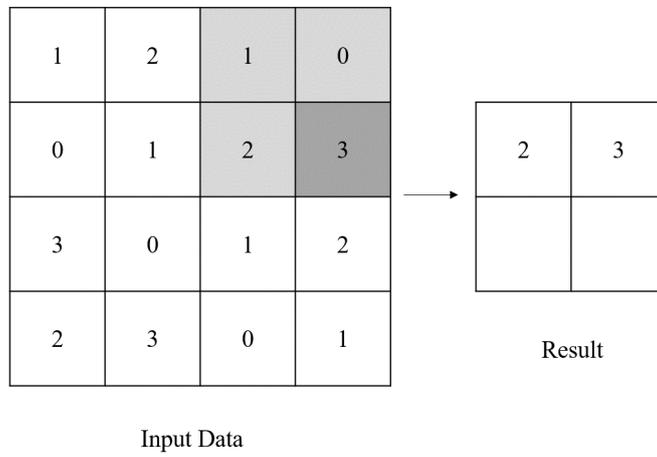


Figure 2.2: The Data Size in Convolutional Process



(a) First Max Pooling



(b) Second Max Pooling

Figure 2.3: Feature Map Generation by Pooling

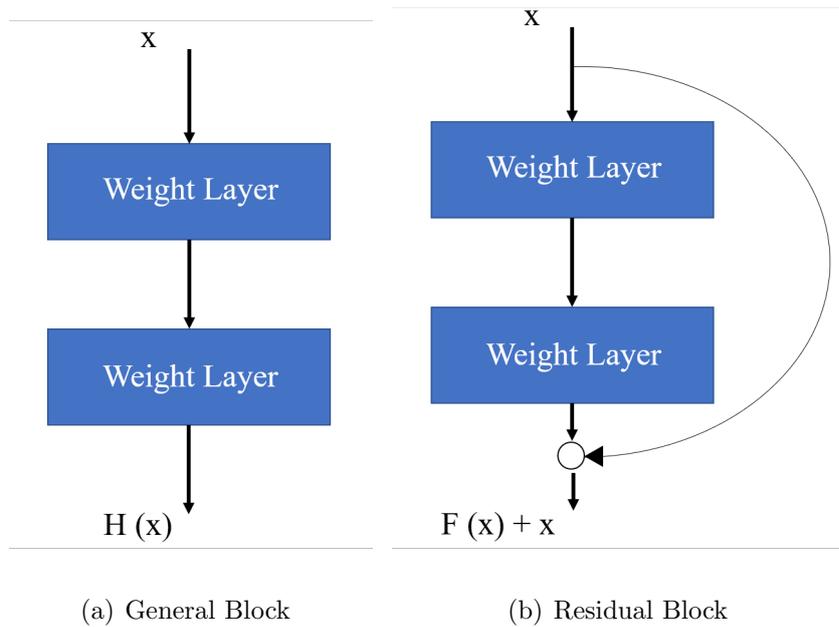


Figure 2.4: Shortcut-Connection

added computation as efficiently. This model employed fewer parameters than VGGNet by several updates, *e.g.*, introducing the Batch Normalization [12] and Global average pooling (GAP) [13]. However, deeper networks start degradation with increasing layers too much, which is overfitting. In [9], residual learning addressed this problem. The main key of this approach was shortcut-connection (Figure 2.4). Shortcut connection integrates the previous compressed weight on the output. If one of them were not needed, its weight would be 0. For this reason, ResNet realized several deeper models, such as ResNet 18, 34, 50, 101, and 152 (Values mean the number of layers). Therefore ResNet are a significant breakthrough for CNNs classification.

## 2.2 Segmentation with CNNs

Image segmentation is one of computer vision techniques to detect objects and make the given image easier for analysis. This thesis applied its technologies to cell nucleus segmentation. Nucleus segmentation is essential for digital pathology because nuclei reflect the current disease stage well. Also, according to Gurcan *et al.*, the features related to cell nuclei are generally useful for evaluating histopathology [5]. For this reason, recently a lot of automatic segmentation methods have been proposed for histopathological image processing, and segmentation techniques with CNNs are one of the hottest topics in this field [14] [15] [16].

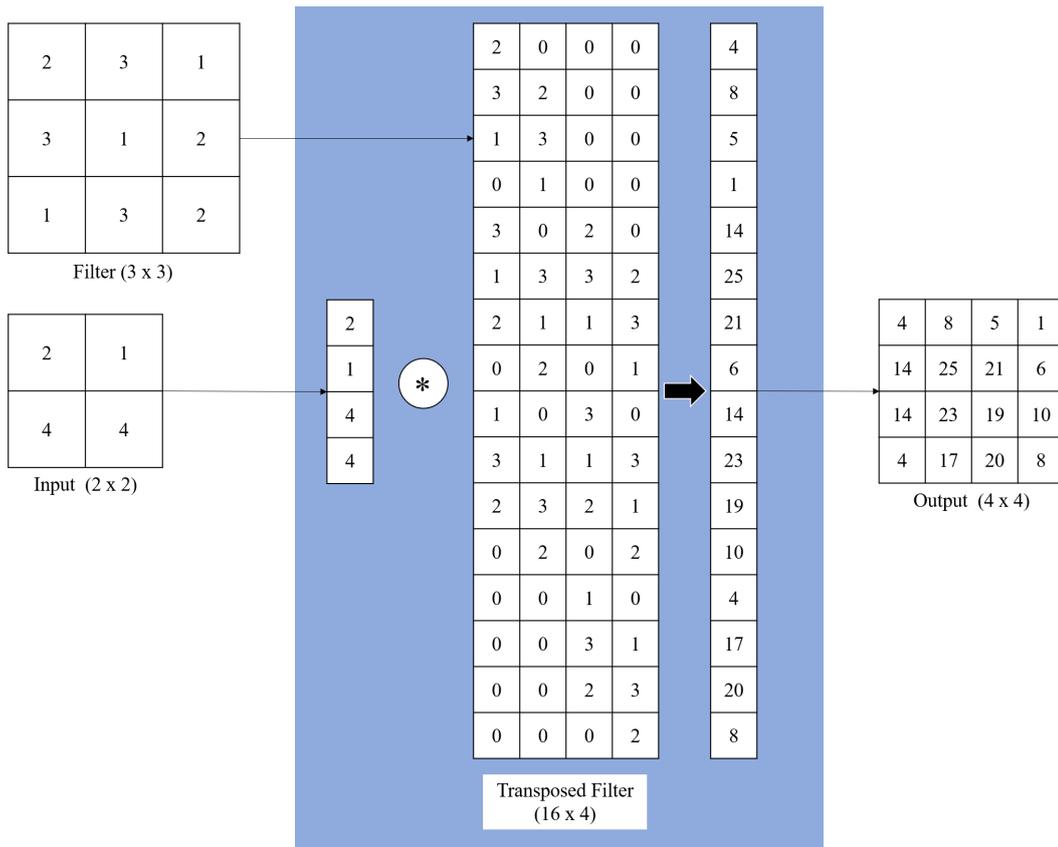


Figure 2.5: Transposed Convolution

In the segmentation process with CNNs, an input image has reduced the size using the convolutional layer and pooling layer. It is called downsampling, and feature maps are generated through the downsampling. Then it needs to convert a feature map to a high-resolution image to recover the “WHERE” information. Transposed convolution is an approach for upsampling an image with learnable parameters. Figure 2.5 shows an example of the transposed convolution process. When the output ( $4 \times 4$ ) is required from the input ( $2 \times 2$ ), then the filter ( $16 \times 4$ ) are used. To do this, the convolutional filter ( $3 \times 3$ ) and input ( $2 \times 2$ ) are converted to ( $16 \times 4$ ) and ( $1 \times 4$ ) with transpose, respectively. The output can be reshaped into ( $4 \times 4$ ) data after the convolutional calculation. U-Net employs a transpose convolutional layer and skip-connection and suggested it for biomedical images. The architectures of U-Net and U-Net++ are shown in Figure 2.6, and the greatest characteristic of U-Net is the shape of the process like “U”. Usually, through the pooling layer, the information of “WHERE” is dropped. In U-Net, skip-connection passes each encode feature map step into each decoding step for fixing the above problem. For instance, X. Li *et al.* proposed a tumor segmentation method for CT scanned images [17].

Furthermore, Zhou *et al.* proposed U-Net++, which was an improved U-Net [16] (Figure 2.6 (b)). The difference with U-Net is decoding from each encoding step. Also, the result of each node is used for prediction and loss calculation. In [16], they applied U-Net++ to various medical images and showed the performance of U-Net++ was good enough compared to other approaches.

Moreover, Chen *et al.* proposed DeepLab v3+, which was an encoder-decoder network with atrous separable convolution for semantic image segmentation in 2018. The encoder module encodes multi-scale contextual information by applying atrous convolution at multiple scales, while the simple yet effective decoder module refines the segmentation results along object boundaries [18].

## 2.3 CNNs Explanation Technologies

A lot of researchers efforts to understand what CNNs learned [19][20]. In [20], Zhou *et al.* proposed methods that various layers of the CNNs behave as unsupervised object detectors using a new technique called CAM (Class Activation Mapping). Then Selvaraju *et al.*[21] came up with an efficient generalization of CAM, known as Grad CAM, which fuses the class-conditional property of CAM with existing pixel-space gradient visualization techniques.

When learning CNNs model with GAP layer predicted  $c$ , which is one of classes, its final classification score  $Y^c$  can be defined using  $A_{ij}^k$  (feature map of GAP layer) by

$$Y^c = \sum_k w_k^c \sum_i \sum_j A_{ij}^k \quad (2.1)$$

where  $A_{ij}^k$  is coordinate  $(i, j)$  of channel  $k$ . Also,  $w_k^c$  expresses importance degree of  $\sum_i \sum_j A_{ij}^k$  to return  $Y^c$ , when  $L_{ij}^c$  is defined as

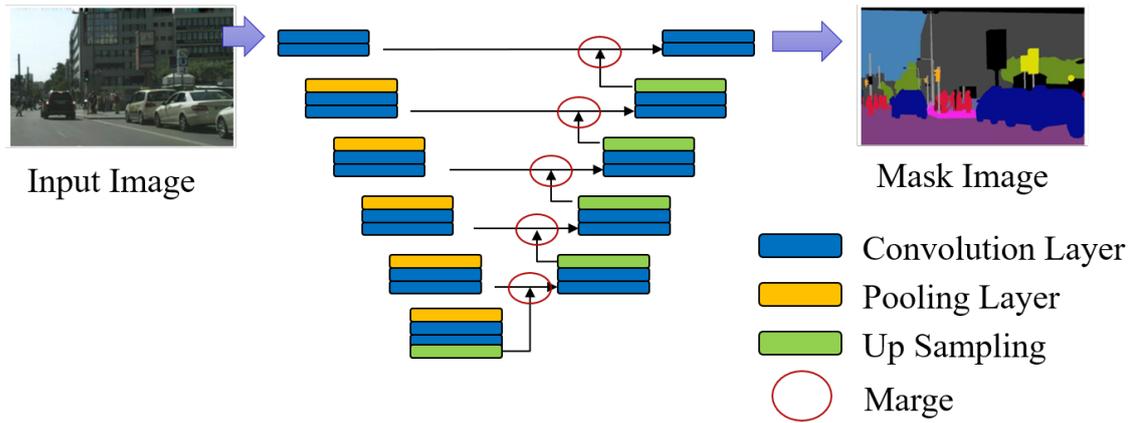
$$L_{ij}^c = \sum_k w_k^c A_{ij}^k, \quad (2.2)$$

then  $Y^c$  can be expressed by

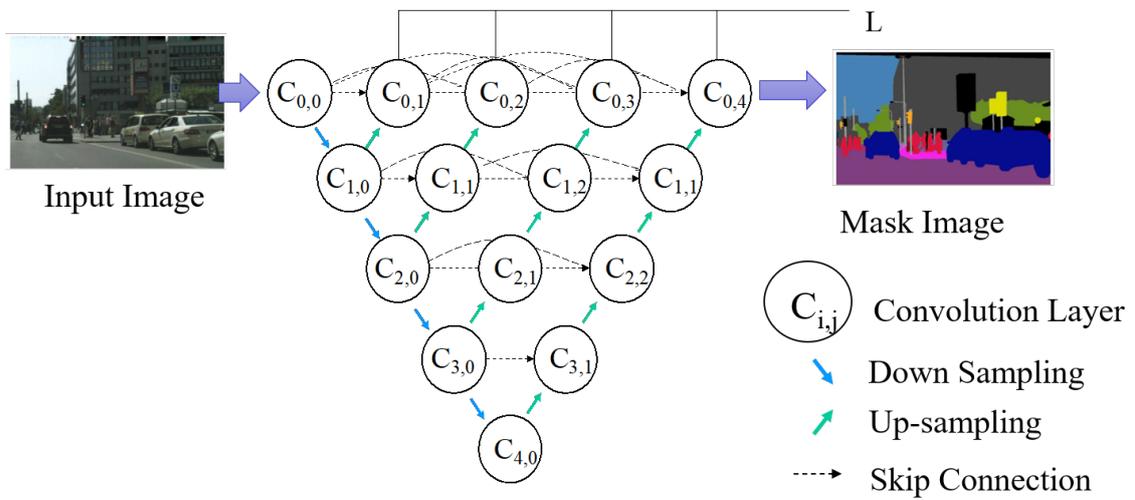
$$Y^c = \sum_i \sum_j L_{ij}^c. \quad (2.3)$$

The defined  $L_{ij}^c$  is important degree when pixel position  $(i, j)$  was decided class  $c$ . Thus to derive  $L_{ij}^c$  means to draw important heatmap of pixel position  $(i, j)$ . In [21], they defined weight  $w_k^c$  for a particular feature map  $A_k$  and class  $c$  as

$$w_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k} \quad (2.4)$$



(a) U-Net



(b) U-Net++

Figure 2.6: Segmentation Architectures

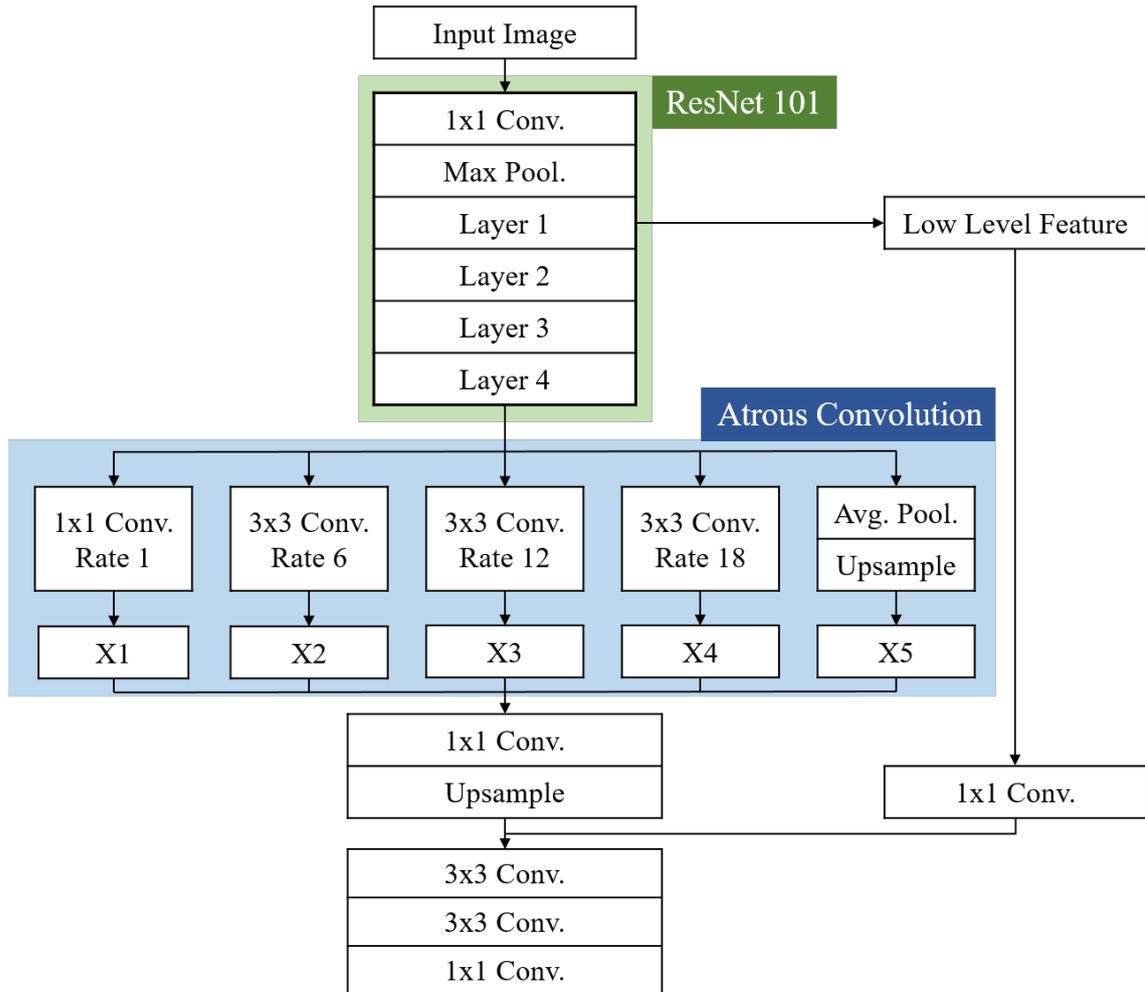


Figure 2.7: DeepLab v3

where  $Z$  is the number of activation map. On the other hand, [6] formulated the structure of the weights  $w_k^c$  as

$$w_k^c = \sum_i \sum_j \alpha_{ij}^{kc} \text{relu} \left( \frac{\partial Y^c}{\partial A_{ij}^k} \right) \quad (2.5)$$

where  $\text{relu}$  is the Rectified Linear Unit activation function. From Eq. (2.1) and Eq. (2.5),  $Y^c$  can be reformulate as

$$Y^c = \sum_k \left\{ \sum_a \sum_b \alpha_{ab}^{kc} \text{relu} \left( \frac{\partial Y^c}{\partial A_{ab}^k} \right) \right\} \sum_i \sum_j A_{ij}^k. \quad (2.6)$$

Here,  $(i, j)$  and  $(a, b)$  are iterators over the same activation map  $A^k$  and are used to avoid confusion. Without loss of generality, Eq. (2.6) drops the  $\text{relu}$  in our derivation as it only functions as a threshold for allowing the gradients to flow back. Taking partial derivative regarding  $A_{ij}^k$  on both sides:

$$\frac{\partial Y^c}{\partial A_{ij}^k} = \sum_a \sum_b \alpha_{ab}^{kc} \frac{\partial Y^c}{\partial A_{ab}^k} + \sum_a \sum_b A_{ab}^k \left\{ \alpha_{ij}^{kc} \frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} \right\} \quad (2.7)$$

Taking a further partial derivative regarding  $A_{ij}^k$ :

$$\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} = 2\alpha_{ij}^{kc} \frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} + \sum_a \sum_b A_{ab}^k \left\{ \alpha_{ij}^{kc} \frac{\partial^3 Y^c}{(\partial A_{ij}^k)^3} \right\} \quad (2.8)$$

from Eq. (2.7) and Eq. (2.8),  $\alpha_{ij}^{kc}$  can be defined by:

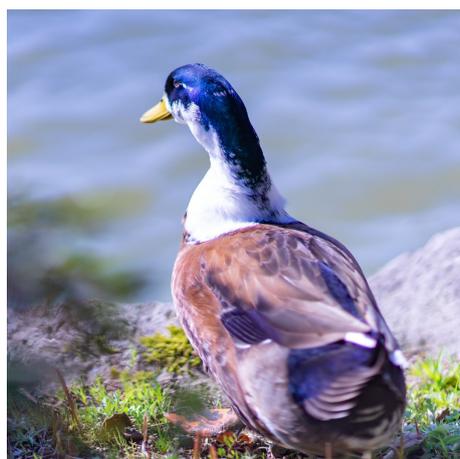
$$\alpha_{ij}^{kc} = \frac{\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2}}{2 \frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} + \sum_a \sum_b A_{ab}^k \left\{ \frac{\partial^3 Y^c}{(\partial A_{ij}^k)^3} \right\}} \quad (2.9)$$

Substituting Eq. (2.9) in Eq. (2.5), Grad CAM++ weights can be obtained as follows;

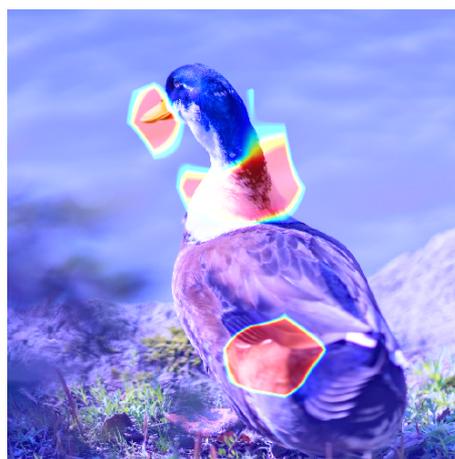
$$w_k^c = \sum_i \sum_j \frac{\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2}}{2 \frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} + \sum_a \sum_b A_{ab}^k \left\{ \frac{\partial^3 Y^c}{(\partial A_{ij}^k)^3} \right\}} \text{relu} \left( \frac{\partial Y^c}{\partial A_{ij}^k} \right). \quad (2.10)$$

If  $\alpha_{ij}^k c = 1/z$  was substituted in Eq. (2.10), Eq. (2.10) becomes the same as Eq. (2.4). Thus unlike Grad CAM, which simply looked at the average of pixels, the weight  $\alpha$  has appeared. By doing this, even small features can be visualized properly.

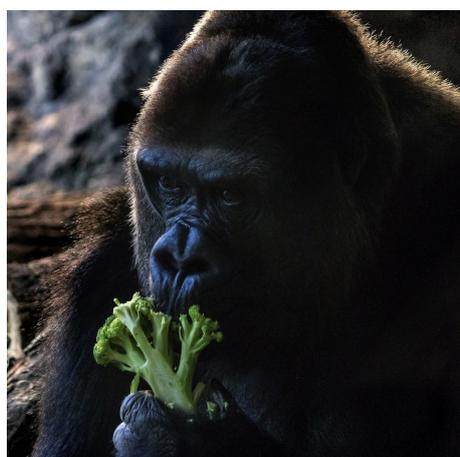
Figure 2.8 shows the examples of Grad CAM++ using ImageNet weight. These heatmaps show the important region as red color. For instance, in (a) and (b), the author can understand that a bill and neck were contributed to the prediction of ptarmigan.



(a) Input Image 1

(b) Obtained Heatmap  
(prediction : ptarmigan)

(c) Input Image 2

(d) Obtained Heatmap  
(prediction : turtle)

(e) Input Image 3

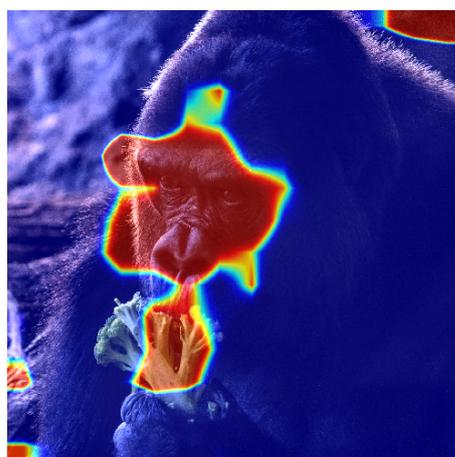
(f) Obtained Heatmap  
(prediction : gorilla)

Figure 2.8: Prediction with ImageNet Weight and Visualization by Grad CAM++

## Chapter 3

# Experiments

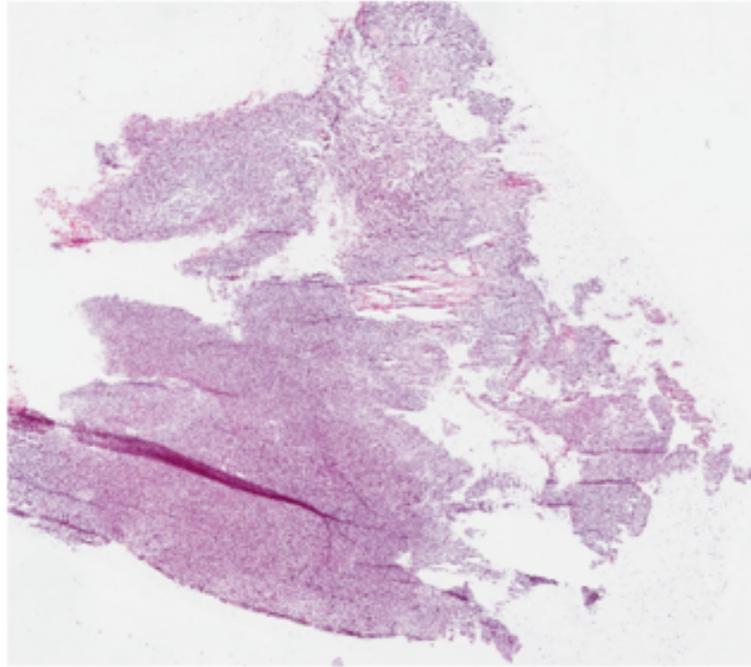
### 3.1 Materials and Data set

In this paper, this thesis focused on Glioma histopathological images, which was one of the brain tumors. Malignant brain tumor has characteristics that precede the stages quickly so that it is crucial to find the disease earlier. Also, Glioma has various types of corresponded for cause and stage. The author put two labels Glioblastoma (GBM) and Low-Grade-Glioma (LGG) on the given images. Glioma histopathological images are available on the official website of The Cancer Genome Atlas (TCGA) [22]. As the experimental material, the author obtained Glioma images from TCGA. Figure 3.1 (a) and (b) show examples of obtained original images.

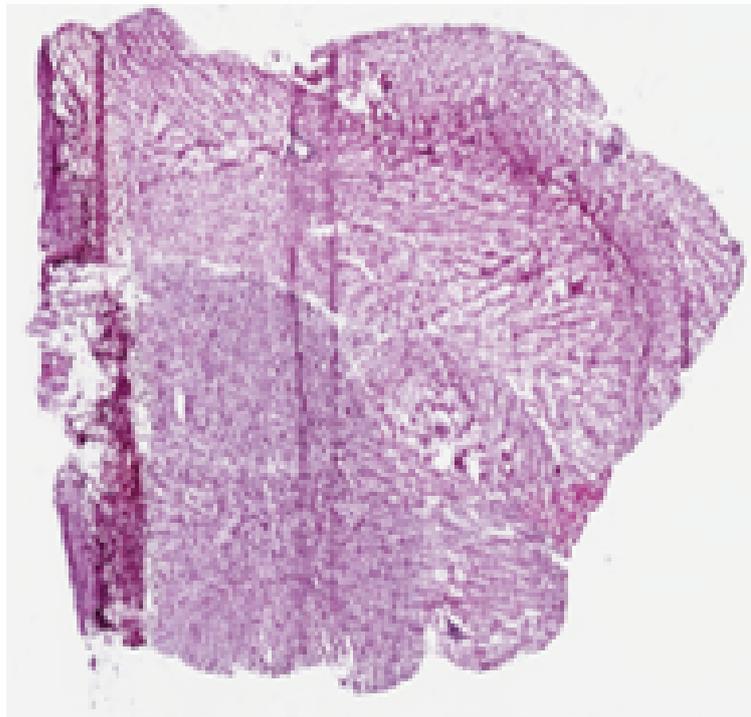
In this work, two CNNs models were created. One of them is the model for the disease stages classification, respectively. The other is for the segmentation model of cell nuclei. This section explains the data set to create two CNNs models in the following paragraphs.

In the model creation for the classification, the first process patched Glioma histopathological images into  $1000 \times 1000$  pixels images because the obtained images from TCGA are too large for learning (Figure 3.2 (a)). This training process also used 20,000 (LGG: 10,000, GBM: 10,000) images that were patched from 604 patients' data. Then this paper partitioned the above images into training (80%), validation (10%), and test (10%) to construct the classification model. Furthermore, two data sets were also prepared in the same proportion for the Cross-Validation.

In addition, this thesis used CNNs for the mask image generation of the cell nuclei. For this reason, the author put cell nuclei annotation into the images for the segmentation. Figure 3.2 (b) shows a mask image of Figure 3.2 (a) with the annotation of cell nuclei. It was put by ourselves for segmentation model construction. However, it was required to convert  $1000 \times 1000$  pixels to a more tiny size because the image size is too large for the segmentation using CNNs. Thus, the pathological images were divided into  $256 \times 256$  pixels. Furthermore, the author used HE and 14 patients' mask images distributed

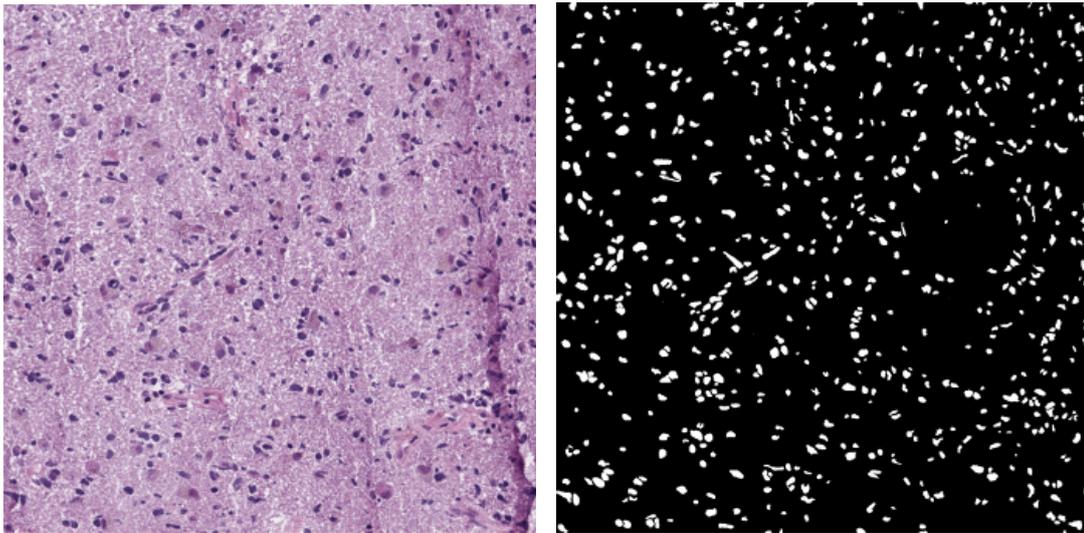


(a) GBM



(b) LGG

Figure 3.1: Obtained Histopathological Images from TCGA



(a) Pathological Image  
(1000 × 1000 pixels)

(b) Annotation Image

Figure 3.2: Patched Pathological Image and Mask Image of Cell Nucleus Annotation

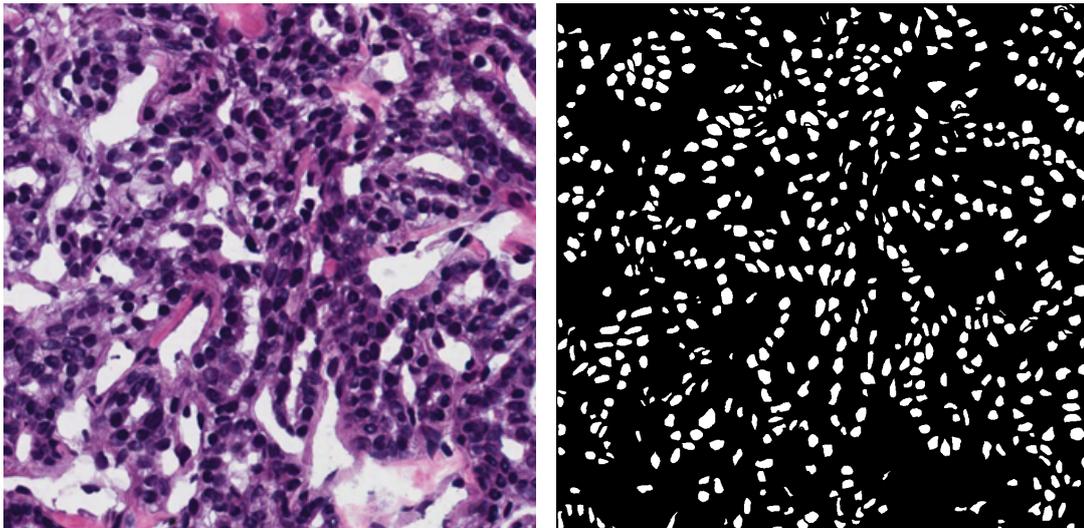
by MoNuSeg [23] as training data set to enhance segmentation accuracy in this work (Figure 3.3). A total of 8,400 images were used for training and test of segmentation. The number of training, validation, and test data were prepared 6400, 1600, and 400, respectively. MoNuSeg image was not used for evaluation. Also, the testing data were prepared considering the patient ID, *i.e.*, the patients used for training were not used for testing.

## 3.2 Model Creation with CNNs

In this study, the above CNNs technologies implemented on the server (Table 3.1) for the classification and segmentation. Also, sections 3.2.1 and 3.2.2 explain CNNs' architectures using symbols of Table 3.2.

### 3.2.1 Classification for Glioma Histopathological Images

In this study, the author constructed a classification model with VGG16[8], which was a kind of VGGNet model, and the VGG16 learned various parameters to classify disease stages for glioma histopathological images. In particular, our model employed five blocks for feature map generation. Block-1 and Block-2 have two convolutional layers, and a max-pooling layer is installed under the convolutional layer. As well as these blocks,



(a) Histopathological Image  
(1000 × 1000 pixels)

(b) Annotation Image

Figure 3.3: An Example of MoNuSeg Data Set

Table 3.1: Experimental Environment for Segmentation

Item	Specification
OS	Ubuntu 18.04.5 LTS
CPU	Intel Core i7 6850K 6core/12thread 3.6GHz
GPU	NVIDIA GeForce GTX 2080Ti 11GB
RAM	32GB (8GBx4)
Storage 1	480GB (Intel SSDSC2BB48)
Storage 2	1TB (WDC WD10JPLX-00M)
Storage 3	1TB (WDC WD10JPLX-00M)
Deep Learning Framework	Keras 2.2.2 (Classification Model) Keras 2.2.4 (Other Experiments)
Language	Python3.5 (Classification Model) Python3.7 (Other Experiments)

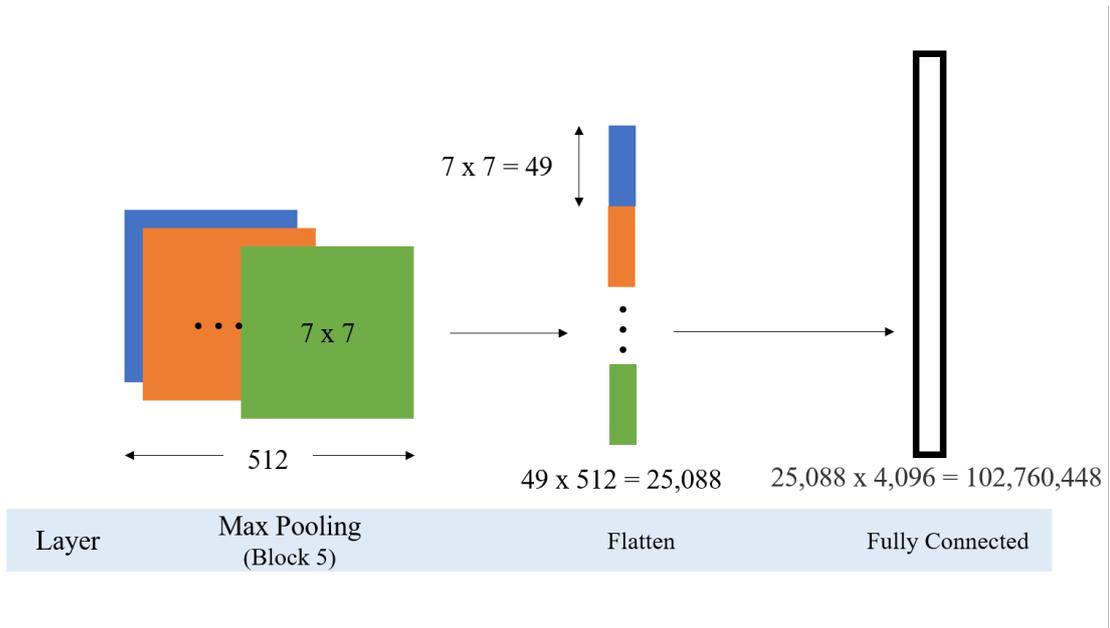
Table 3.2: Meanings of Symbols in Network Architecture Table

Symbol	Meaning
Image	Input Image Data
Conv.	2D Convolution Layer
Max Pool.	Max Pooling Layer
GAP	Global Average Pooling Layer
T. Conv.	2D Transposed Convolution Layer
Concatenate	Mathematical Operation (Concatenation for Features)
ch	Number of Channels
k	Kernel Size (Filter Size) of Layer
p	Padding Size before Operation
s	Stride Size while Operation
Output Shape	Output Shape after Operation
ReLU	ReLU Activation Function
BN	Batch Normalization

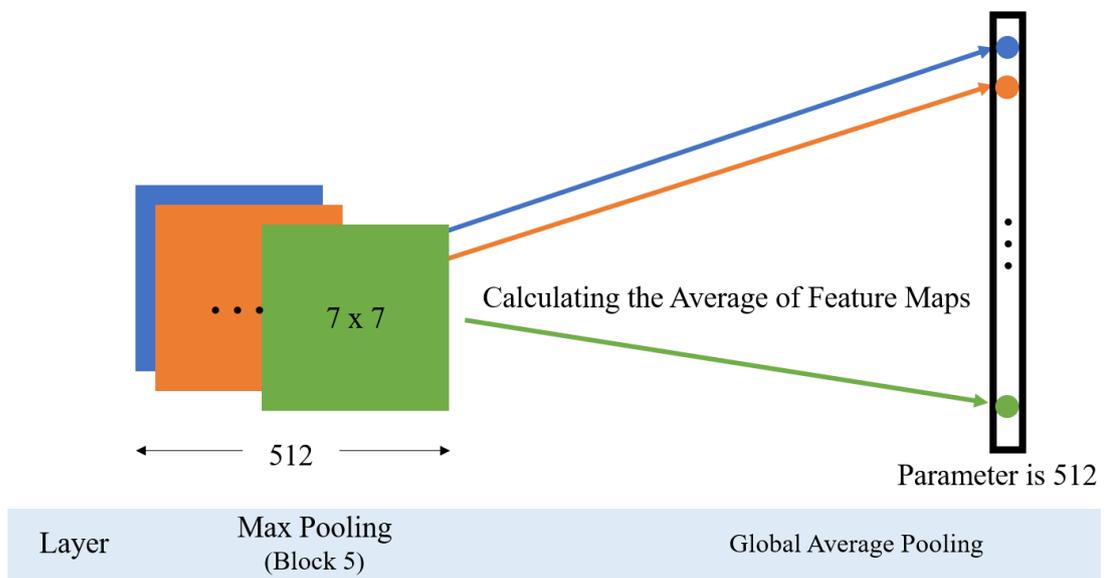
Block-3 to Block-5 have three convolutional layers and one max pooling layer. This article fed ImageNet weights until block-4, and block-5 trained to classify Glioma disease stages. Also, the author input pathological images to CNNs model after down-scale processing. In this step, the size of the images was changed from  $1000 \times 1000$  pixels to  $256 \times 256$  pixels. However, the flatten-layer and fully-connected layer have too many parameters in the original VGGNet, such as Figure 3.4 (a). To avoid the overfitting risk, the author installed the GAP layer (Table 3.4) instead of flatten-layer and fully-connected layers. Finally, the architecture of our model was Table 3.3.

Table 3.3: Our Classification Model

Block	Operation	ch	k	p	s	Output shape	Activation
	Input	3				(299, 299)	
1	Conv.	64	3	1	1	(299, 299)	ReLU
	Conv.	64	3	1	1	(299, 299)	ReLU
	Max Pool.	64	2	0	2	(149, 149)	
2	Conv.	128	3	1	1	(149, 149)	ReLU
	Conv.	128	3	1	1	(149, 149)	ReLU
	Max Pool.	128	2	0	2	(74, 74)	
3	Conv.	256	3	1	1	(74, 74)	ReLU
	Conv.	256	3	1	1	(74, 74)	ReLU
	Conv.	256	3	1	1	(74, 74)	ReLU
	Max Pool.	256	2	0	2	(37, 37)	
4	Conv.	512	3	1	1	(37, 37)	ReLU
	Conv.	512	3	1	1	(37, 37)	ReLU
	Conv.	512	3	1	1	(37, 37)	ReLU
	Max Pool.	512	2	0	2	(18, 18)	
5	Conv.	512	3	1	1	(18, 18)	ReLU
	Conv.	512	3	1	1	(18, 18)	ReLU
	Conv.	512	3	1	1	(18, 18)	ReLU
	Max Pool.	512	2	0	2	(9, 9)	
	GAP	512				(1, 512)	ReLU, Softmax



(a) Original Architecture of VGG16



(b) Global Average Pooling (GAP)

Figure 3.4: Comparative Fully-Connected-Layer vs. GAP

### 3.2.2 Training of Cell Nucleus Segmentation

The author focused on U-Net [15], U-Net++ [16], and DeepLabV3 [18] as the methods of cell nucleus segmentation, and compared these models. Also, Tables 3.4 to 3.14 show the structures of these models, respectively.

#### **Back Bone:**

This thesis used ResNet [9] (34-layers and 50-layers) as an encode part. Tables 3.4 to 3.11 show the architecture of each block of ResNet. ResNet has skip connections that connect the block and one previous block. Down sampling is used for skip connection in the first block to adjust the feature shape. The numbers after Loop Block in Stage-1 to Stage-3 mean the number of the same blocks.

#### **U-Net :**

Table 3.12 shows the architecture of U-Net. In the concatenate process, the output features were generated by using the features shown in Output.

#### **U-Net++ :**

Table 3.13 shows the architecture of U-Net++. In the concatenate process, U-Net++ calculates the upsampling from each encode part to enhance the accuracy more than U-Net. Thus this table expresses the detail of Input and Output as  $X_n$  ( $n$  means location numbers).

#### **DeepLabV3 :**

Table 3.14, and Figure 2.7 describe the architecture of DeepLab v3+. Figure 2.7 shows the entire structure of deeplab including ResNet101.

Table 3.4: Stage 1 (34-layer)

Part	Operation	ch	k	p	s	Activation
Down Sampling	Conv.	64	1	0	1	
First Block	Conv.	64	3	0	1	BN + ReLU
	Conv.	64	3	1	1	BN + ReLU
Loop Block (2)	Conv.	64	3	0	1	BN + ReLU
	Conv.	64	3	1	1	BN + ReLU

Table 3.5: Stage 1 (50-layer, 101-layer)

Part	Operation	ch	k	p	s	Activation
Down Sampling	Conv.	256	1	0	1	
First Block	Conv.	64	1	0	1	BN + ReLU
	Conv.	64	3	1	1	BN + ReLU
	Conv.	256	1	0	1	BN + ReLU
Loop Block (2, 2)	Conv.	64	1	0	1	BN + ReLU
	Conv.	64	3	1	1	BN + ReLU
	Conv.	256	1	0	1	BN + ReLU

Table 3.6: Stage 2 (34-layer)

Part	Operation	ch	k	p	s	Activation
Down Sampling	Conv.	128	1	0	2	
First Block	Conv.	128	3	1	2	BN + ReLU
	Conv.	128	3	1	2	BN + ReLU
Loop Block (3)	Conv.	128	3	1	1	BN + ReLU
	Conv.	128	3	1	1	BN + ReLU

Table 3.7: Stage 2 (50-layer, 101-layer)

Part	Operation	ch	k	p	s	Activation
Down Sampling	Conv.	512	1	0	2	
First Block	Conv.	128	1	0	1	BN + ReLU
	Conv.	128	3	1	2	BN + ReLU
	Conv.	512	1	0	1	BN + ReLU
Loop Block (3, 3)	Conv.	128	1	0	1	BN + ReLU
	Conv.	128	3	1	1	BN + ReLU
	Conv.	512	1	0	1	BN + ReLU

Table 3.8: Stage 3 (34-layer)

Part	Operation	ch	k	p	s	Activation
Down Sampling	Conv.	256	1	0	2	
First Block	Conv.	256	3	1	2	BN + ReLU
	Conv.	256	3	1	2	BN + ReLU
Loop Block (5)	Conv.	256	3	1	1	BN + ReLU
	Conv.	256	3	1	1	BN + ReLU

Table 3.9: Stage 3 (50-layer, 101-layer)

Part	Operation	ch	k	p	s	Activation
Down Sampling	Conv.	1024	1	0	2	
First Block	Conv.	256	1	0	1	BN + ReLU
	Conv.	256	3	1	2	BN + ReLU
	Conv.	1024	1	0	1	BN
Loop Block (5, 22)	Conv.	256	1	0	1	BN + ReLU
	Conv.	256	3	1	1	BN + ReLU
	Conv.	1024	1	0	1	BN + ReLU

Table 3.10: Stage 4 (34-layer)

Part	Operation	ch	k	p	s	Activation
Down Sampling	Conv.	512	1	0	2	
First Block	Conv.	512	3	2	2	BN + ReLU
	Conv.	512	3	2	2	BN + ReLU
Second Block	Conv.	512	3	2	1	BN + ReLU
	Conv.	512	3	2	1	BN + ReLU
Third Block	Conv.	512	3	2	1	BN + ReLU
	Conv.	512	3	2	1	BN + ReLU

Table 3.11: Stage 4 (50-layer, 101-layer)

Part	Operation	ch	k	p	s	Activation
Down Sampling	Conv.	2048	1	0	1	
First Block	Conv.	512	1	0	1	BN + ReLU
	Conv.	512	3	2	1	BN + ReLU
	Conv.	2048	1	0	1	BN + ReLU
Second Block	Conv.	512	1	0	1	BN + ReLU
	Conv.	512	3	4	1	BN + ReLU
	Conv.	2048	1	0	1	BN
Third Block	Conv.	512	1	0	1	BN + ReLU
	Conv.	512	3	8	1	BN + ReLU
	Conv.	2048	1	0	1	BN + ReLU

Table 3.12: U-Net Architecture

No	Operation	ch	k	p	s	Size	Activation	Output
0	Input	3				(256, 256)		
1	Conv.	64	7	3	2	(128, 128)	BN + ReLU	
	Max Pool.	64	3	0	2	(64, 64)		
2	Stage1	64				(64, 64)	BN + ReLU	X1
3	Stage2	128				(32, 32)	BN + ReLU	X2
4	Stage3	256				(16, 16)	BN + ReLU	X3
5	Stage4	512				(8, 8)	BN + ReLU	X4
6	T. Conv.	16	4	1	2	(16, 16)	BN + ReLU	X5
	Concatenate	512				(16, 16)	BN + ReLU	X4, X5
	Conv.	512	3	1	1	(16, 16)	BN + ReLU	
7	T. Conv.	128	4	1	2	(32, 32)	BN + ReLU	X6
	Concatenate	256				(32, 32)	BN + ReLU	X3, X6
	Conv.	128	3	1	1	(32, 32)	BN + ReLU	
8	T. Conv.	64	4	1	2	(64, 64)	BN + ReLU	X7
	Concatenate	128				(64, 64)	BN + ReLU	X2, X7
	Conv.	64	3	1	1	(64, 64)	BN + ReLU	
9	T. Conv.	32	4	1	2	(128, 128)	BN + ReLU	X8
	Concatenate	96				(128, 128)	BN + ReLU	X1, X8
	Conv.	32	3	1	1	(128, 128)	BN + ReLU	
10	T. Conv.	16	4	1	2	(256, 256)	BN + ReLU	X9
	Conv.	16	3	1	1	(256, 256)	BN + ReLU	
11	Conv.	1	1	0	1	(256, 256)		

Table 3.13: U-Net++

No	Operation	ch	k	p	s	size	Activation	Input	Output	
0	Input	3				(256, 256)			$X_0$	
1	Conv.	64	7	3	2	(128, 128)	BN + ReLU	$X_0$	$X_{1_1}$	
	Max Pool.	64	3	0	2	(64, 64)			$X_{1_2}$	
2	Stage1	64				(64, 64)	BN + ReLU	$X_{1_2}$	$X_2$	
3	Stage2	128				(32, 32)	BN + ReLU	$X_2$	$X_3$	
4	Stage3	256				(16, 16)	BN + ReLU	$X_3$	$X_4$	
5	Stage4	512				(8, 8)	BN + ReLU	$X_4$	$X_5$	
6	T. Conv.	16	4	1	2	(16, 16)	BN + ReLU	$X_5$	$X_{6_1}$	
	Concatenate	272				(16, 16)			$X_4, X_{6_1}$	$X_{6_2}$
	Conv.	16	3	1	1	(16, 16)			$X_{6_2}$	$X_{6_3}$
7	T. Conv.	256				(32, 32)	BN + ReLU	$X_4$	$X_{7_1}$	
	Concatenate	384				(32, 32)			$X_3, X_{7_1}$	$X_{7_2}$
	Conv.	256	3	1	1	(32, 32)			$X_{7_2}$	$X_{7_3}$
8	T. Conv.	256	4	1	2	(32, 32)	BN + ReLU	$X_{6_3}$	$X_{8_1}$	
	Concatenate	640				(32, 32)			$X_3, X_{8_1}, X_{7_3}$	$X_{8_2}$
	Conv.	256	3	1	1	(32, 32)			$X_{8_2}$	$X_{8_3}$
9	T. Conv.	128	4	1	2	(64, 64)	BN + ReLU	$X_3$	$X_{9_1}$	
	Concatenate	192				(64, 64)			$X_2, X_{9_1}$	$X_{9_2}$
	Conv.	128	3	1	1	(64, 64)			$X_{9_2}$	$X_{9_3}$
10	T. Conv.	128	4	1	2	(64, 64)	BN + ReLU	$X_{7_3}$	$X_{10_1}$	
	Concatenate	320				(64, 64)			$X_2, X_{10_1}, X_{9_3}$	$X_{10_2}$
	Conv.	128	3	1	1	(64, 64)			$X_{10_2}$	$X_{10_3}$
11	T. Conv.	128	4	1	2	(64, 64)	BN + ReLU	$X_{8_3}$	$X_{11_1}$	
	Concatenate	448				(64, 64)			$X_{11_1}, X_{9_3}$	$X_{10_2}$
	Conv.	128	3	1	1	(64, 64)			$X_{11_2}$	$X_{11_3}$
12	T. Conv.	64	4	1	2	(128, 128)	BN + ReLU	$X_2$	$X_{12_1}$	
	Concatenate	128				(128, 128)			$X_{1_2}, X_{12_1}$	$X_{12_2}$
	Conv.	64	3	1	1	(128, 128)			$X_{12_2}$	$X_{12_3}$
13	T. Conv.	64	4	1	2	(128, 128)	BN + ReLU	$X_{9_3}$	$X_{13_1}$	
	Concatenate	192				(128, 128)			$X_{1_2}, X_{13_1}$	$X_{13_2}$
	Conv.	64	3	1	1	(128, 128)			$X_{13_2}$	$X_{13_3}$
14	T. Conv.	64	4	1	2	(128, 128)	BN + ReLU	$X_{10_3}$	$X_{14_1}$	
	Concatenate	256				(128, 128)			$X_{1_2}, X_{12_3},$ $X_{13_3}, X_{14_1}$	$X_{14_2}$
	Conv.	64	3	1	1	(128, 128)			$X_{14_2}$	$X_{14_3}$

*Continued on next page*

Table 3.13 – *Continued from previous page*

No	Operation	ch	k	p	s	size	Activation	Input	Output
15	T. Conv.	64	4	1	2	(128, 128)	BN + ReLU	$X11_3$	$X15_1$
	Concatenate	320				(128, 128)		$X2,12_3,13_3,$ $X14_3,X15_1$	$X15_2$
	Conv.	64	3	1	1	(128, 128)	BN + ReLU	$X15_2$	$X15_3$
16	T. Conv.	32	4	1	2	(256, 256)	BN + ReLU	$X15_3$	$X16_1$
	Conv.	32	3	1	1	(256, 256)	BN + ReLU	$X16_1$	$X16_2$
17	Conv.	1	3	1	1	(256, 256)		$X16_2$	$X17_1$

### Loss Function:

Loss function is one of the important factors for CNNs model construction [24] so that this thesis used two loss functions and compared them. Cross-entropy [25] was defined as measuring the difference between two probability distributions. It is widely used for classification objectives, and as segmentation is pixel-level classification, it works well. BCE can be defined by

$$L_{BCE}(y, p) = -(y \log(p) - (1 - y) \log(1 - p)) \quad (3.1)$$

where  $p$  is prediction value.

Meanwhile, dice coefficient is widely used as an evaluation index by calculated the similarity between two images. It has also been adapted as a loss function known as Dice Loss (DL) [26]. DL can be defined by

$$DL(y, p) = 1 - \frac{2yp + 1}{y + p + 1} \quad (3.2)$$

where 1 of numerator and denominator is located to avoid the case of division by 0. Finally, this thesis conducted the training to tune hyper-parameters according to Table 3.15.

### 3.2.3 CNNs Evaluation

#### Classification Model:

The confusion matrix summarizes the results of the binary classification problem. It is generally used as an index to measure the performance of the binary classification of the machine learning model. This paper gives  $T_p$  (True Positive) label into the output when the learning model predicted GBM correctly. Similarly,  $T_n$  (True Negative) means the correct prediction of LGG. On the other hand, the author gives  $F_p$  (prediction GBM

Table 3.14: DeepLab v3+

No.	Operation	ch	k	p	s	d	output	Activation	Note
0	Image	3					(256, 256)		
1	Conv.	64	7	3	2	1	(128, 128)	BN + ReLU	
	Max Pool.	64	3	1	2	1	(64, 64)		
2	Stage 1	256					(64, 64)		X0
3	Stage 2	512					(32, 32)		
4	Stage 3	1024					(16, 16)		
5	Stage 4	2048					(16, 16)		X
6	Conv. (X)	256	1	0	1	1	(16, 16)	BN + ReLU	X1
7	Conv. (X)	256	3	6	1	6	(16, 16)	BN + ReLU	X2
8	Conv. (X)	256	3	12	1	12	(16, 16)	BN + ReLU	X3
9	Conv. (X)	256	3	18	1	18	(16, 16)	BN + ReLU	X4
10	Avg. Pool. (X)	2048					(1, 1)	BN+ReLU	X5
	Conv.	256	1	0	1	1	(1, 1)		
	Upsample	256					(16, 16)		
11	Concatenate	1280					(16, 16)	BN + ReLU	X1 ... X5
	Conv.	256	1	0	1	1	(16, 16)		
	Upsample	256					(64, 64)		
12	Conv. (X0)	48	1	0	1	1	(64, 64)	BN + ReLU	X7
	Concatenate	304					(64, 64)		
13	Conv.	256	3	1	1	1	(64, 64)	BN + ReLU	X6, X7
	Conv.	256	3	1	1	1	(64, 64)		
	Conv.	6	1	0	1	1	(64, 64)		
	Upsample	6					(256, 256)		

Table 3.15: Parameters for Training Segmentation.

Architecture	Backbone	Loss Function	Optimizer	Learning Rate
UNet	ResNet34	Binary Cross Entropy	Adam	0.001
UNet	ResNet34	Dice coefficient	Adam	0.001
UNet	ResNet50	Binary Cross Entropy	Adam	0.001
UNet	ResNet50	Dice Coefficient	Adam	0.001
UNet++	ResNet34	Binary Cross Entropy	Adam	0.001
UNet++	ResNet34	Dice coefficient	Adam	0.001
UNet++	ResNet50	Binary Cross Entropy	Adam	0.001
UNet++	ResNet50	Dice Coefficient	Adam	0.001
DeepLabV3	ResNet101	Binary Cross Entropy	Adam	0.001
DeepLabV3	ResNet101	Dice Coefficient	Adam	0.001

incorrectly) and  $F_n$  (prediction LGG incorrectly), respectively. Then evaluation value provided by confusion matrix can be defined as follows;

$$Accuracy = \frac{T_p + T_n}{T_p + F_p + T_n + F_n} \quad (3.3)$$

$$Precision = \frac{T_p}{T_p + F_p} \quad (3.4)$$

$$Recall = \frac{T_p}{T_p + T_n} \quad (3.5)$$

$$F - Measure = \frac{2Recall * Precision}{Recall + Precision} \quad (3.6)$$

### Segmentation Model :

When all constructed models shown in Table 3.15 were compared, the equations (3.4) to (3.6) were used. The author gave  $T_p$  (True Positive) label into the output when the learning model correctly predicted the region of cell nuclei. Similarly,  $T_n$  (True Negative) means the correct prediction of background. In the incorrect case,  $F_p$  (False Positive) and  $F_n$  (False Negative) were given. In addition, the author evaluated models with  $IoU$ ,  $DSC$ , and  $NOC$ . These indexes are defined by

$$IoU = \frac{T_p}{T_p + T_n + F_n} \quad , \quad (3.7)$$

$$DSC = \frac{2|P \cap G|}{|P| + |G|} \quad , \quad (3.8)$$

$$NOC = \begin{cases} 1 - \frac{|N_p - N_t|}{N_p} & (N_p < 2N_t) \\ 0 & (N_p > 2N_t) \end{cases} \quad (3.9)$$

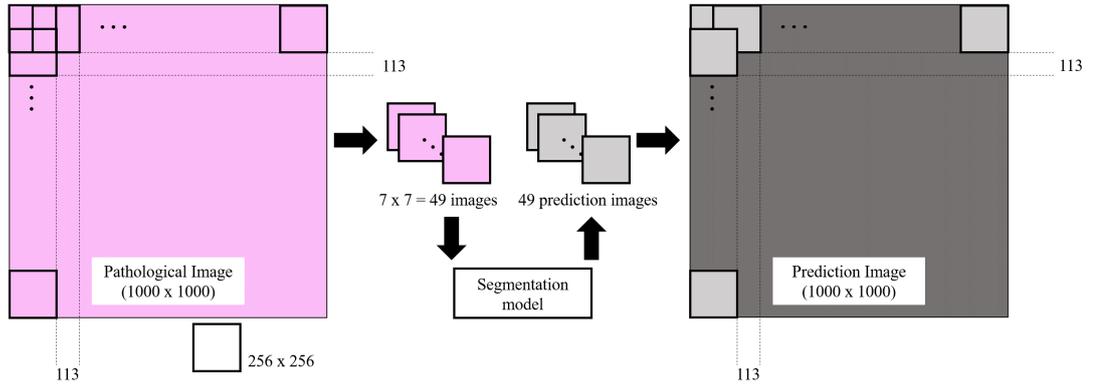


Figure 3.5: The Process of Mask Image Generation

where  $DSC$  means Dice coefficient and  $NOC$  is the number of cell nuclei. In  $DSC$ ,  $P$  is prediction result, and  $G$  is the ground truth.  $N_p$  and  $N_g$  mean the number of cell nuclei in the predicted and ground truth images, respectively. If  $N_p$  is more than  $2N_g$ ,  $NOC$  becomes 0.  $DSC$  and  $NOC$  are more important factor than others to select the best hyper-parameters.

### 3.3 Object-Level-Features and Interpretability of CNNs classification

#### 3.3.1 Mask Image of Large Size

This thesis created a cell nucleus segmentation model for glioma images in Sections 3.2.2 to 3.2.3. However, the optimal size of the constructed model is  $256 \times 256$  pixels, so that it was required to apply our model to  $1000 \times 1000$  pixels pathological images.

Thus this process patched glioma images of  $1000 \times 1000$  pixels into  $256 \times 256$  pixels with overlapping. Specifically, the author divided the given pathological images into 7 and patched them in the horizontal or vertical axis direction. Thus 49 images were generated from an image. Generally, segmentation performance is declined in the images' edge. In algorithm of this work, overlapping was employed to prevent the above problem. Then constructed model predicted all patched images. From the mask images of prediction, the author connected them with except for the images' edge. This process could realize to remove low accuracy prediction region. After that, nucleus mask images whose size is the same as input pathological images were obtained. Figure 3.5 illustrates the above process.

### 3.3.2 Visualization

This thesis applied Grad CAM++ to the proposal classification model. Section 2.3 explained how Grad CAM++ detects important regions for CNNs classification. In order to find the important region from the heatmap automatically, the author defined important area for CNNs classification, in which Hue was  $H < 20^\circ$ , or  $H > 200^\circ$  and  $S > 128^\circ$  in HSV color space.

### 3.3.3 Object Level Features

The cell nucleus's size and positional relationship are useful for pathological analysis [5]. This paper investigated the relationship between CNNs' decision and Object-Level-Features, which can be extracted after cell nuclei was approximated into several shapes. In particular, this work used the shapes of the ellipse, convex hull, bounding box, and boundary such as Figure 3.6. Also, the detail of Object-Level-Features were as follows.

#### 1. Elliptical Features

- Major length - It is the major length of the ellipse.
- Minor length - It is the minor length of the ellipse.
- Aspect ratio - It is the ratio of the major/minor length.
- Angle - It is the angle of the ellipse.
- Area - It is the area of the ellipse.
- Eccentricity - It is the eccentricity of the ellipse.
- Extent - It is the area ratio of ellipse and cell nuclei.

#### 2. Convex Hull Features

- Area - It is the area of the convex hull area.
- Solidity - It is the longest distance of the displacing point.
- Defect point - It is the number of defect points.

#### 3. Boundary Features

- Radii (average) - It is an average of all distances from the centre to each boundary point.
- Radii (median) - It is a median of all distances from the centre to each boundary point.
- Perimeter - It is the perimeter of cell nuclei.

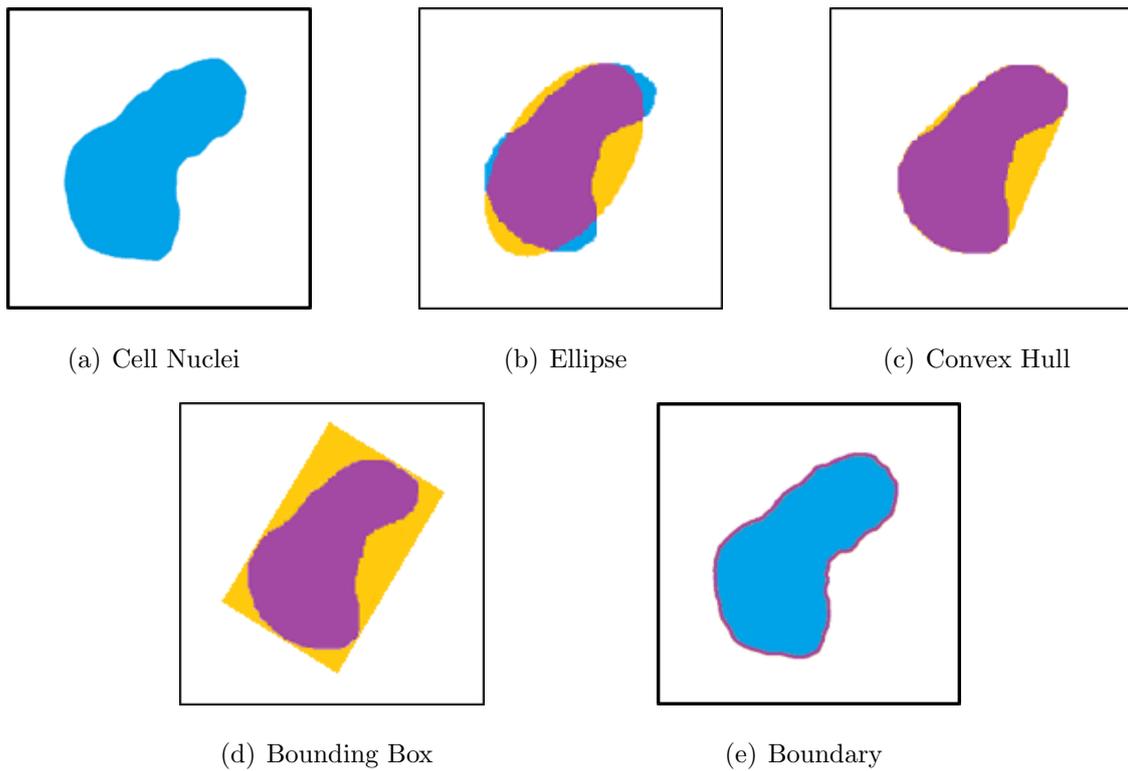


Figure 3.6: Explanation

- Reciprocal - It is Reciprocal of radii.

#### 4. Boundary Box Features

- Accept Ratio - It is the ratio of the minor/major length.
- Extent - It is the area ratio of the bounding box and cell nuclei.
- Area - It is the area of the Boundary Box.

In this work, OpenCV was used because it can detect object contours easily. In this tool, the detected contours are expressed as white objects in the black background. Furthermore, OpenCV has other functions for the various objects' approximation. The author constructed the algorithm of Object-Level-Feature extraction with OpenCV.

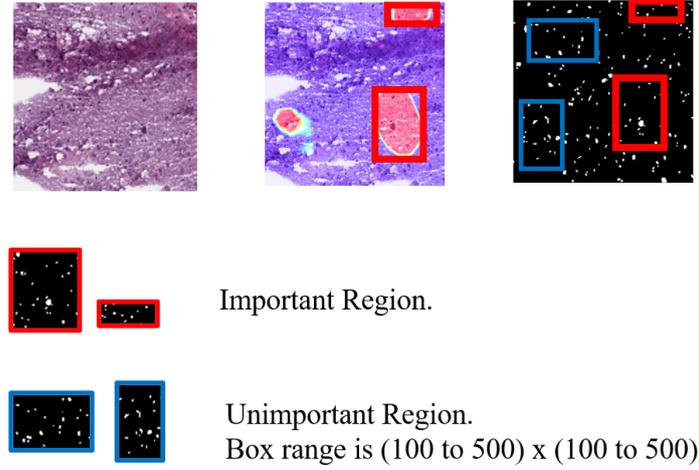


Figure 3.7: Important Region and Random Region on the Mask Image

### 3.3.4 Feature Selection

In Sections 3.3.2 and 3.3.1, our segmentation model generated a mask image of the cell nucleus, and Grad CAM++ found an important area for the CNNs classification. By comparing mask images and heatmap, the author obtained mask images of an important area. Furthermore, unimportant areas were extracted to compare with the above data. Figure 3.7 explains the outline of the above process. For instance, this work used the bounding box with the width of 100 to 500 pixels and height of 100 to 500 pixels to define the area. Then the author extracted its region from a random point of mask images of  $1000 \times 1000$  pixels. This area was set to the unimportant area. The point was that the author obtained important/unimportant mask images by referring the both of nucleus mask image and heatmap, and then object-level-features including in these areas were compared.

Finally, the author input the obtained features into K-S test (Kolmogorov–Smirnov test). K-S test is a nonparametric test of the equality of continuous, one-dimensional probability distributions that can be used to compare two samples. For instance, when a sample X is distributed  $x_1, x_2, \dots, x_n$ , also a sample Y is distributed  $y_1, y_2, \dots, y_m$ , each cumulative probability distribution  $S_n(x)$  (sample X) and  $S_m(x)$  (sample Y) can be defined by

$$S_n = \frac{1}{n} \sum_{i=0}^n X_i(x) \quad \begin{cases} 1 & (x_i \leq x) \\ 0 & (x_i > x) \end{cases} \quad (3.10)$$

$$S_m = \frac{1}{m} \sum_{i=0}^n X_i(x) \quad \begin{cases} 1 & (y_i \leq x) \\ 0 & (y_i > x) \end{cases} \quad (3.11)$$

Furthermore, the maximum value of the difference of above two cumulative probability distribution as KS-Static can be defined by

$$D = \max_{-\infty < x < \infty} |S_n(x) - S_m(x)|. \quad (3.12)$$

If the following formula is true, there is the significant difference between important region and unimportant region.

$$D > C(\alpha) \sqrt{\frac{nm}{n+m}} \quad (3.13)$$

In the above equation, the value of  $C(\alpha)$  is given in the table below for the most common levels of  $\alpha$ .

Table 3.16: Significance Level and  $C(\alpha)$

$\alpha$	0.20	0.15	0.10	0.05	0.025	0.01
$C(\alpha)$	1.073	1.138	1.224	1.358	1.48	1.628

In this study, the author set up the 5% significance level ( $\alpha = 0.05$ ) and confirmed whether a significance difference exists between the regions or not.

## Chapter 4

# Result and Discussion

### 4.1 Performance of Classification Model

Table 4.1 shows the result of the learning with deep CNNs. The classification accuracy was around 99% in every experiment. Table 4.2 provides the confusion matrix, and evaluation value of CNNs model with the highest performance in this paper. In particular, the classification accuracy of the constructed model was 99.2% so that it was higher than the classification those of the previous research [27]. These results indicate that CNNs was learned successfully for disease stage classification. Also, Figure 4.1 shows the training curve when three data sets were used. This graph shows that any problems did not exist in the training process. In the following experiments, the author used the learning model with the performance shown in Table 4.2.

### 4.2 Segmentation

Figures (4.2 to 4.4) show the learning curves of each data set. Also, Table 4.3 and Figure 4.5 show the result of all learning models.

In the pathological image segmentation, predicting the background is easy for the learning model. Also, many pathological images have background region a lot. Thus, almost the learning model had good accuracy in *Precision*, *Recall*, *IoU*, and *F-measure* evaluation values.

Table 4.1: Classification Accuracy for Glioma Pathological Image.

	Classification Accuracy [%]
Data Set 1	99.2
Data Set 2	99.3
Data Set 3	99.2
Average	99.2

Table 4.2: Confusion Matrix for Learning CNNs with the Most Highest Performance.

Table Head	Learning Model Prediction		
	<i>GBM</i>	<i>LGG</i>	<i>Recall</i>
GBM	995	5	99.5
LGG	11	989	98.9
Precision	98.9	99.7	
F-Measure	99.3	99.3	

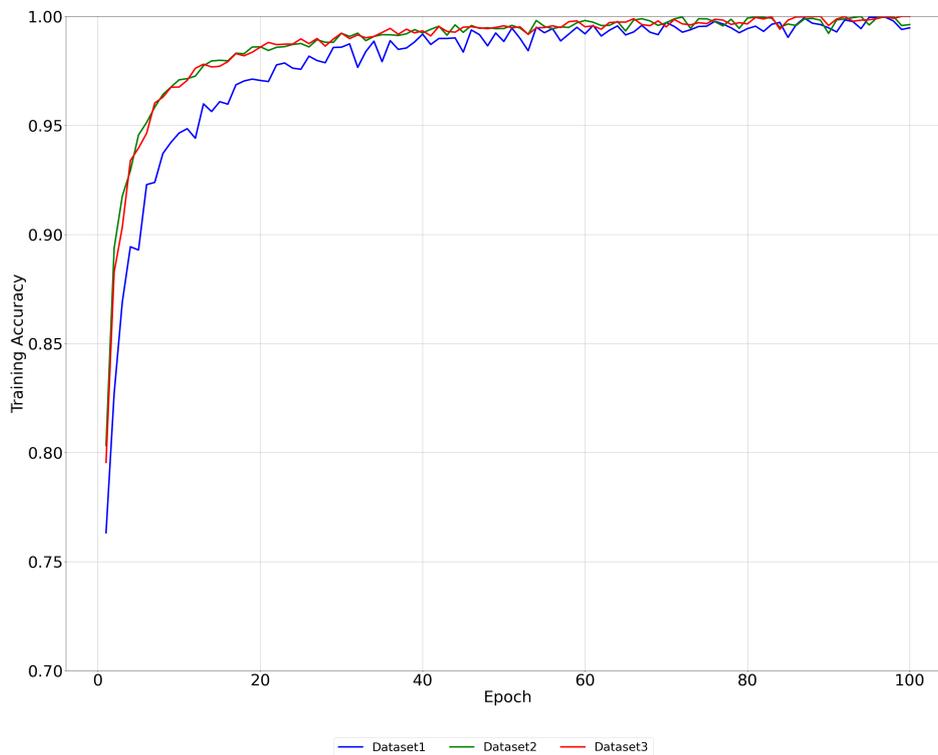


Figure 4.1: Learning Curve of VGG16 Training

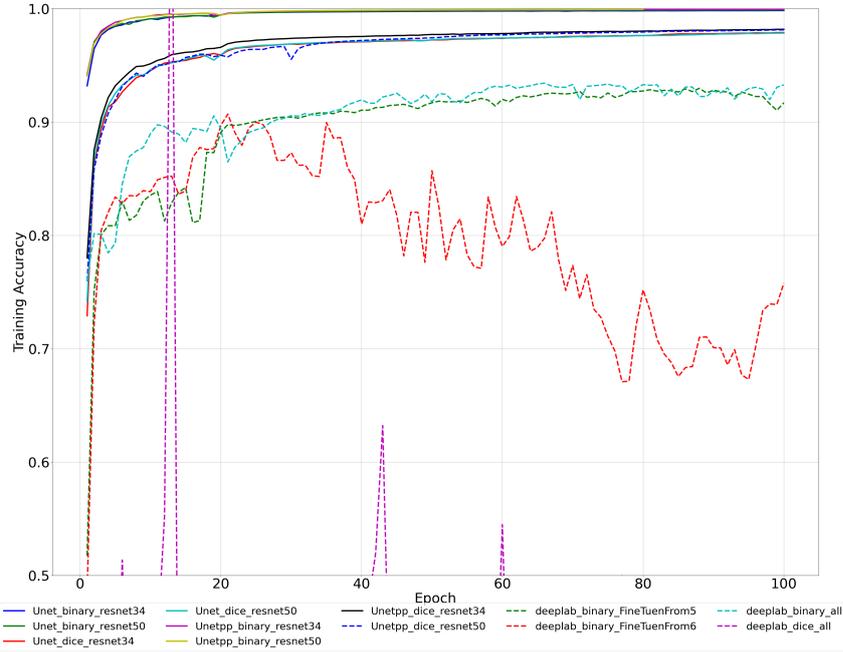


Figure 4.2: Learning Curves of All Models (Data Set 1)

The author confirmed that there was critical difference between the best model and other models in *DSC* and *NOC*. In particular, DeepLabV3 model with binary cross-entropy had high performance in *Accuracy*, *Recall*, *Precision*, *F-measure*, and *IoU*, but *NOC* value was much smaller than those of U-Net and U-Net++. For instance, in Figures 4.4 to 4.7, the prediction quality of the learning model with DeepLabV3 was much low, especially model construction using dice coefficient was a failure. Also, the learning curves of DeepLabV3 (Figures 4.2 to 4.4) show that there are some problems in the learning process. This result indicates that the learning model had low performance in the cell nuclei prediction.

On the other hand, the training of U-Net and U-Net++ was finished successfully. (Figures 4.2 to 4.4). Also, *DSC* and *NOC* scores were good (Figure 4.5). Furthermore, models with ResNet34 had high performance compare to those with ResNet50. This problem was easy because it was just a problem to detect cell nuclei or not. Thus the author concluded that ResNet50 was too deep for this segmentation problem. Finally, U-Net++ with ResNet34 and binary cross-entropy had the highest *NOC* value of all models, and it had a good score in other evaluation values. Therefore, this paper used this model for the following experiments.

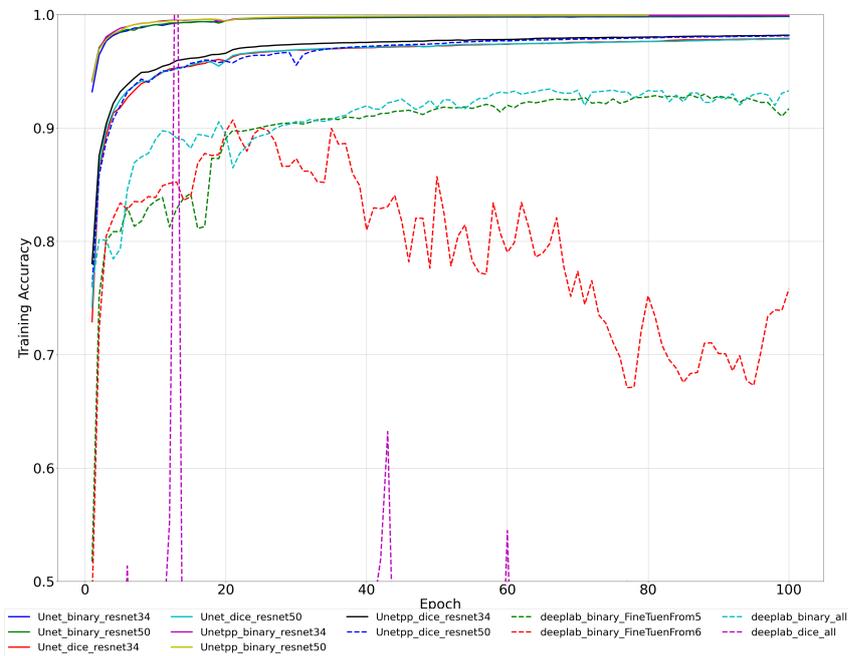


Figure 4.3: Learning Curves of All Models (Data Set 2)

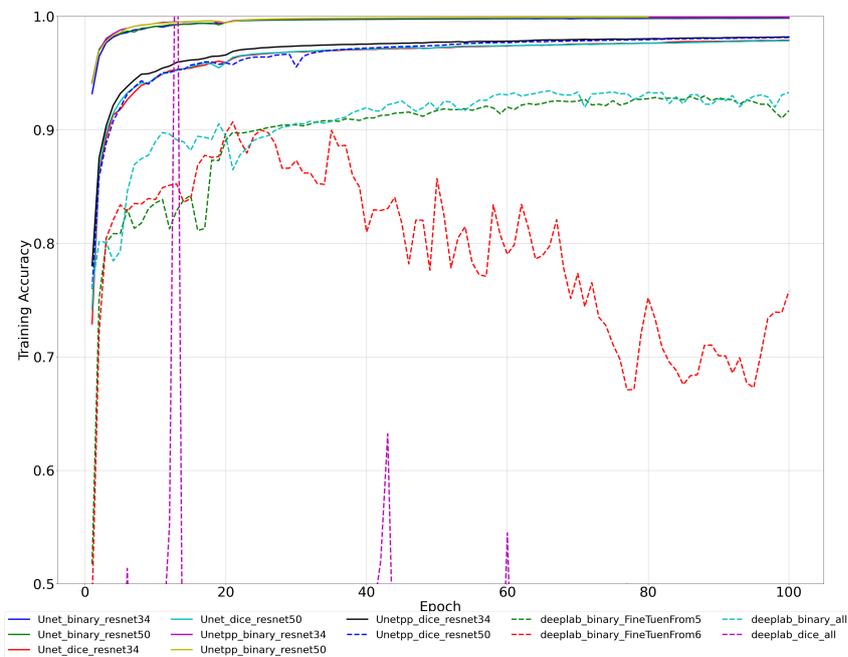


Figure 4.4: Learning Curves of All Models (Data Set 3)

Table 4.3: Result of All Segmentation Models

Loss Function # of ResNet' layer	U-Net				U-Net++				DeepLabV3	
	Dice 34	Dice 50	BCE 34	BCE 50	Dice 34	Dice 50	BCE 34	BCE 50	Dice -	BCE -
DSC	0.92	0.92	0.91	0.91	0.92	0.92	0.91	0.92	0.50	0.86
NOC	0.67	0.66	0.73	0.68	0.68	0.66	0.75	0.74	0.0584	0.28
precision	0.93	0.93	0.92	0.93	0.92	0.92	0.92	0.92	0.85	0.86
recall	0.98	0.98	0.97	0.98	0.9	0.98	0.98	0.98	0.52	1.00
f1	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.56	0.92
IoU	0.89	0.89	0.88	0.89	0.90	0.90	0.90	0.90	0.50	0.99

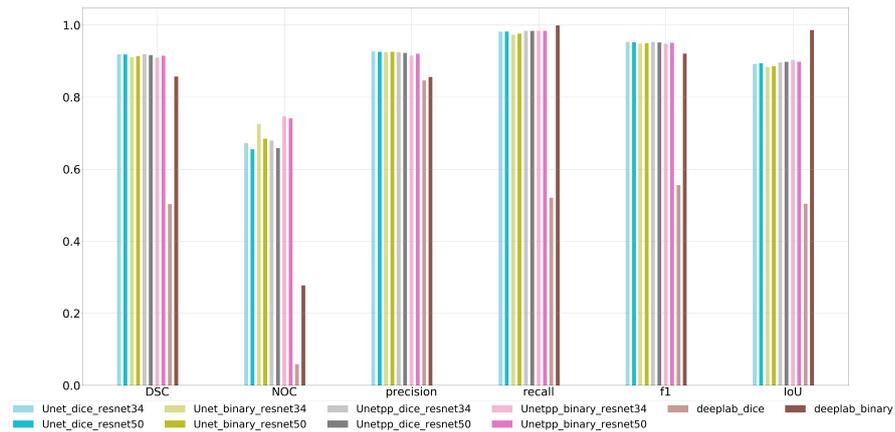


Figure 4.5: Bar Graph of All Segmentation Models

Table 4.4: Prediction Result I

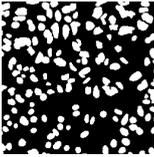
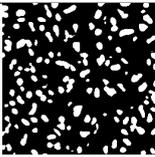
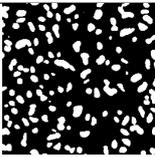
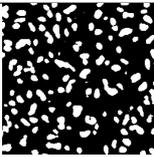
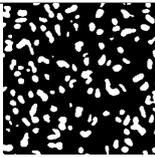
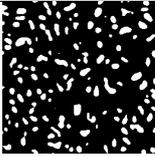
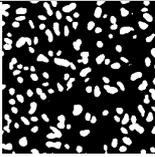
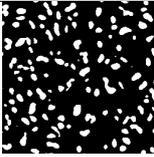
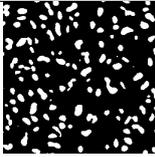
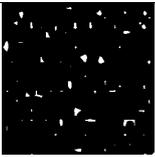
Model	Parameter	 Ground Truth			
UNet					
	Back Born	ResNet34	ResNet34	ResNet50	ResNet50
	Loss Func	BCE	Dice	BCE	Dice
	DSC	0.875	0.894	0.876	0.889
	NOC	0.943	0.967	0.943	0.976
UNet++					
	Back Born	ResNet34	ResNet34	ResNet50	ResNet50
	Loss Func	BCE	Dice	BCE	Dice
	DSC	0.863	0.899	0.861	0.880
	NOC	0.919	0.976	0.878	0.886
DeepLabV3					
	Loss Func	BCE		dice	
	DSC	0.737		0.706	
	NOC	0.163		0.016	

Table 4.5: Prediction Result II

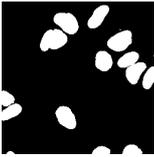
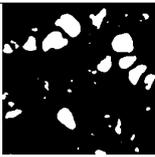
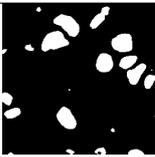
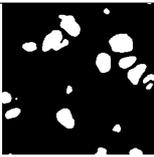
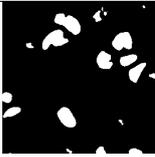
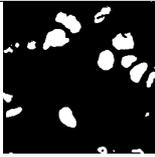
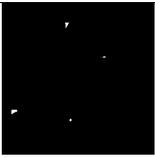
Model	Parameter	 Ground Truth			
UNet					
	Back Born	ResNet34	ResNet34	ResNet50	ResNet50
	Loss Func	BCE	Dice	BCE	Dice
	DSC	0.921	0.954	0.919	0.937
	NOC	0.000	0.500	0.000	0.583
UNet++					
	Back Born	ResNet34	ResNet34	ResNet50	ResNet50
	Loss Func	BCE	Dice	BCE	Dice
	DSC	0.934	0.952	0.939	0.947
	NOC	0.667	0.583	0.583	0.583
DeepLabV3					
	Loss Func	BCE		dice	
	DSC	0.852		0.816	
	NOC	0.083		0.167	

Table 4.6: Prediction Result III

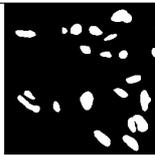
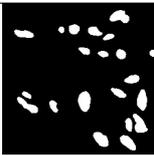
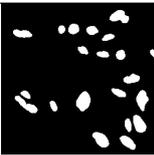
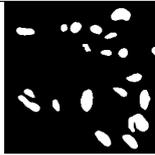
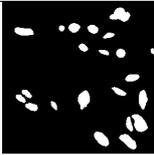
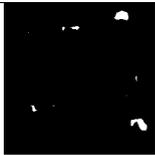
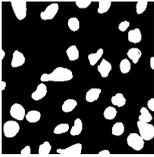
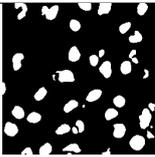
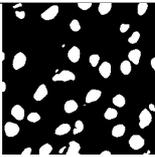
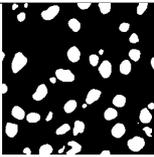
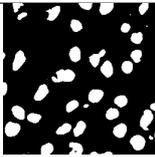
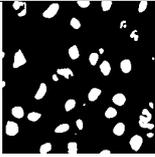
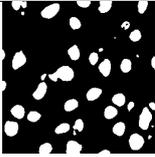
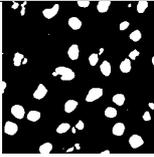
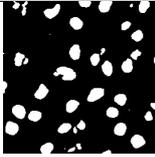
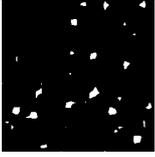
Model	Parameter	 Ground Truth			
UNet					
	Back Born	ResNet34	ResNet34	ResNet50	ResNet50
	Loss Func	BCE	Dice	BCE	Dice
	DSC	0.949	0.957	0.952	0.955
	NOC	1.000	0.952	1.000	1.000
UNet++					
	Back Born	ResNet34	ResNet34	ResNet50	ResNet50
	Loss Func	BCE	Dice	BCE	Dice
	DSC	0.944	0.957	0.947	0.958
	NOC	1.000	0.952	0.952	1.000
DeepLabV3					
	Loss Func	BCE		dice	
	DSC	0.878		0.833	
	NOC	0.190		0.095	

Table 4.7: Prediction Result IV

Model	Parameter	 Ground Truth			
UNet					
	Back Born	ResNet34	ResNet34	ResNet50	ResNet50
	Loss Func	BCE	Dice	BCE	Dice
	DSC	0.923	0.935	0.927	0.931
	NOC	0.914	0.886	0.914	0.743
UNet++					
	Back Born	ResNet34	ResNet34	ResNet50	ResNet50
	Loss Func	BCE	Dice	BCE	Dice
	DSC	0.911	0.934	0.915	0.933
	NOC	0.971	0.943	0.886	0.971
DeepLabV3					
	Loss Func	BCE		dice	
	DSC	0.804		0.751	
	NOC	0.429		0.057	

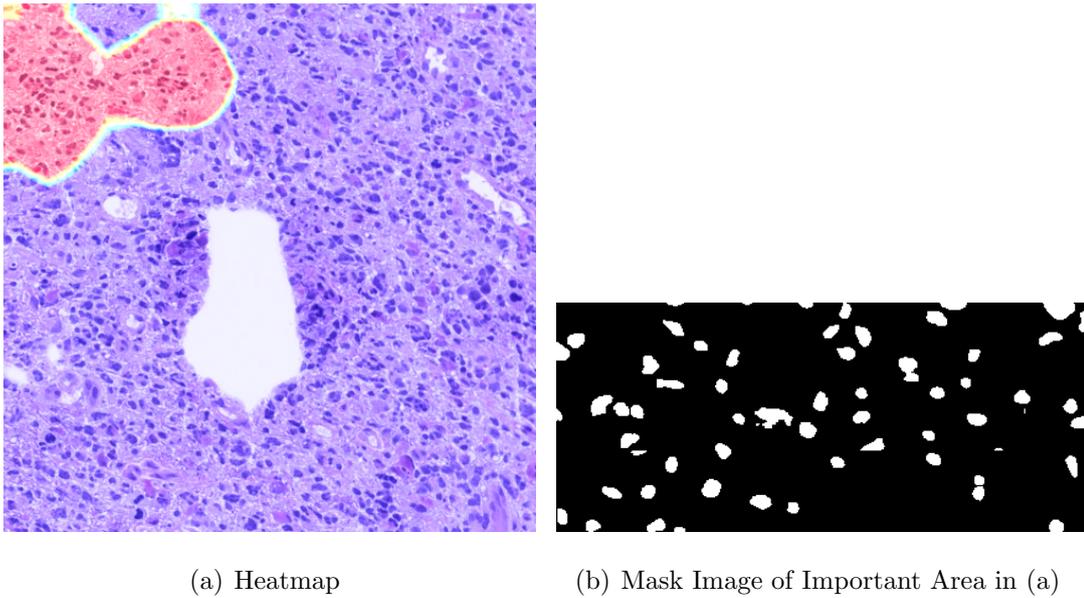


Figure 4.6: Heatmap by Grad CAM++ and Mask Image

### 4.3 Relationship between CNNs Decision and Object-Level-Features

Figure 4.6 (a) shows an example of heatmap obtained by Grad CAM++, and this image shows positions of important areas. Then the author can obtain mask images of important areas like Figure 4.6 (b). After that, object-level-features were extracted from the mask images, and K-S test was employed to compare important areas with unimportant areas. Table 4.8 shows the result of K-S test. In this table, “O” means statistically significant difference *i.e.*, formula  $D\sqrt{\frac{nm}{n+m}}$  was greater than 1.36.

In contrast, “–” means an insignificant difference. This table shows that 11 features were effective for histopathological image classification using CNNs. This result indicates that CNNs can learn and acquire pathologists’ knowledge thorough the training. In particular, most all features with respect to the area including elliptical features were significant. This result indicates that CNNs extracts these features for classification. Furthermore, boundary and convex hull features had many significant marks in the table. These features will be also effective for CNNs classification.

Table 4.8: Result of K-S Test

Nuclei Shape	Feature	Result
Elliptical	Accept Ratio	-
	Angle	-
	Area	-
	Eccentricity	O
	Extent	O
	Major Length	O
	Minor Length	-
Bounding Box	Accept Ratio	O
	Area	O
	Extent	-
Boundary	Radii (average)	O
	Radii (median)	O
	Radii Recipient	O
	Perimeter	O
Convex Hull	Area	O
	Defect Point	-
	Solidity	O

## Chapter 5

# Concluding Remarks

### 5.1 Conclusion

The purpose of this project was the investigation how CNNs make a decision. Last decade, CNNs technologies were growing quickly. In particular, CNNs' diagnosis is expected that pathologists' work will reduce remarkably and those criteria will be quantitative. However, in pathological analysis, there are many scenarios where doctors have to explain the detail of their diagnosis. In a similar, constructed CNNs model is required the diagnosis reason even if CNNs' classification ability superior.

This thesis discussed the relationship between CNNs decision and object-level-features. From this investigation, the author obtained the result that CNNs can learn and acquire pathologists' knowledge thorough the training. In particular, most all features with respect to the area including elliptical features were significant. This result indicates that CNNs extracts these features for classification. Furthermore, boundary and convex hull features had many significant marks in the table. These features will be also effective for CNNs classification.

### 5.2 Future Works

Firstly, this work will have to investigate the priority in object-level-features that has the significant difference. Important features with CNNs were determined by using K-S test. This test indicates just a whether there is a significant difference or not. In order to explain the CNNs classification process clearly, visualizing the important degree should be conducted.

Secondly, it is required to investigate the relationship between CNNs decision and other features. CNNs extracted features from the overall image. Thus there is possibility that CNNs used other features for the classification. In addition, according to [5], the positional features of cell nucleus are useful for histopathological analysis. The author

did not investigate regarding these features still in this thesis. For the above reason, this project will have to research regarding other features.

## Acknowledgement

First of all, I really would like to express my deepest gratitude to Prof. Shinji Tsuruoka, who is the honorary professor of Mie University and Associate Prof. Hiroharu Kawanaka at Graduate School of Engineering, Mie University who offered continuing supports and constant encouragements.

I am also grateful to Prof. Bruce J. Aronow and Prof. V. B. Surya Prasath at Cincinnati Children's Hospital Medical Center, USA. They provided a lot of technical help and encouragement. Prof. Aronow accepted me as a short-term study abroad student at his laboratory in Cincinnati Children's Hospital Medical Center. Thanks to his help and encouragement, I was able to make good progress in my project. Prof. Surya gave me a lot of technical and language help in my research project and writing papers.

In addition, this study was partially supported by Mie University study abroad program. Thanks to this program, I could stay Cincinnati Children's Hospital Medical Center for a month to obtain great progress and discuss the project with experts. I would like to sincerely thank this scholarship program. And finally, I would like to appreciate other members related my research project.

## Reference

- [1] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, p. 436, 2015.
- [2] P. Nardelli, D. Jimenez-Carretero, D. Bermejo-Pelaez, G. R. Washko, F. N. Rahaghi, M. J. Ledesma-Carbayo, and R. San José Estépar, “Pulmonary artery–vein classification in ct images using deep learning,” *IEEE Transactions on Medical Imaging*, vol. 37, no. 11, pp. 2428–2440, 2018.
- [3] A. Kumar, J. Kim, D. Lyndon, M. Fulham, and D. Feng, “An ensemble of fine-tuned convolutional neural networks for medical image classification,” *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 1, pp. 31–40, 2017.
- [4] M. Dinesh Kumar, M. Babaie, S. Zhu, S. Kalra, and H. R. Tizhoosh, “A comparative study of cnn, bovw and lbp for classification of histopathological images,” in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2017.
- [5] M. N. Gurcan, L. E. Boucheron, A. Can, A. Madabhushi, N. M. Rajpoot, and B. Yener, “Histopathological image analysis: A review,” *IEEE Reviews in Biomedical Engineering*, vol. 2, pp. 147–171, 2009.
- [6] A. Chattopadhyay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, “Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks,” *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 839–847, 2018.
- [7] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009.
- [8] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *Conference on Learning Representations (ICLR)*, 2015.
- [9] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

- [10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [12] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *International Conference on Machine Learning (ICML)*, 2015.
- [13] M. Lin, Q. Chen, and S. Yan, “Network in network,” in *International Conference on Learning Representations (ICLR)*, 2014.
- [14] V. B. S. Prasath, K. Fukuma, B. J. Aronow, and H. Kawanaka, “Cell nuclei segmentation in glioma histopathology images with color decomposition based active contours,” in *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2015.
- [15] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *Mexican International Conference on Artificial Intelligence (MICAI)*, 2015.
- [16] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, “Unet++: A nested u-net architecture for medical image segmentation,” in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, 2018.
- [17] X. Li, H. Chen, X. Qi, Q. Dou, C.-W. Fu, and P. A. Heng, “H-denseunet: Hybrid densely connected unet for liver and tumor segmentation from ct volumes,” in *IEEE Transactions on Medical Imaging*, vol. 37, no. 12, 2018.
- [18] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018.
- [19] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *European Conference on Computer Vision (ECCV)*, 2014.
- [20] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning deep features for discriminative localization,” in *European Conference on Computer Vision*, 2015.

- [21] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-cam: Visual explanations from deep networks via gradient-based localization,” *International Journal of Computer Vision*, vol. 128, no. 2, 2019.
- [22] “The cancer genome atlas,” 2018/2 accessed, <https://cancergenome.nih.gov>.
- [23] “Monuseg - grand challenge,” 2020/5 accessed, <https://monuseg.grand-challenge.org/>.
- [24] “A survey of loss functions for semantic segmentation.”
- [25] Ma Yi-de, Liu Qing, and Qian Zhi-bai, “Automated image segmentation using improved pcnn model based on cross-entropy,” in *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing*, 2004.
- [26] F. Milletari, N. Navab, and S.-A. Ahmadi, “V-net: Fully convolutional neural networks for volumetric medical image segmentation,” in *Fourth International Conference on 3D Vision (3DV)*, 2016.
- [27] A. Yonekura, H. Kawanaka, V. B. S. Prasath, B. J. Aronow, and H. Takase, “Automatic disease stage classification of glioblastoma multiforme histopathological images using deep convolutional neural network,” *Biomedical Engineering Letters*, vol. 8, no. 3, pp. 321–327, 2018.

# Publication List

## Journal Paper

(1) 齊藤 大祐, 川中 普晴, V. B. Surya Prasath, Bruce J. Aronow, "CNNを用いたGliomaの疾患進行度評価における形状特徴量に関する一検討", 電気学会論文誌C (電子・情報・システム部門誌) Vol.140 No.12 (2020) 特集:電気・電子・情報関係学会東海支部連合大会, pp. 1367-1368, 2020.

## International Conferences

(1) Daisuke Saito, Hiroharu Kawanaka, V. B. Surya Prasath, and Bruce J. Aronow, "Glioma Histopathological Images Classification with Deep CNN and Object Level Features," in Joint 2020 9th International Conference on Informatics, Electronics & Vision (ICIEV) & 2nd International Conference on Imaging, Vision & 4th Pattern Recognition (icIVPR), pp. 47, 2020.

(2) Daisuke Saito, Hiroharu Kawanaka, V. B. Surya Prasath, and Bruce J. Aronow, "A Study on CNNs Decision and Object-Level-Features for Glioma Histopathological Images," in the 9th International Symposium for Sustainability by Engineering at Mie University (Research Area C), pp. 49-50, 2019.

(3) Daisuke Saito, Hiroharu Kawanaka, V. B. Surya Prasath, and Bruce J. Aronow, "Comparative Study on Automatic Segmentation Approaches for Glioma Histopathology Images," in International Workshop of Regional Innovation Studies (IWRIS 2020), pp. 25-28, 2020.

(4) Daisuke Saito, Hiroharu Kawanaka, V. B. Surya Prasath, and Bruce J. Aronow, "A Study on CNNs Decision and Object-Level-Features for Glioma Histopathological Images," in the 10th International Symposium for Sustainability by Engineering at Mie University (Research Area C), pp. 41-42, 2020.

## Domestic Conferences

- (1) 齊藤大祐, 川中普晴, V. B. Surya Prasath, Bruce J. Aronow, “CNNを用いた脳組織病理画像の分類問題における注目領域の可視化に関する一検討,” 令和元年度電気・電子・情報関係学会東海支部連合大会講演論文集, E4-3, 2019.
- (2) 齊藤大祐, 川中普晴, V. B. Surya Prasath, Bruce J. Aronow, “CNNを用いたGliomaのサブタイプ分類に関する一検討,” 2019年度日本生体医工学会東海支部大会, pp. 27, 2019.
- (3) 齊藤大祐, 川中普晴, V. B. Surya Prasath, Bruce J. Aronow, “CNNを用いた脳組織病理画像における抽出特徴量に関する一検討,” 地域イノベーション学会2019, pp. 6, 2019.
- (4) 齊藤大祐, 川中普晴, V. B. Surya Prasath, Bruce J. Aronow, “Glioma病理画像における細胞核のフロー特徴量に関する一検討,” 令和2年度電気・電子・情報関係学会東海支部連合大会講演論文集, E4-3, 2020.
- (5) 齊藤大祐, 川中普晴, V. B. Surya Prasath, Bruce J. Aronow, “CNNを用いた病理画像診断における判断根拠と細胞核フロー特徴量の関係性に関する一考察,” 2020年度日本生体医工学会東海支部大会, pp. 11, 2020