Basic Research on Image Diagnosis of Trigeminal Neuralgia Using MRI Image and Deep Learning

福井, 喬太
九州大学総合理工学府総合理工学専攻機械・システム理工学メジャー

2022
Thesis for the degree of Master

Basic Research on Image Diagnosis of Trigeminal Neuralgia Using MRI Image and Deep Learning

Interdisciplinary Graduate School of Engineering Science

Kyushu University

Mechanical and Systems Engineering
Bioenergy Engineering Laboratory

Kyota FUKUI

Supervisor Mitsugu TODO

Submission Date 27 January 2023

Basic Research on Image Diagnosis of Trigeminal Neuralgia Using MRI Image and Deep Learning

Kyota Fukui

Contents

Contents of Figure

Contents of Table

1 Introduction

1.1 Trigeminal Neuralgia

Facial sensation is innervated by trigeminal neuralgia with some exceptions, and facial pain and sensory disturbance are caused by lesions of the trigeminal nerve, its branches, and various areas of the central nervous system, including the thalamus and cortical sensory cortex [1]. Trigeminal neuralgia, a typical disorder of facial pain, is unilateral (rarely bilateral) paroxysmal facial pain that occurs in the area innervated by the trigeminal nerve. The pain is limited to the area innervated by the trigeminal nerve branches, is evoked by non-nociceptive stimuli, and is repeated for a short time [2]. It is a disease that produces intense pain that feels like an electric shock, and thus has a significant impact on daily life. Trigeminal neuralgia is classified into several categories and is classified as trigeminal neuralgia and painful trigeminal neuropathy by the International Headache Society in the third edition of ICHD [3]. Within that trigeminal neuralgia, it is divided into typical trigeminal neuralgia, secondary trigeminal neuralgia, and idiopathic trigeminal neuralgia. The trigeminal neuralgia guidelines of the European Neurological Association classify trigeminal neuralgia into primary and secondary trigeminal neuralgia, and primary trigeminal neuralgia is divided into typical and idiopathic trigeminal neuralgia according to the degree of neurovascular compression. In this study, we focused on typical trigeminal neuralgia, which is the most frequent type of trigeminal neuralgia.

1.1.1 Patient’s conditions

The lifetime incidence is estimated to be 0.16~0.3% [4][5][6]. It tends to be more common in women and increases with age. The incidence of new cases of trigeminal neuralgia is reported to be 4.3~27/100,000 persons. The mean age at onset of typical trigeminal neuralgia (CTN) was 53 years and that of secondary trigeminal neuralgia (STN) was 43 years, with typical trigeminal neuralgia tending to be older. The age distribution of typical trigeminal neuralgia (TN) is widely distributed from young to old, and according to a report based on tertiary care centers, secondary trigeminal neuralgia accounts for 14~20% of all trigeminal

neuralgia (TN) cases [7]. The age distribution of typical trigeminal neuralgia (CTN) and secondary trigeminal neuralgia (STN) is shown in Fig. 1.

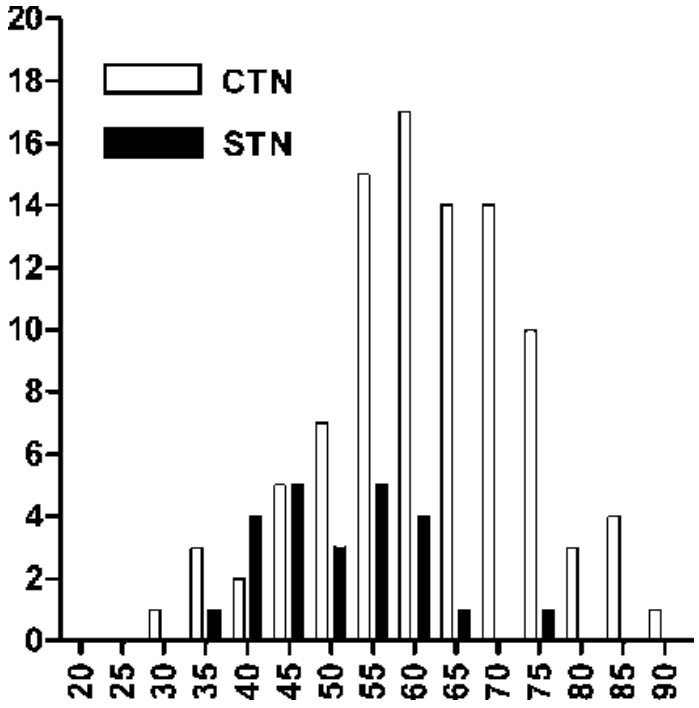

Fig. 1 Frequency distribution of onset age in 120 patients with CTN (n=96) and STN (n =24) trigeminal neuralgia. y axis = number of patients; x axis = years.

Source:[7]

1.1.2 Mechanism

Trigeminal neuralgia is caused by stimulation of the trigger zones of the second and third branches of the trigeminal nerve, which are located around the lips, nasal cavity, and cheeks. An overview of the trigeminal nerve is shown in Fig. 2. The latency between stimulation of the trigger zone and the onset of pain and the refractory period after a pain attack have raised the possibility that the central nervous system is involved in the pathogenesis of trigeminal neuralgia. Since then, many similar cases have been reported, and it has been shown that trigeminal neuralgia subsides when the blood vessels compressing the trigeminal nerve at the trigeminal nerve origin in the cerebellar bridge angle are decompressed by treatment or other means [10][11]. An MRI image of a blood vessel compressing the trigeminal nerve is shown in Fig. 3. Typical trigeminal neuralgia is currently believed to be caused by compression of the trigeminal nerve by surrounding blood vessels or tumors and is included in the criteria for typical trigeminal

neuralgia in the International Classification of Headache, Third Edition. The vessels compressing trigeminal neuralgia are often arteries, most commonly the superior cerebellar artery, but also the basilar artery and the anterior inferior cerebellar artery. However, there are also cases in which veins cause compression [12], and it is thought that, as with arteries, compression is influenced by the pulsation of the blood vessels.

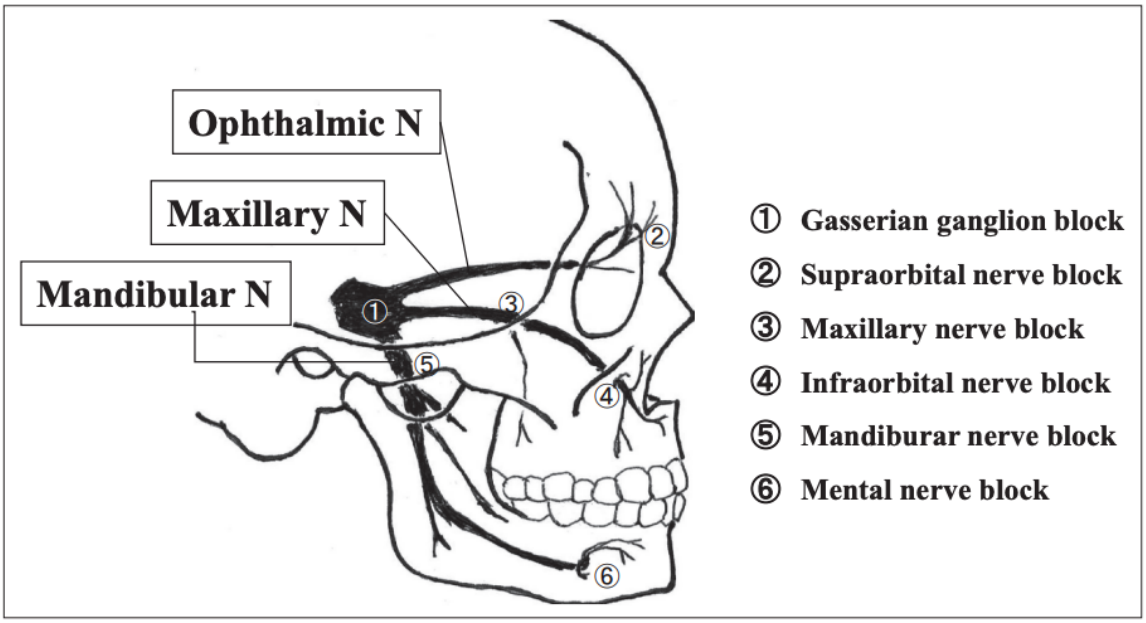

Fig. 2 Overview of the trigeminal nerve (N: Nerve)

Source:[13]

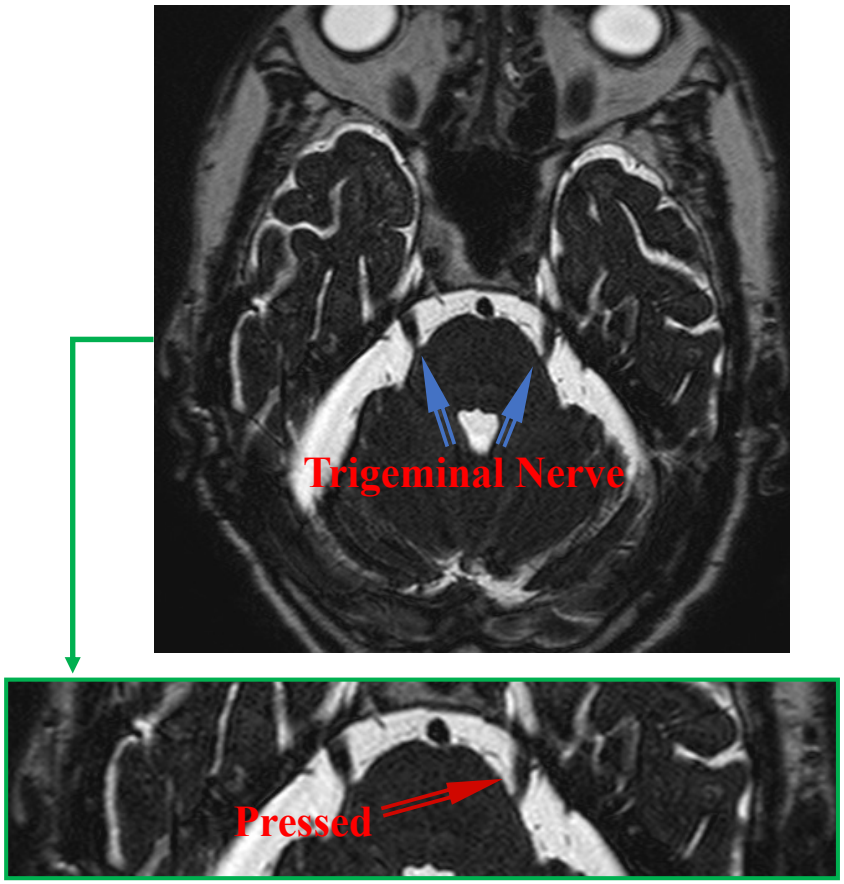

Fig. 3 MRI image of a blood vessel compressing the trigeminal nerve

1.1.3 Diagnosis and Treatment

Currently, the diagnosis is based on a medical interview and imaging studies. However, in order to differentiate trigeminal neuralgia from other facial pain-causing diseases, imaging is often used to diagnose the disease. In imaging, MRI cisternal imaging is used to detect the location of nerve compression. Treatment methods for trigeminal neuralgia include drug therapy, surgical treatment such as microvascular decompression, and gamma knife therapy. The first choice among these treatments is drug therapy [14].

1.2 Artificial Intelligence and Medical Field

Artificial intelligence (AI) experienced its third AI boom in the 2000s with the advent of big data, machine learning, and deep learning. Today, AI is used in numerous fields. The

application of AI in the medical field began in the late 1970s in the field of medical expert systems and developed in the 1980s [15]. In the current medical field, AI has been used for knowledge-based applications, image/video data, and biomedical measurement data, among which many studies on AI technology applications using image data were conducted in Japan [16]. Among them, many studies using AI technology with image data have been conducted in Japan [16]. Reasons for the expected application of AI in the medical field include the prevention of human error, the resolution of personnel shortages in the medical field, and the ease of diagnosis in remote areas such as isolated islands and in home medical care. However, ethical, legal, and social issues still exist [17]. In addition, it has been pointed out that AI is a black box in the medical field. Therefore, it is necessary to consider the positioning of AI utilization with limited scope of medical roles and responsibilities. In this study, we focused on the use of AI for trigeminal neuralgia, for which imaging diagnosis has been established as the main diagnostic method, and decided to introduce AI as a diagnostic aid system in imaging diagnosis.

1.3 Research Objectives

The purpose of this study is to construct an automated system for image diagnosis of trigeminal neuralgia. Therefore, in this study, we examine and propose an image analysis method using deep learning for AI diagnosis as fundamental research in this field. We discuss the learning model with the practice of identifying an appropriate deep learning model.

2 Image Analysis Method

2.1 Annotation

The original image is extracted into a training image using OpenCV, an open-source library specialized for image processing and analysis. The original image is used to create a binarized image in which the trigeminal area is white (pixel value: 0) and all other areas are black (pixel value: 255). The binarized image is overlapped with the original image and extracted from the center of the white area to a size of 60 mm in height and width. The extracted image is used as the training data image. Annotation overview is shown in Fig.4.

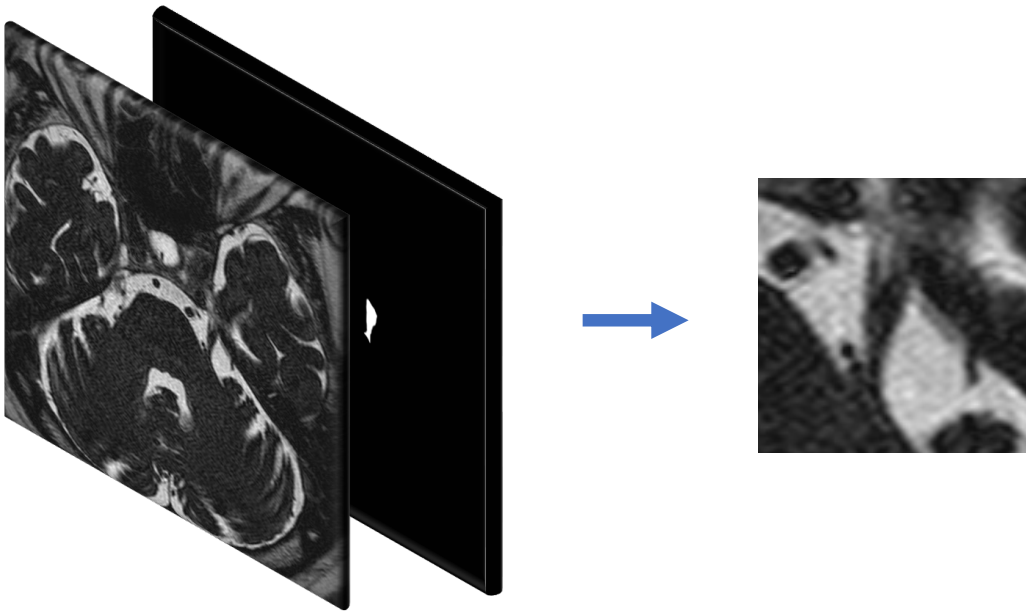

Fig. 4 Annotation overview (left: overlap between binary image and original image, right: extracted image)

2.1.1 Augmentation

Deep Learning requires a large amount of data (number of images) for training because of the huge number of parameters it has. However, in the medical field, the number of data tends to be relatively small because of privacy concerns. Therefore, in order to learn sufficiently with a small number of data, this study employed Data Augmentation [18], which is a method of creating a large number of completely different data by inverting, changing brightness, rotating, translating, and merging from an existing data set, thereby increasing

the amount of data by a factor of several to several dozen. Data augmentation is also expected to avoid overfitting, improve model robustness, reduce model sensitivity to images, and avoid sample imbalance. Excessive use, however, can affect model performance by destroying key property information and creating augmented images with incorrect or ambiguous labels. Therefore, in this study, we used two methods of data expansion: flipping the data upside down and left to right, and rotating and sliding the data.

2.2 Convolutional Neural Network

This study employed convolutional neural network (CNN), which are among the deep learning models that are commonly used for image analysis. We used three CNN models that are relatively easy to handle and compared the accuracy of each model.

LeNet [18] developed by Yann Lecun et al. is the prototype for CNN, which consists of a convolutional layer and a pooling layer. LeNet overview is shown in Fig.5. The convolution layer can process data while preserving the shape of the data (3D data in the case of image data). While the general all-join layer processes data as one- dimensional data, the convolutional layer allows data to be output without losing spatial information. Convolution create feature maps by applying filters (kernels) to the input data. In this research3 × 3 filter is applied to the input data. An example of convolution is shown in Fig. 6. The pooling layer performs operations such as the average and the maximum on the elements of the output data obtained in the convolutional layer. Therefore, there are no parameters to be learned, and the number of channels does not change.

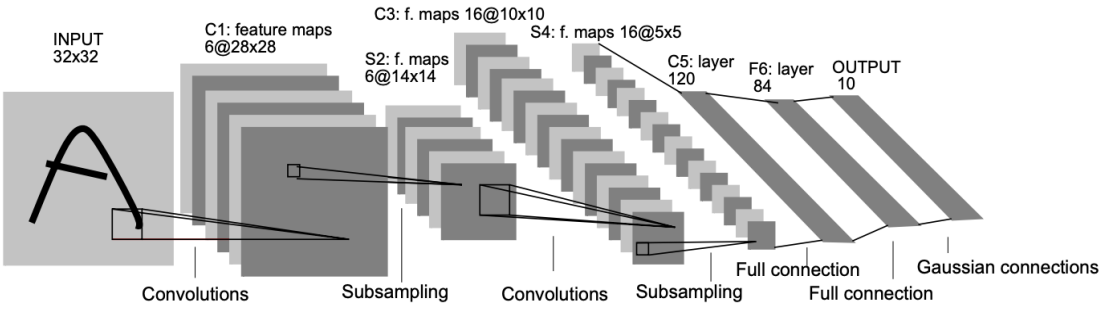

Fig. 5 Architecture of LeNet a Convolutional Neural Network

Source:[18]

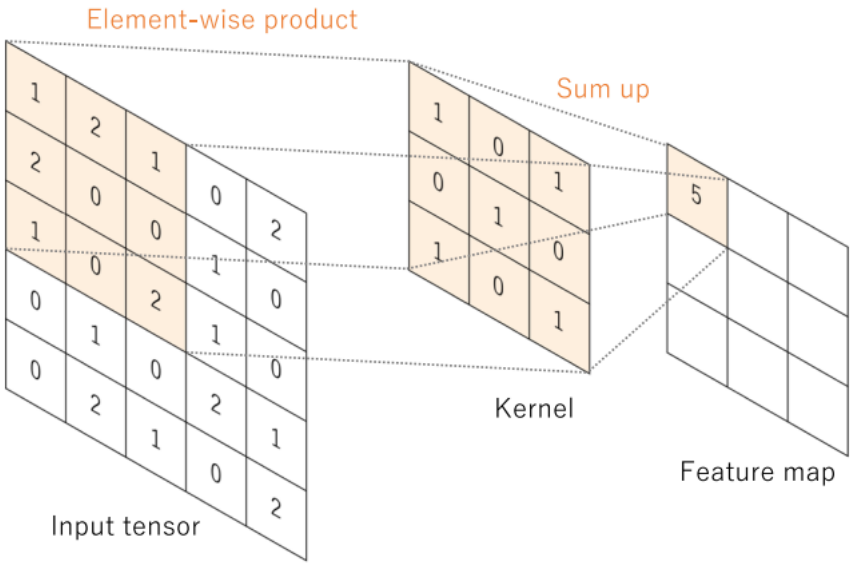

Fig. 6 Convolution

Source: [19]

2.2.1 VGG 16

VGG16 is a CNN model proposed by Simonyan Karen [20] et al. that consists of 16 layers (13 convolutional layers and 3 all-connected layers) and blocks (blocks) in each pooling layer. The VGG16 structure in this study is shown in Fig. 7. 3 × 3 The number of filter channels in the convolutional layer is 64 (denoted by conv3-64), which is doubled for each block. Pooling is performed by five max pooling layers. Max pooling is performed in a 2 × 2 pixel window with a stride of 2. The max pooling layer is followed by several convolutional layers, and finally the output of the convolutional layers is converted to a full concatenated layer (denoted by FC) with 4096 channels.1 × 2 The output of the convolutional layer is converted to a 3-dimensional array and then subjected to binary classification by the Softmax function. In addition, the1 × 1 convolutional layer, but in this study, we use a filter in the3×3 In this study, we used the VGG16 model of convolutional layer filtering. All hidden layers are equipped with ReLU functions.

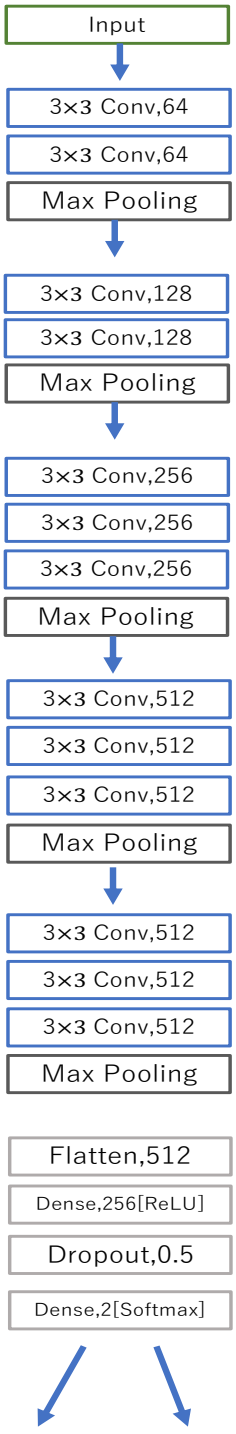
Input
3×3 Conv,64
3×3 Conv,64
Max Pooling
3×3 Conv,128
3×3 Conv,128
Max Pooling
3×3 Conv,256
3×3 Conv,256
3×3 Conv,256
Max Pooling
3×3 Conv,512
3×3 Conv,512
3×3 Conv,512
Max Pooling
3×3 Conv,512
3×3 Conv,512
3×3 Conv,512
Max Pooling
Flatten,512
Dense,256[ReLU]
Dropout,0.5
Dense,2[Softmax]

Fig. 7 VGG16

2.2.2 VGG 19

VGG19, like VGG16, is a CNN model proposed by Simonyan Karen [20] et al. It consists of 19 layers (16 convolutional layers and 3 all-connected layers) and a pooling layer. The VGG19 structure in this study is shown in Fig. 8. VGG19 also corresponds to 3D RGB image data, and the number of filter channels in the 3 × 3 convolutional layer is 64, which is doubled in each convolutional layer block. In VGG19, four convolution layers are configured at Block3~Block5. Each pooling layer consists of a Max pooling layer. The output of the convolution layer is transformed to 1 × 2 dimensional over 4096 channels in all the coupled layers, and finally binary classification is performed by Softmax function. Similarly, in VGG19, it consists only of filters in the 3 × 3 convolution layer, with ReLU in all hidden layers.

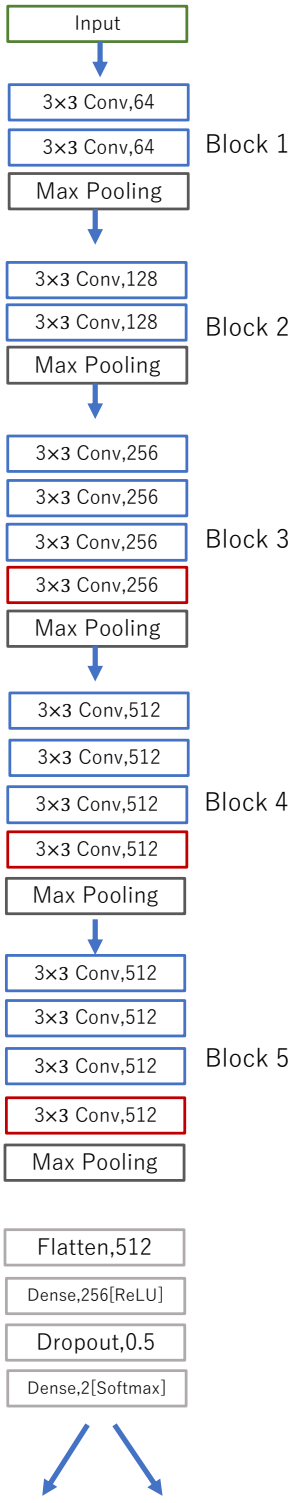

Fig. 8 VGG19

2.2.3 ResNet 50

Resnet50, proposed by Kaiming He et al [21], is a CNN model with 50 layers in depth. Although the accuracy was improved by increasing the depth of the layers, the effect of

increasing the depth of the layers caused the gradient disappearance problem and the degradation problem, which resulted in poor learning. Various approaches, such as activation functions, have been used to solve the gradient loss problem, but focusing on the degradation problem, the Residual Network (ResNet) was devised as a network architecture that can learn even with deep layers. To address this degradation problem, a deep residual learning framework is introduced. Rather than expecting each of several stacked layers to directly fit the desired mapping, these layers are explicitly adapted to the residual mapping. Residual learning in the deep residual learning framework is shown in Fig. 9. assuming that it is optimal to learn F(x)=x and the identity mapping, the parameters w of the nonlinear function F needs to be adjusted to learn the identity mapping, based on the consideration that this is difficult and may cause degradation problems. A detour called Shortcut Connection or Identity Mapping was added, and F(x)+x was configured to be the output. In this case, learning the identity mapping is simpler than in the former case, since the parameters need to be learned so that F(x)=0, i.e., w=0. The Residual Network is a network of multiple layers of residual blocks, called a residual block (building block), consisting of several convolutional layers and shortcut connections. The architecture of the Residual Network is shown in Fig. 10. In this study, we adopted the Bottleneck architecture's Residual block, which consists of 1×1, 3×3, and 1×1 convolutional layers, to create a 50-layer Resnet. The Bottleneck architecture is shown in Fig. 11.

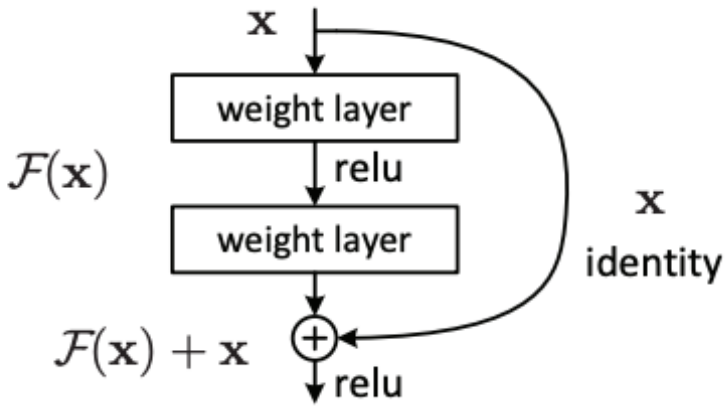

Fig. 9 Residual learning: a building block

Source:[21]

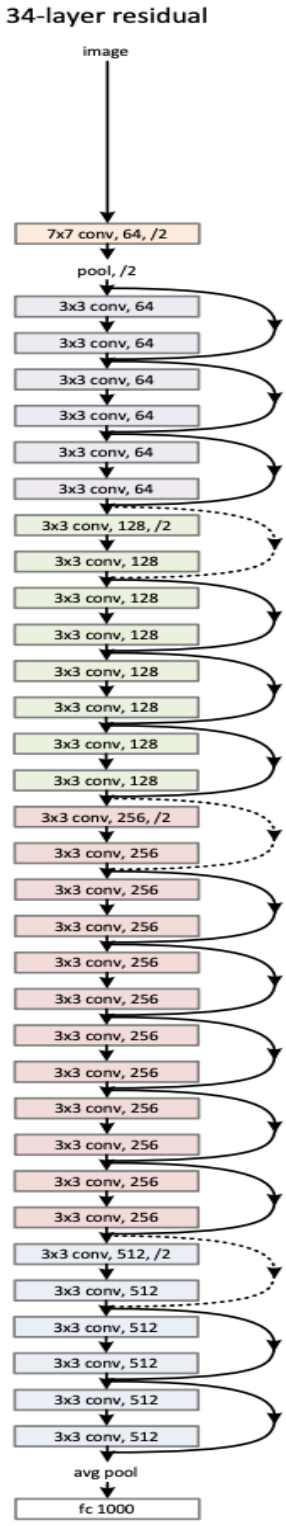

Fig. 10 Residual Network

Source: [21]

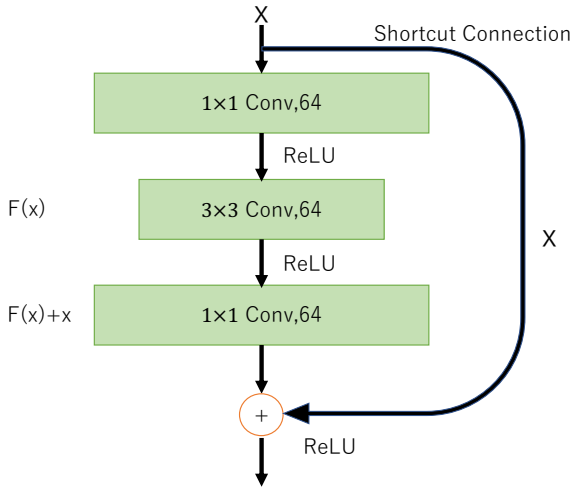

Fig. 11 Bottleneck

2.3 Transfer Learning

Transfer learning is an efficient learning method in which what has been learned in one domain (learned model) is applied to another domain. This reduces learning time and computational cost (parameter updates) compared to learning a model from scratch. It can learn efficiently even with a small amount of data, thus achieving stable accuracy. The pre-training model is based on ImageNet, which has been trained on a large dataset. In VGG16, we fix weights up to layer 15 and retrain the following layers. Resnet50 fixes weights up to layer 143 and retrains the subsequent layers.

2.4 Grad-CAM

In the medical application of AI, high reliability is required for the results output by AI. while AI can be used for complex tasks that require flexible responses, its flexible processing has left the issue of not being able to clearly present the basis for the processing results. Therefore, accountability and transparency of AI decisions are required. Against this background, there is a trend toward the demand for Explainable AI in medical applications. In this study, we applied Grad-CAM [22] developed by Ramprasaath R. Selvaraju et al. to visualize the basis for AI decisions. Grad-CAM is a method for displaying the identified points in an image as a heat map for a given input and its prediction for a CNN-based image

recognition model. Grad-CAM is a generalization of CAM heatmap computation that is not limited by model constraints. The overview of Grad-CAM is shown in Fig.12. Grad-CAM is a generalized version of the CAM heat map calculation that is not limited by model constraints. Images and classes of interest are given as input, the images are forward propagated through the CNN part, and then through task-specific computation to obtain the raw scores for the categories. The gradient is set to zero for all classes except the desired class. The signal is back-propagated to create a parallelized convolutional feature map of interest, which is combined to compute a coarse Grad-CAM localization. The Grad-CAM localization (blue heatmap) represents where the model must look to make a particular decision. Finally, the heatmap is crossed with the original image to obtain a Grad-CAM visualization. An example using Grad-CAM is shown in Fig. 13.

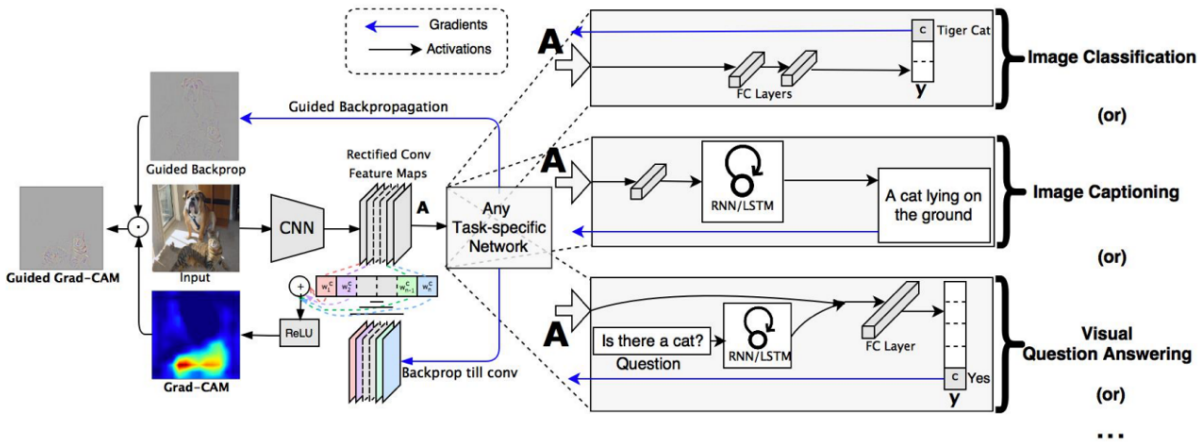

Fig. 12 Grad-CAM overview

Source:[22]

Fig. 13 Example of Grad-CAM visualizing the basis for bike decisions

Source:[22]

2.5Experimental data

MRI images of the left and right trigeminal nerves were obtained from a total of 43 donors: 28 trigeminal neuralgia positive (5 males, 20 females, 3 unknown) and 15 trigeminal neuralgia negative (5 males, 10 females). A total of 240 non-trigeminal and trigeminal neuralgia images (A set), a data set of 960 images (B set), and a data set of 2400 images (C set) were created for a total of three different data sets. A table of the datasets is shown in Tab. 1. Eighty percent of the dataset is used as the training dataset, the remaining 10% as the validation dataset, and the remaining 10% as the test dataset.

The training dataset is used to train the model and the validation dataset is used to adjust the piper parameters of the training model. Finally, the training model is evaluated using the test dataset.

Tab. 1 Dataset

	Positive	Negative	Sum
A set	120	120	240
B set	480	480	960
C set	1200	1200	2400

2.6Program experiment protocol

For the experiments, Python 3.0 was used for CNN model building. Keras2.0 and scikit-learn libraries were mainly used on TensorFlow2.0. Batch size was set to 32 and epoch to 300. The experiments were conducted using binary classification, but “categorical_crossentropy”, which showed the highest accuracy, was used for the loss function. The learning rate was fixed at 0.00005 with no change from epoch to epoch. The image size was set to80 × 80 mm.

2.7Evaluation Method

The performance of the CNN model is evaluated from Accuracy, Precision, Recall, Specificity, F-score, AUC (ROC curve: Receiver Operating Characteristic curve) based on the Confusion Matrix of True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN), with two classifications: true positive (TP), false positive (FP), false negative (FN), and true negative (TN). The Confusion Matrix is shown in Tab. 2.

Tab. 2 Confusion Matrix

		Predicted value	
		Positive	Negative
Correct value	Positive	TP (True positive)	FP (False positive)
	Negative	FN (False negative)	TN (True negative)

Accuracy is simply the number of correct classifications in the overall data. Accuracy is used to adjust the hyperparameters, mainly when using Validation data. The formula for Accuracy is shown in (1).

$$Accruracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

Precision indicates the probability that the positive answer is actually the correct answer among the predicted positives. Therefore, it is an indicator to prevent false positives. It is an important index in precision testing in the medical field. Equation (2) of Precision is shown below.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

Recall (Sensitivity) indicates the probability that among all data for which the correct answer (true value) is "Positive", the prediction by the machine learning model is also "Positive" and correct. Positive indicates the probability that the prediction by the machine learning model is "Positive" and the true value is also "Positive" and correct. Indicates the degree to which the prediction by the learning model reproduces the correct answer when the correct answer is "Positive". In the medical field, the evaluation of recall is important because false positives can be problematic. The formula for Recall is shown in Equation (3).

$$Recall = \frac{TP}{TP + FN} \quad \textbf{(3)}$$

The F-score (Dice coefficient) is an indicator for evaluating Precision and Recall together. The F-score is used to evaluate both Precision and Recall, and is used when comprehensively judging whether the desired detection is made. It is the most efficient and well-balanced machine learning model. Equation (4) of the F-score is shown.

$$F = \frac{TP}{TP + \frac{1}{2} * (FP + FN)} = \frac{1}{\frac{1}{Precision} + \frac{1}{Recall}} \quad \textbf{(4)}$$

The ROC curve is shown as a graph with the True Positive Rate (TPR) on the vertical axis and the False Positive Rate (TPR) on the horizontal axis. The range of both the vertical and horizontal axes is 0.0~1.0, and the points corresponding to the threshold values are placed on these axes. On the ROC curve, a high recall and a low false positive rate are indicated by the upper left point. In other words, the higher the ROC curve is to the upper left, the higher the discrimination performance. Therefore, it is one of the most important evaluation methods for diagnosis in the medical field.

AUC (Area Under Curve) is a numerical indicator of the ROC curve and refers to the area under the ROC curve between 0.0 and 1.0. The maximum value is 1.0, which can be indicated as recall 1.0 and false positive rate 0.0 if the judgment is perfect.

3 Experimental results

3.1 Results in CNN

We prepared three types of datasets, A set, B set, and C set, from data augmentation by creating images in which disease regions were detected using annotation from the aforementioned MRI images. Each dataset was divided into training data (including validation data) and test data. Training was performed on the three CNN models, VGG16, VGG19, and Resnet 50, using the training data, and model performance was evaluated based on results obtained from the test data.

3.1.1 Training process

During training, the training was performed under the conditions of the aforementioned experimental protocols. Fig.14, Fig.15, and Fig.16 show the learning process for each dataset of VGG16. The horizontal axis represents Epoch, and the vertical axis represents Accuracy. The red line shows the process on the training data, and the blue line shows the process on the validation data. In B and C set, as the number of images increases, the accuracy improves compared to A set. In addition, while there was a behavior (variation) in accuracy during training in B set, it can be seen that the behavior during training converges in C set. This is because the number of images increases as the number of images increases, and the accuracy of B and C set increases as the number of images increases. This is thought to be due to the fact that the accuracy of the validation data was improved by the increase in the number of images, which allowed more features to be captured from the images than in the A and B set, which had a relatively small number of images.

The learning process of VGG19 on each dataset is shown in Fig.17, Fig.18, and Fig.19. As in VGG19, in the A set, learning reaches a certain relatively low level of accuracy, and learning convergence on the training data is slow. Compared to VGG16, the training converges and the validation process shows high accuracy. However, in the C set, there is less variability in the validation process than in the B set, but the accuracy is lower. It is thought that over fitting occurred because the data acted as excessive noise for VGG19,

which is a more complex model (more parameters) than VGG16. Therefore, a 4-fold data augmentation is appropriate for VGG19.

Fig.20, Fig.21, and Fig.22 show the training process for each dataset of ResNet50. The accuracy of the training data as well as the validation data varied, and the training did not converge well. However, as the number of image data is increased, the accuracy variation becomes smaller. In addition, the deeper the layers, the smaller the size. Therefore, it is difficult to extract features in a generalized manner for images with an input size of 80 × 80. It is believed that the accuracy is low (does not increase) due to these reasons.

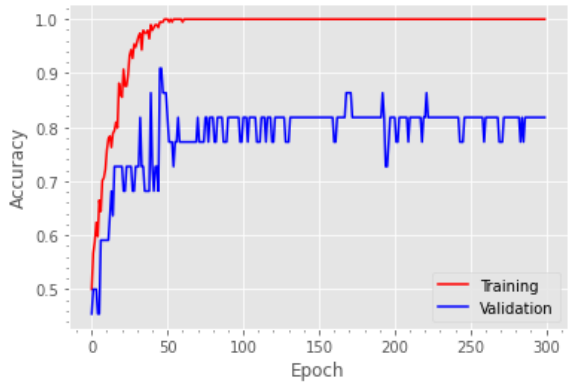

Fig. 14 Learning process with A set (VGG16)

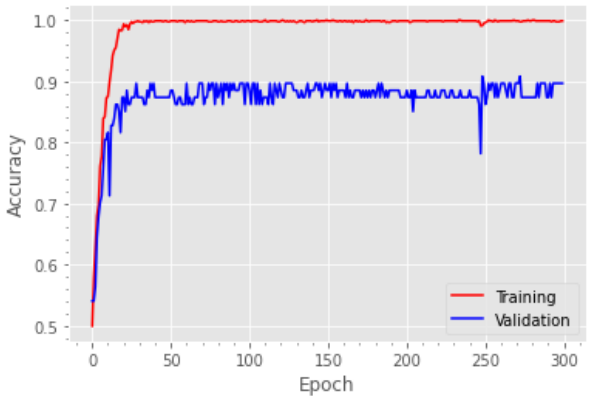

Fig. 15 Learning process with B set (VGG16)

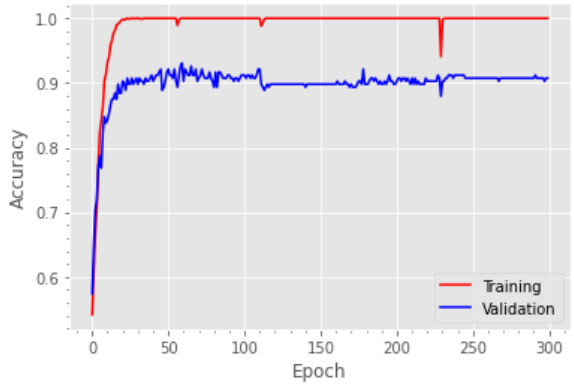

Fig. 16 Learning process with C set (VGG16)

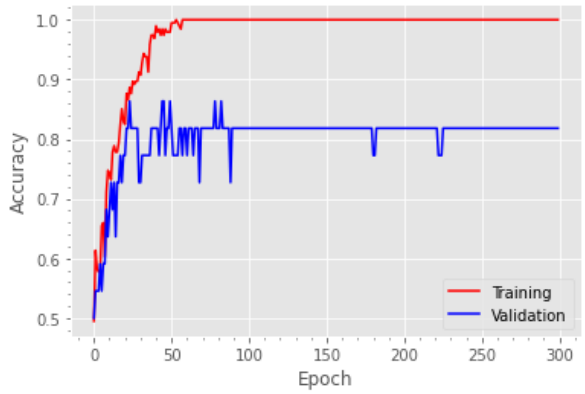

Fig. 17 Learning process with A set (VGG19)

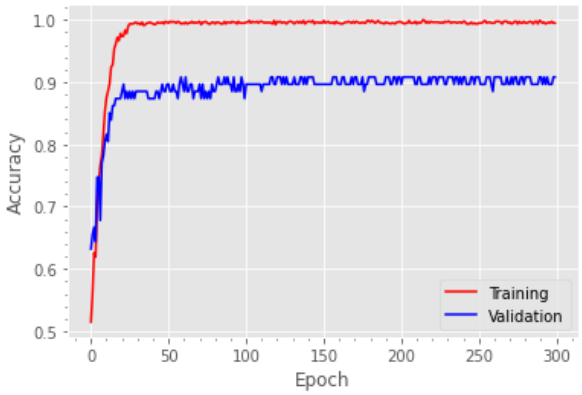

Fig. 18 Learning process with B set (VGG19)

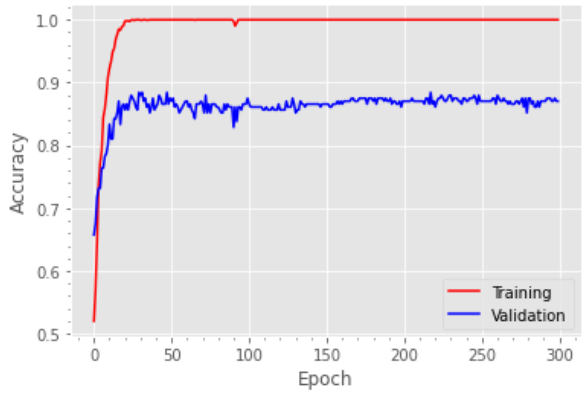

Fig. 19 Learning process with C set (VGG19)

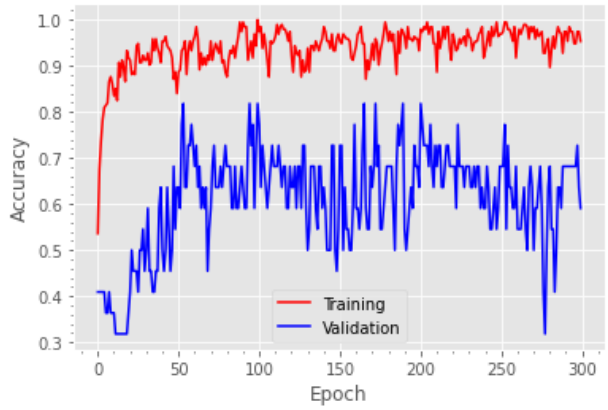

Fig. 20 Learning process with A set (ResNet50)

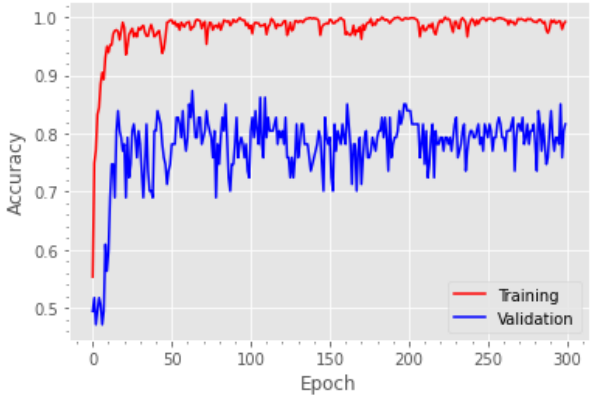

Fig. 21 Learning process with B set (ResNet50)

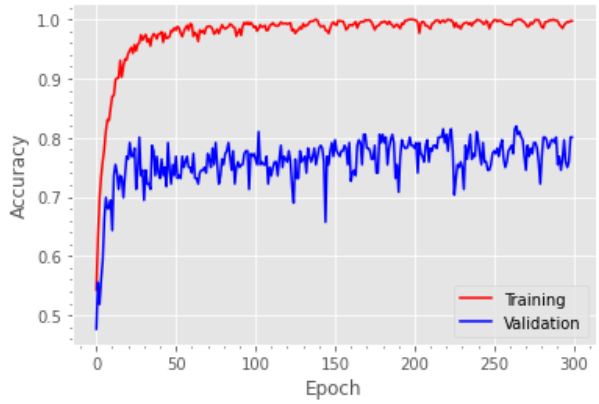

Fig. 22 Learning process with C set

(ResNet50)

3.1.2 Loss function

Fig. 23, Fig. 24, and Fig. 25 show the process of the loss function for each dataset in VGG16. In A set, the loss values for the validation data diverge without decreasing after epoch 50, indicating that the learning process is not going well. In the B set, the loss value rises to around 1.0 and then drops to around 0.4, indicating that the excessive divergence of the loss value is suppressed, although some variation is observed. However, compared to the loss value of the training data, the loss value of the validation data is large, indicating that overlearning has occurred, which is also the case in the B set. Although the loss values are relatively small, it can be said that overlearning is also occurring in the C set.

Fig. 26, Fig. 27, and Fig. 28 show the process of the loss function for each dataset of VGG19. Overall, the loss values of the validation data are lower than those of VGG16, and the trend of increasing values is relatively gradual. The accuracy of the validation data during the training process was not significantly different from that of VGG16, but the loss values showed a large difference. Compared with VGG16, the accuracy of the validation data during the learning process was not significantly different from that of VGG16, as in the A set. In comparison with VGG16, the accuracy of the validation data during the learning process was not much different from that of VGG16, but the loss values were low and did not vary greatly, indicating that the learning process was relatively stable. Therefore, the accuracy of the validation data is also lower in the C set than in the B set.

Fig. 29, Fig. 30, and Fig. 31 show the process of the loss function for each of the ResNet50 data sets. Although the variation of the loss value is suppressed as the number of image data increases, the overall loss value does not decrease as the number of data increases, as is the case with other models. As described in the training process in 3.1.1, it is not possible to extract features in a generalized manner.

Fig. 23 Loss process with A set (VGG16)

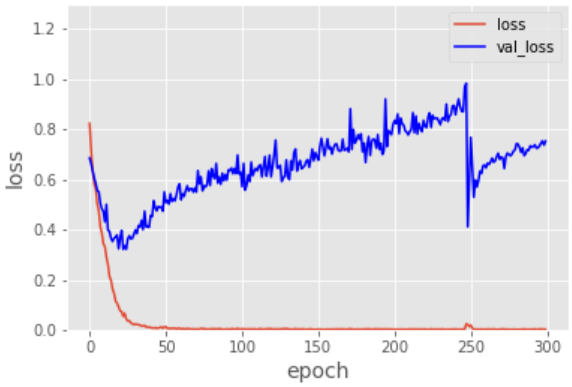

Fig. 24 Loss process with B set (VGG16)

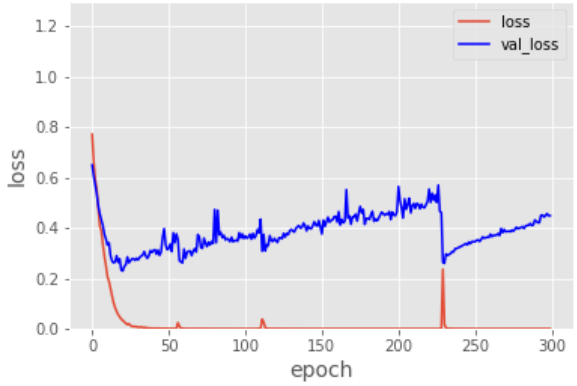

Fig. 25 Loss process with C set (VGG16)

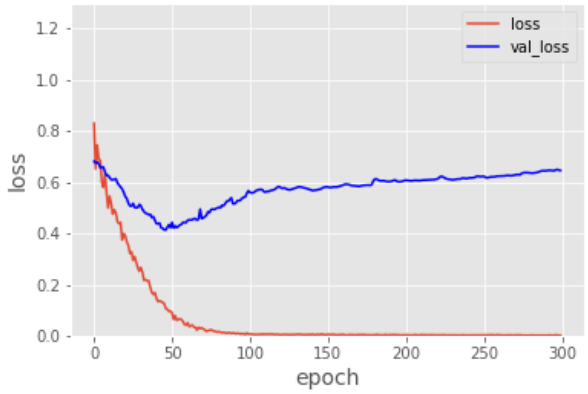

Fig. 26 Loss process with A set (VGG19)

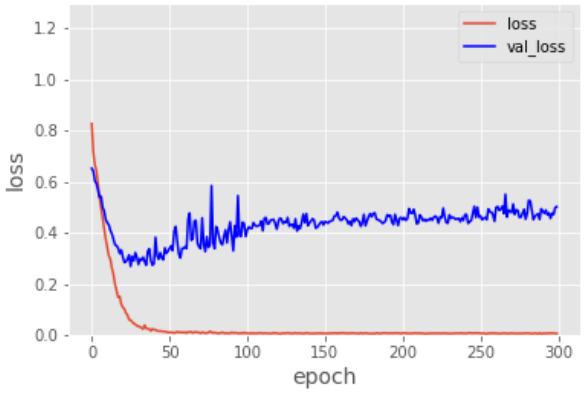

Fig. 27 Loss process with B set (VGG19)

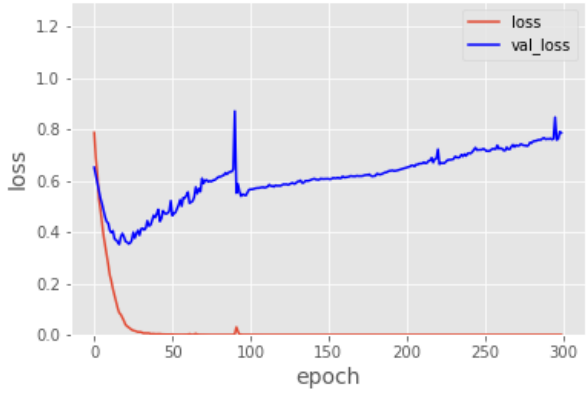

Fig. 28 Loss process with C set (VGG19)

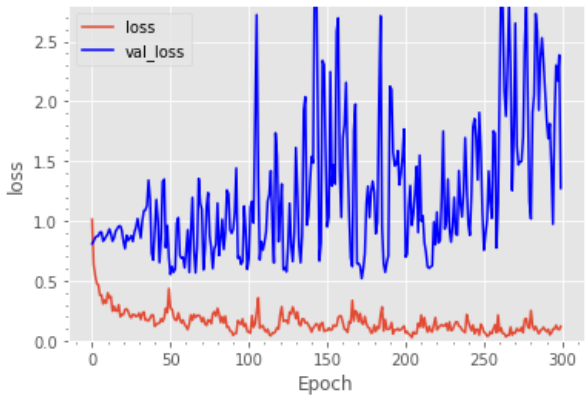

Fig. 29 Loss process with A set (ResNet50)

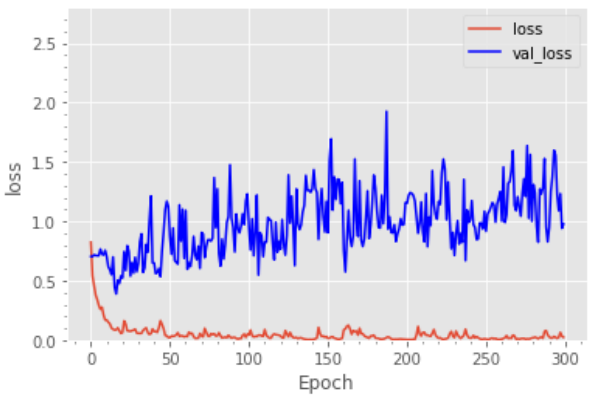

Fig. 30 Loss process with B set (ResNet50)

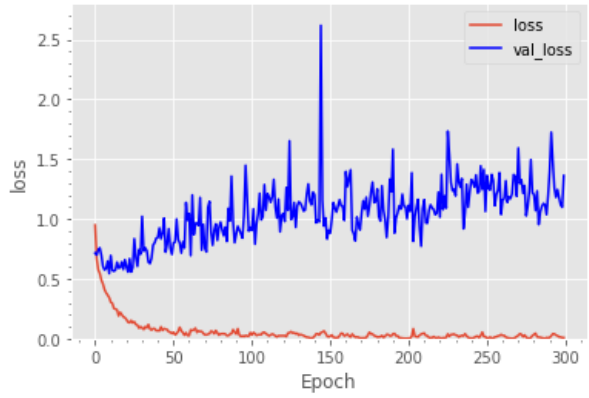

Fig. 31 Loss process with C set

(ResNet50)

3.1.3 Evaluation on test data

The results (accuracy, precision, recall, F1-score, auc) for each test data in VGG16 are shown in Tab.3. Tab.3 shows that the results for C set are accuracy: 0.942, precision: 0.958, recall: In comparison between the C set and the A set, the accuracy was 0.109, precision 0.14, recall 0.108, F1-score 0.124, and auc 0.115 for accuracy. The overall accuracy improved as the number of image data increased. These results confirm the effectiveness of data augmentation in VGG16. To visualize the evaluation of VGG16 on the best performing C set, a confusion matrix is shown in Fig. 32. 14 of the 240 test images are mispredicted, and Fig. 32 shows that 5 of the 14 images are misclassified as false positives and 9 as false negatives. Fig. 32 shows that out of the 14 images, 5 are false positives and 9 are false negatives.

Tab.4 shows the results for each test data in VGG19. Tab.4 shows that the B set has the best performance in recall with recall: 0.920, F1-score: 0.906, and auc: 0.964, respectively. Although recall is an important index in the medical field, the overall evaluation showed that VGG19 in the C set had the highest performance in VGG19, since the difference between the C set and the B set was only 0.007 of recall. The overall performance of VGG19 was also improved by increasing the number of data in VGG19, confirming the effect of data augmentation. To visualize the evaluation of VGG19 in the best performing C set, a confusion matrix is shown in Fig. 32. 24 out of the 240 test images are mispredicted, and Fig. 32 shows

that 13 of the 24 images are misclassified as false positive and 11 as false negative. Fig. 32 shows that out of 24 images, 13 are false positives and 11 are false negatives.

Tab.5 shows the results of each test on ResNet50. As with ResNet50, as the number of data increased, the overall evaluation increased, confirming the effect of data augmentation. However, the accuracy variability during the training process is significantly higher than that of other models, so the reliability of the evaluation is considered low. To visualize the evaluation of ResNet50, a confusion matrix is shown in Fig. 34. 47 out of 240 test images are mispredicted. 28 out of 47 images are misclassified as false positives and 19 as false negatives, according to Fig. 34.

Comparing the VGG16, VGG19, and ResNet50 models, the VGG model shows the most stable performance, with VGG16 performing the best in the C set. It showed the best performance in all evaluations and was able to learn with high accuracy from the validation data during training. The effectiveness of data augmentation was also demonstrated for all models.

Tab. 3 Test result with VGG16

		Accuracy	Precision	Recall	F1-score	auc
VGG16	Aset	0.833	0.818	0.818	0.818	0.867
	Bset	0.865	0.824	0.913	0.866	0.973
	Cset	0.942	0.958	0.926	0.942	0.982

Tab. 4 Test result with VGG19

		Accuracy	Precision	Recall	F1-score	auc
VGG19	Aset	0.708	0.688	0.846	0.759	0.699
	Bset	0.885	0.868	0.920	0.893	0.947
	Cset	0.900	0.898	0.913	0.906	0.964

Tab. 5 Test result with ResNet50

		Accuracy	Precision	Recall	F1-score	auc
ResNet50	Aset	0.591	0.625	0.454	0.526	0.699
	Bset	0.792	0.771	0.804	0.787	0.836
	Cset	0.804	0.786	0.844	0.814	0.876

Fig. 32 Confusion Matrix of VGG16 on C set

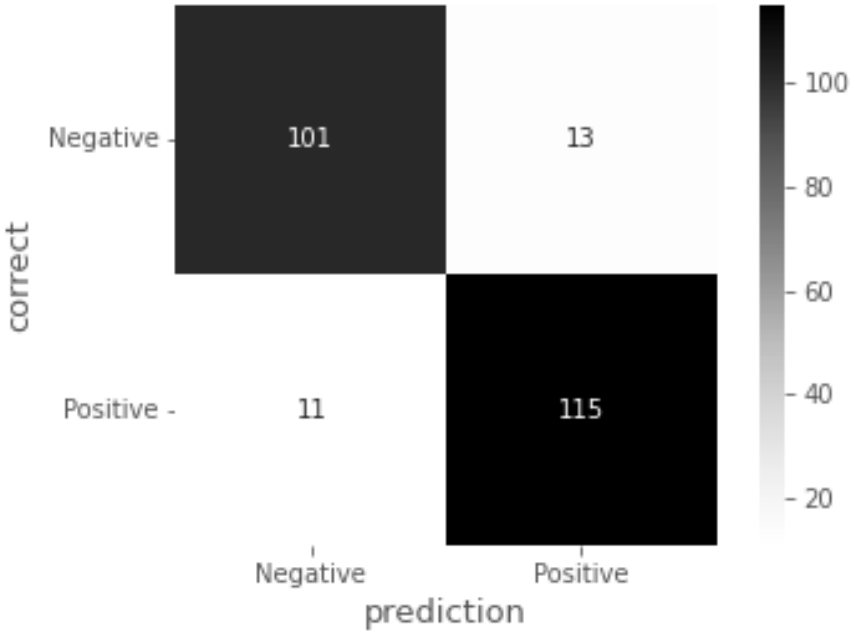

Fig. 33 Confusion Matrix of VGG19 on C set

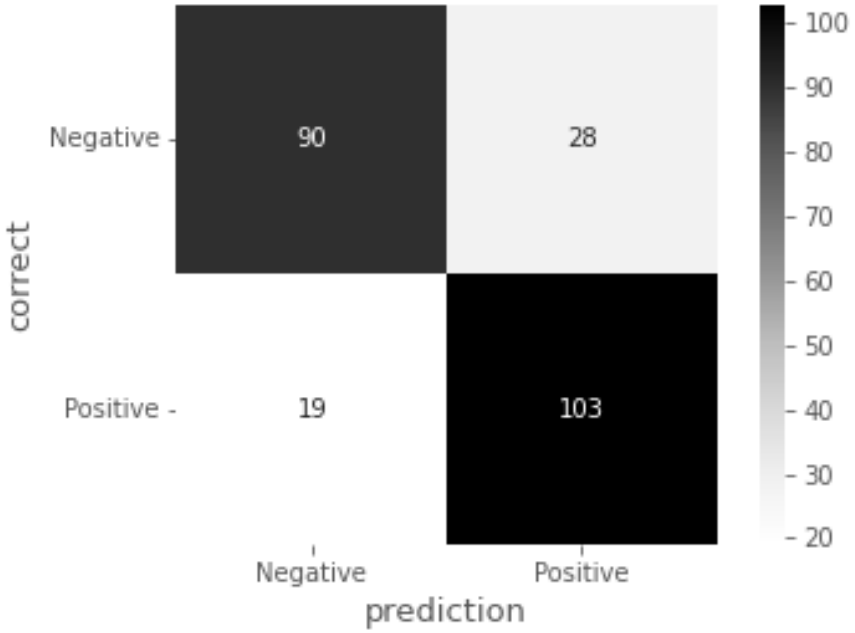

Fig. 34 Confusion Matrix of ResNet50 on C set

3.1.4 ROC curve

The ROC curves for each test data of the three models are shown in Fig. 35, Fig. 36, and Fig. 37. First, Fig. 35, Fig. 36, and Fig. 37 show the ROC curves for the VGG16 test data. Fig. 35, Fig. 36, and Fig. 37 show that the more image data there is in the test data set, the more the curve rises to the left.

Fig. 38, Fig. 39, and Fig. 40 show the ROC curves for each of the VGG19 test data sets. Similarly, for VGG19, although the performance evaluation is low for the A set, the curve increases to the left as the test data set has more image data.

Fig.41, Fig.42 and Fig.43 show the ROC curves for each test data set of ResNet50. In Fig. 42 and Fig. 43, the right-leaning curve is no longer high, but there is no tendency for the curve to rise to the upper left as the number of data increases, which was observed in the VGG model. The AUC is also lower than that of the other models. From these results, it can be concluded that the performance of ResNet50 is low at this stage and that the VGG model has high performance.

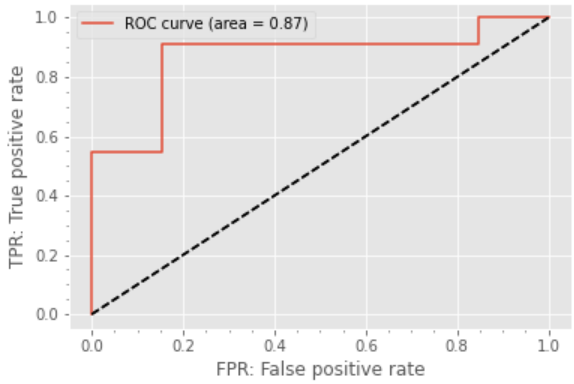

Fig. 35 ROC curve with A set (VGG16)

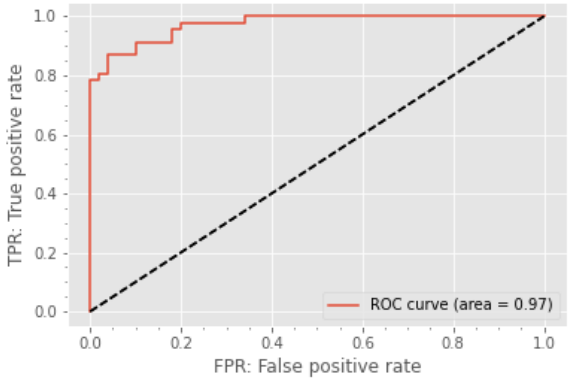

Fig. 36 ROC curve with B set (VGG16)

Fig. 37 ROC curve with C set (VGG16)

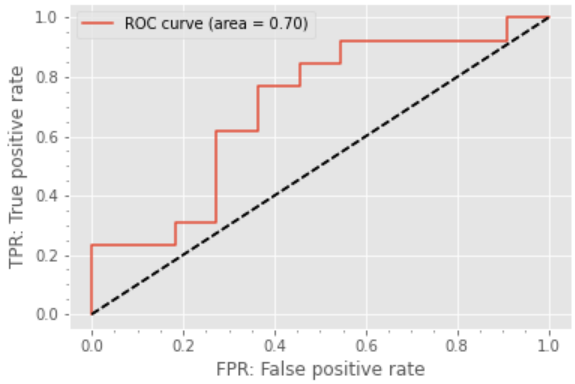

Fig. 38 ROC curve with A set (VGG19)

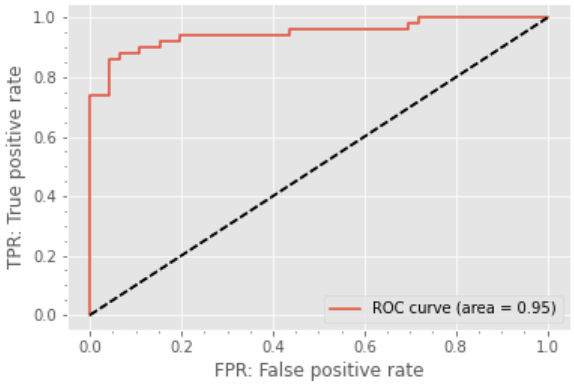

Fig. 39 ROC curve with B set (VGG19)

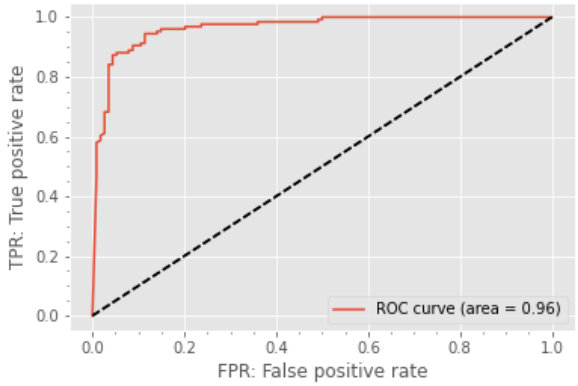

Fig. 40 ROC curve with C set (VGG19)

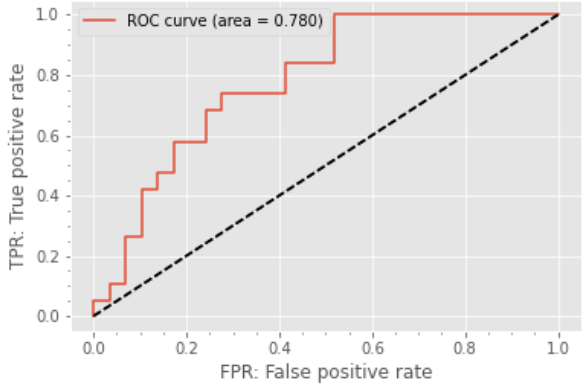

Fig. 41 ROC curve with A set (ResNet50)

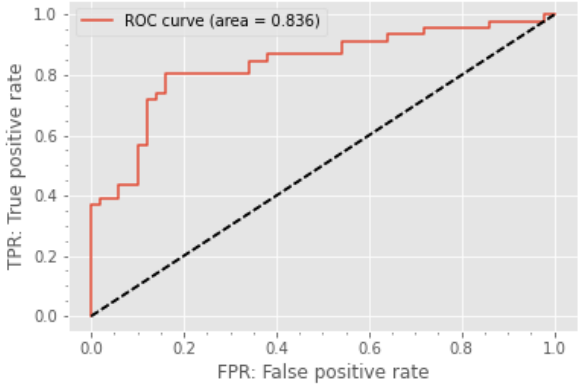

Fig. 42 ROC curve with B set (ResNet50)

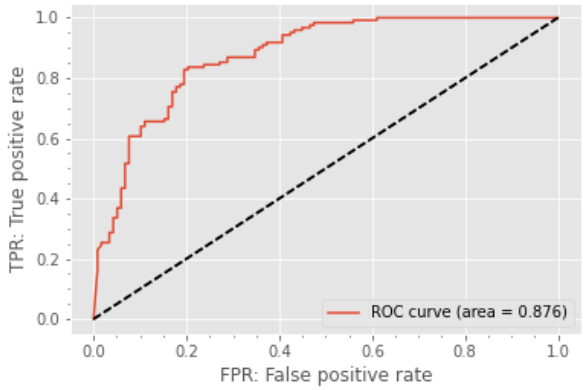

Fig. 43 ROC curve with C set

(ResNet50)

3.1.5 Grad-CAM

Negative and positive image data are prepared, and the basis for judging VGG16, VGG19, and ResNet50 in each data set is visualized using Grad-CAM. An example of a negative image result is shown at the top, and an example of a positive image result is shown at the bottom. The results for each image are, from left to right, the original image, the heatmap, and the image obtained by overlapping the original image and the heatmap (Grad-CAM image).

The Grad-CAM results of VGG16 for each dataset are shown in Fig.44, Fig.45, and Fig.46. Fig.44 shows that no features were captured in the negative images. Fig.45 shows that the negative image fails to capture the trigeminal nerve as in VGG16 of set A. The positive image shows that the trigeminal nerve is captured. Fig. 46 shows that the trigeminal nerve was captured in the C set while it was not captured in the A and B set. It can be said that the discrimination performance was improved by increasing the number of images. In the positive images, the lower part of the trigeminal nerve was captured as in the B set.

Fig.47, Fig.48 and Fig.49 show the Grad-CAM results of VGG19. Fig.48 shows that VGG19 captured the area around the trigeminal nerve while VGG16 did not, indicating that VGG19 has better discrimination performance than VGG16 in the B set. The positive images showed

that the trigeminal nerve was not detected by VGG16. The positive image shows that the area around the trigeminal nerve including the background was captured and widely distributed.

Grad-CAM results of ResNet50 are shown in Fig.50, Fig.51 and Fig.52. Fig.50 and Fig.51 show that the Grad-CAM results for ResNet50 tend to capture a wide range of features, and are not able to capture specific features. Therefore, the background is captured in a wide range, and it can be confirmed that the negative image in Fig. 52 is strongly discriminated from the background. However, in Fig. 52, the trigeminal nerve is captured at the center of the image, which improves the discrimination performance.

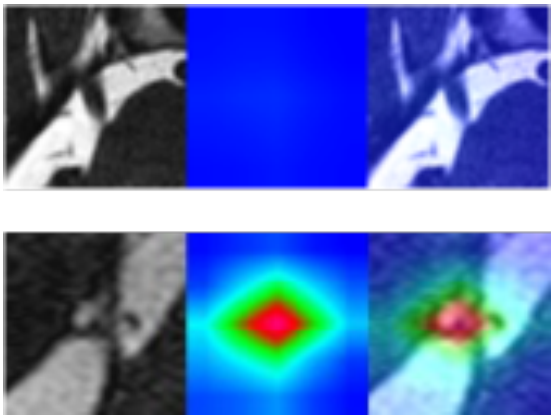

Fig. 44 Grad-CAM of VGG16 on A set

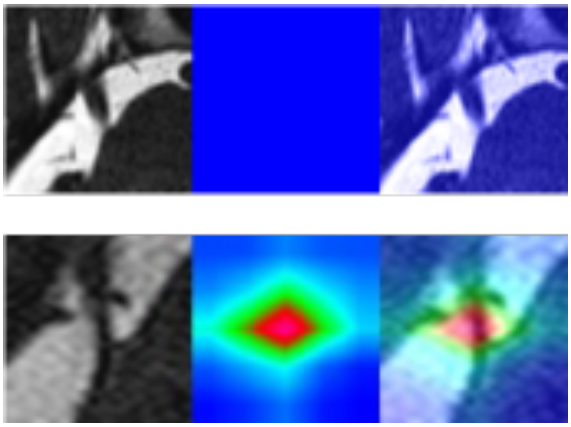

Fig. 45 Grad-CAM of VGG16 on B set

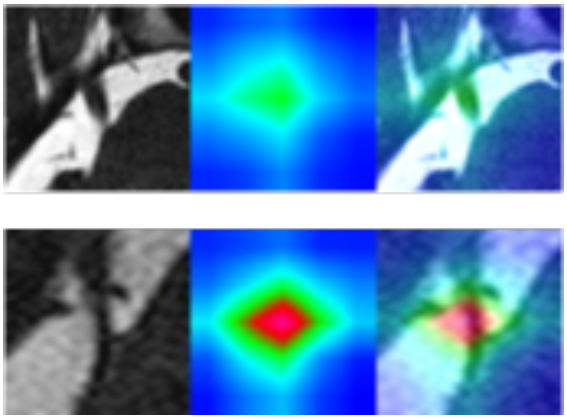

Fig. 46 Grad-CAM of VGG16 on C set

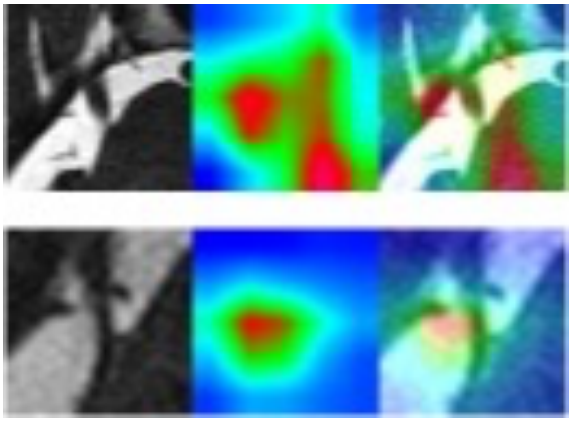

Fig. 47 Grad-CAM of VGG19 on A set

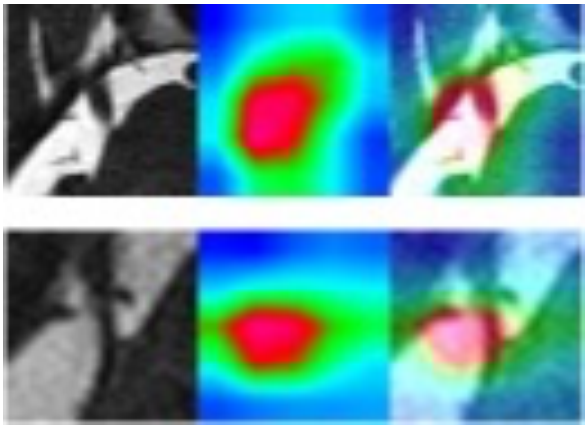

Fig. 48 Grad-CAM of VGG19 on B set

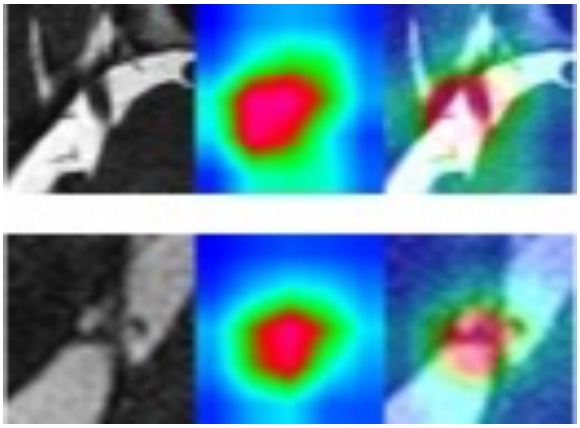

Fig. 49 Grad-CAM of VGG19 on C set

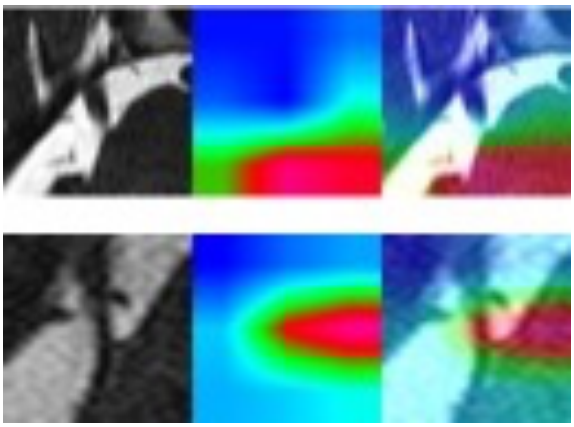

Fig. 50 Grad-CAM of ResNet50 on A set

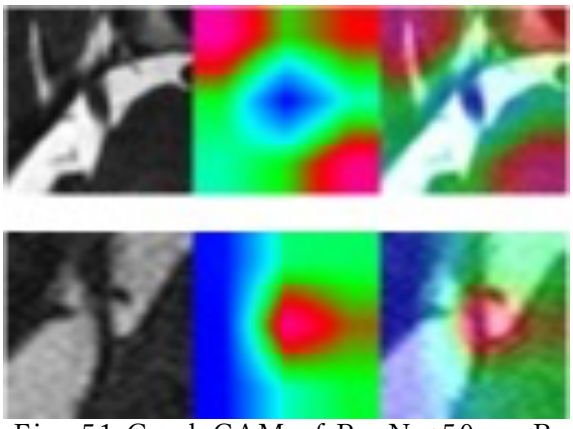

Fig. 51 Grad-CAM of ResNet50 on B set

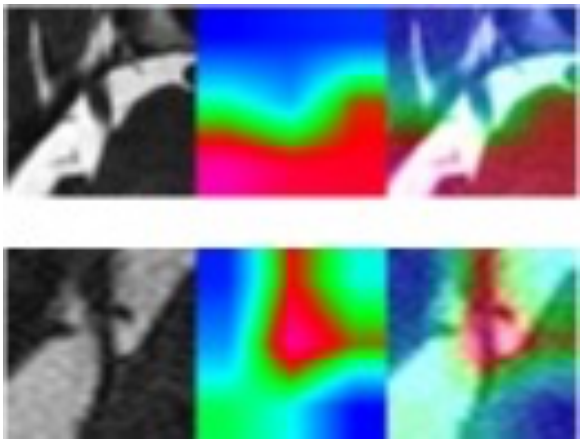

Fig. 52 Grad-CAM of ResNet50 on C set

4 Proposal method of VGG model

The results for the VGG model showed stable performance; however, the performance for the multi-layer model is expected to be lower than for the Resnet50 model, because increasing the number of convolutional layers created a situation where learning did not converge (i.e., accuracy did not increase). Therefore, we focused on the structure of the VGG model and improved its structure in order to propose a more accurate model. To this end, we proposed two methods for improving the structure of the VGG model.

4.1 Support Vector Machine

Support Vector Machines (SVM) are machine learning algorithms introduced by Bernhard E. Boser [24] et al. The goal of the algorithm is to maximize the margin, the distance between the n-1-dimensional plane (not strictly a plane) that classifies n-dimensional data, called the separation hyperplane, and the data closest to that separation hyperplane. The goal of the algorithm is to maximize the margin, which is the distance between the separation hyperplane and the data closest to the separation hyperplane. Based on this idea of margin maximization, the algorithm is mainly good at binary classification. An example of a distribution map identified by the support vector machine is shown in Fig. 53. It shows the case of binary classification (Group A, Group B) of a group of data based on two features (A, B). In this case, the data closest to the separation hyperplane is called the support vector, and its distance is called the margin. By maximizing this margin, the range of discriminative judgments for binary classification becomes wider, and thus, stable accuracy can be achieved. In general, however, it is difficult to achieve perfect discrimination. Therefore, it is necessary to tolerate a certain degree of misclassification, and classification is made possible by setting constraint conditions (constraint formulas) on the margin. This margin is called the soft margin. The maximization of the margin and the kernel method described here enable a support vector machine that can perform nonlinear classification. However, the computational cost of converting all data from all features to vectors and then classifying them is high. The support vector machine also needs to adjust to two piper parameters: the cost parameter (C), which determines how much misclassification is tolerated; the larger C,

the less misclassification is tolerated. The higher C, the less misclassification is tolerated. The second is the RBF kernel parameter (γ), where smaller values of γ result in simpler decision boundaries next to each other and larger values result in more complex decision boundaries. In this study, support vector machines were employed because the number of data handled in training is relatively small and because binary classification is performed. The architecture of VGG+SVM is shown in Fig. 54. The upper VGG convolutional layer blocks are treated as feature extractors, and the lower all-combining layer blocks are converted to SVMs as discriminators. In addition, we changed from Max pooling to Average pooling. Based on the features extracted by the feature extractor, the discriminator SVM performs soft-margin binary classification based on the features.

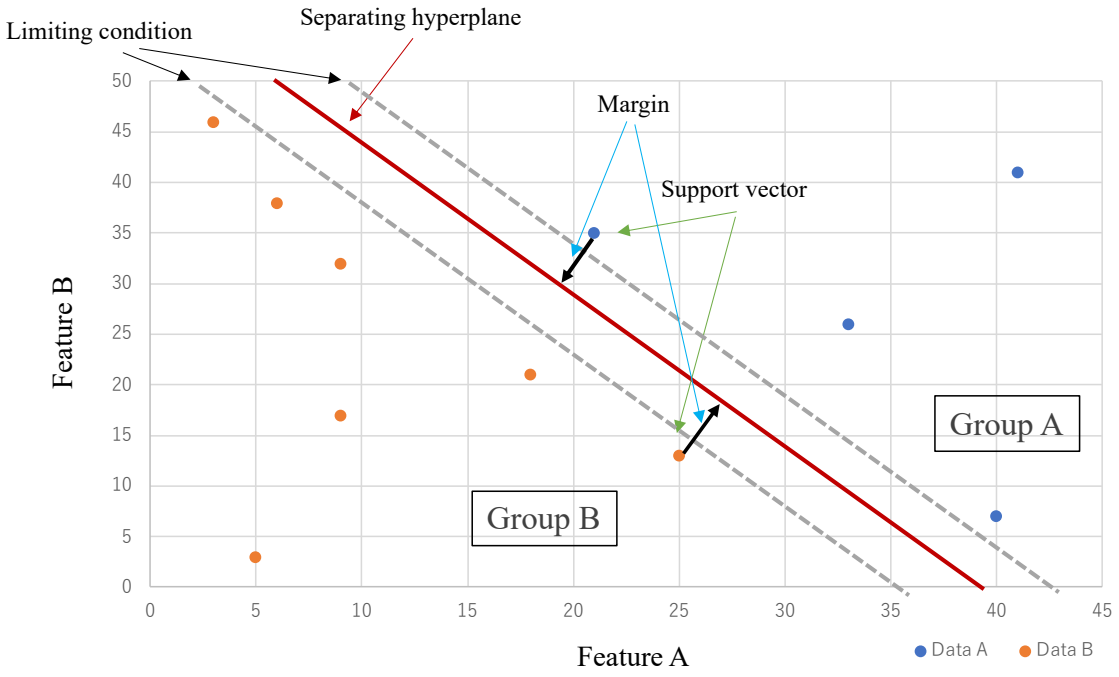

Fig. 53 Example of distribution map classified by support vector machine (Classification is made from the features of Feature A and Feature B. Example of classification into Group A and Group B)

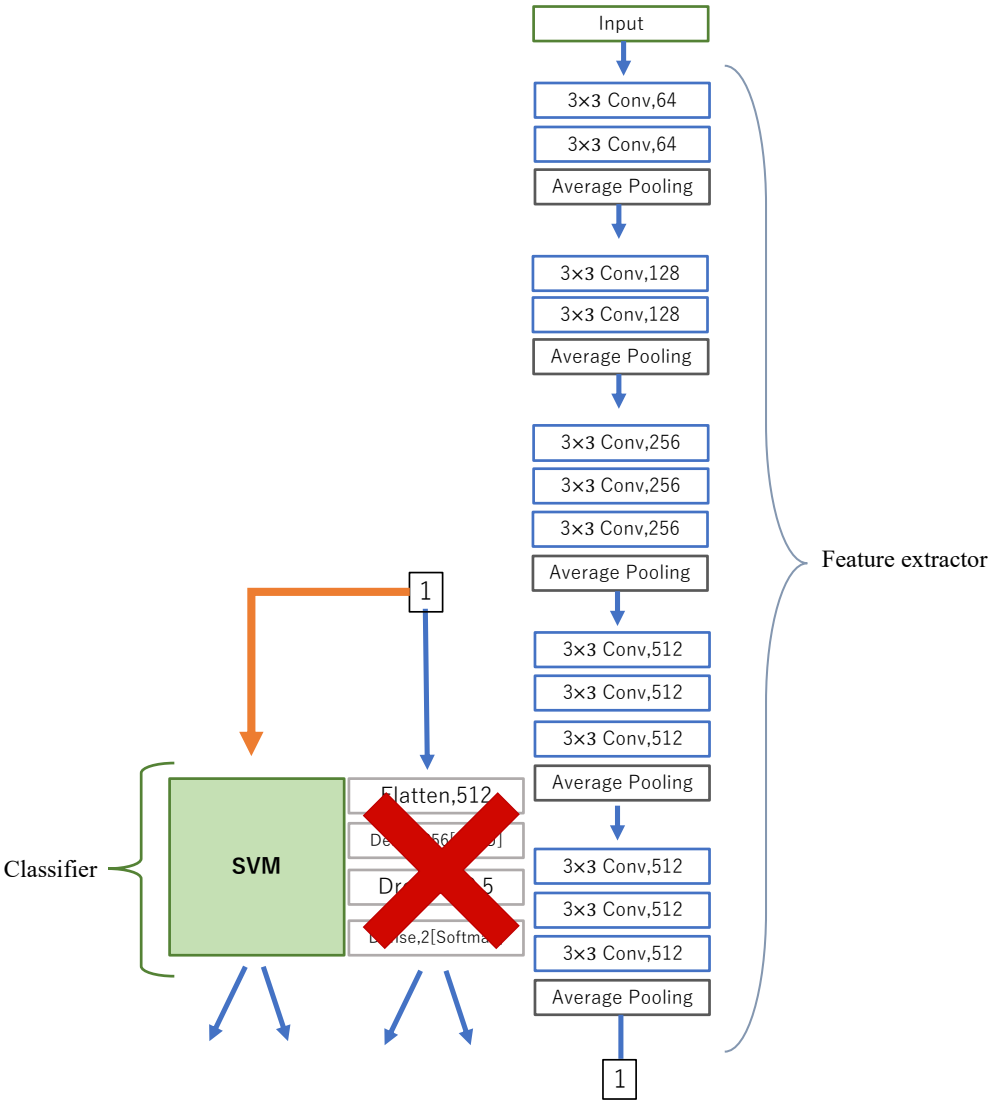

Fig. 54 VGG architecture with discriminator as support vector machine

4.2 Batch Normalization

Batch Normalization is a batch normalization for feature computation proposed by Ioffe Sergey [25] et al. In Deep Networks, hidden layers in deep layers have problems with internal covariate shifts, a phenomenon in which the input distribution changes for each layer and each activation. Covariate shift is a problem. Internal Covariate Shift is not often discussed in ordinary machine learning algorithms in the context of a simple training dataset. However, sometimes the distribution of the training data sampling and test data input is so skewed that the algorithm cannot cope. In the VGG models of 2.2.1 and 2.2.2, only the initial inputs are normalized for each feature, so that in the process of propagating layer by layer, the distribution of inputs to each layer is different for each mini-batch. The distribution of the

inputs to each layer changes in the process of propagating layer by layer. Therefore, by adding batch normalization to the VGG model, the inputs are normalized not only for the initial feature inputs, but also for each layer. This sets the mean of the features to zero and the standard deviation to one for each layer, thereby eliminating the effects of distributional bias and scale. This normalization reduces the need for regularization methods such as Dropout and thus speeds up the learning process. since it has already been added to Resnet 50 in 2.2.3, we have added Batch Normalization for the VGG model. We also add Activation after the Batch Normalization layer. The discriminator was changed to the Global average pooling layer, which is computationally less expensive. Fig.55 and Fig.56 show the architecture of the new VGG16 (VGG16 + BN) and VGG19 (VGG19 + BN) models with the added layer.

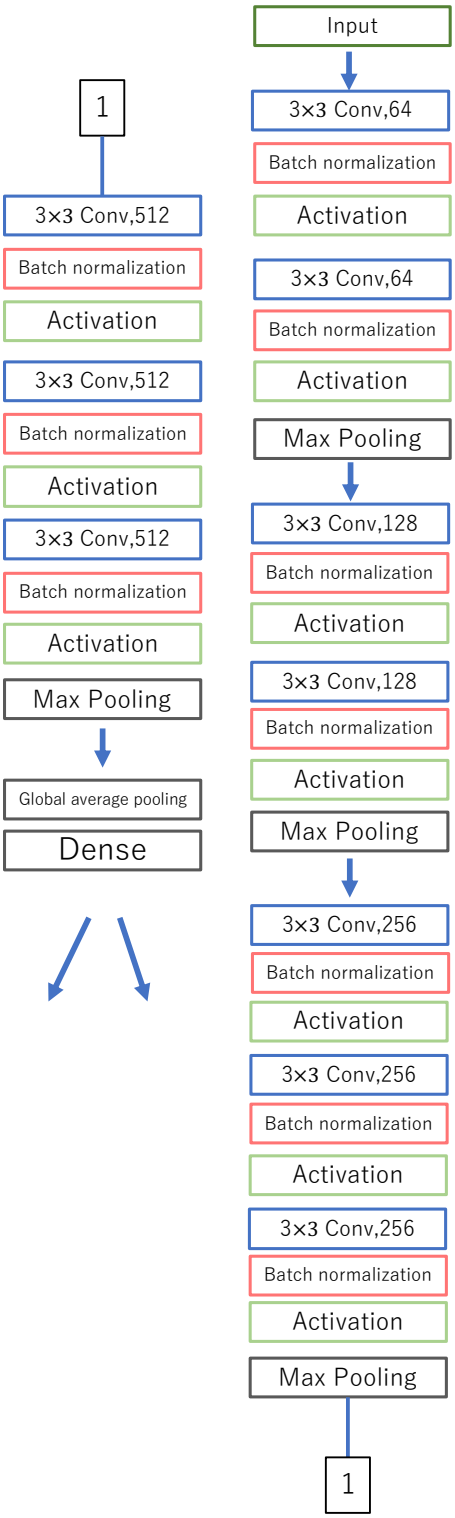

Fig. 55 VGG16 with batch normalization

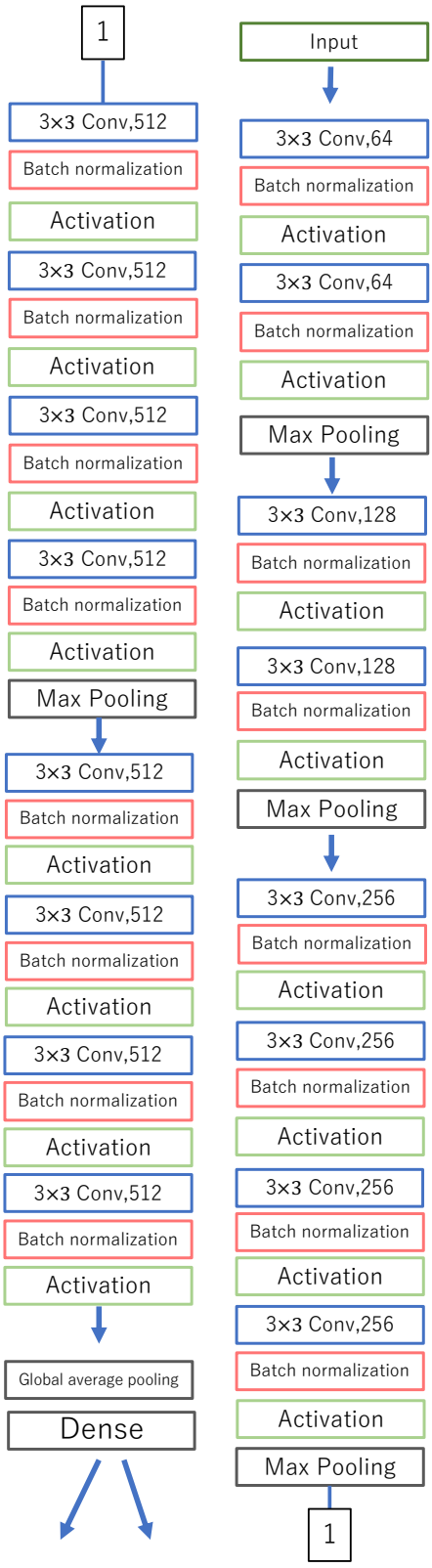

Fig. 56 VGG19 with Batch Normalization

5 Results in the proposed method

5.1 Results on SVM method

We created VGG16+SVM, which is a part of VGG16 converted to SVM, and VGG19+SVM, which is a part of VGG19 converted to SVM. We calculated the optimal hyperparameter values for these architectures and determined the hyperparameters to be C:1000 and gamma: 0.001 for the RBF kernel, respectively. data.

5.1.1 Evaluation on test data

Tab.6 shows the results of the VGG16 + SVM test data, which show an overall decrease in evaluation compared to the results with VGG16 in 3.1.3. In addition, there was no improvement in accuracy by data augmentation for VGG16 + SVM.

Tab.7 shows the results of the VGG19+SVM test data. as with VGG19+SVM, the overall evaluation was lower than that of VGG19 in 3.1.3. data augmentation improved the evaluation for A set and B set, but not for subsequent sets as the number of data increased. However, there was no improvement with the increase in the number of data after that. Based on these results, we can conclude that it is difficult to improve the performance of VGG by SVM in this study. Therefore, we believe that the all-join layer, which allows dimensionality reduction, shows better discriminative performance.

Tab. 6 Test result with VGG16 + SVM

		Accuracy	Precision	Recall	F1-score	auc
VGG16 + SVM	Aset	0.792	0.842	0.696	0.762	0.788
	Bset	0.786	0.774	0.786	0.773	0.786
	Cset	0.765	0.766	0.770	0.768	0.764

Tab. 7 Test result with VGG19 + SVM

		Accuracy	Precision	Recall	F1-score	auc
VGG19 + SVM	Aset	0.708	0.631	0.631	0.631	0.695
	Bset	0.750	0.774	0.727	0.750	0.750
	Cset	0.723	0.721	0.730	0.726	0.723

5.1.2 Grad-CAM

The Grad-CAM results of VGG16+SVM for each dataset are shown in Fig.57, Fig.58 and Fig.59. It can be seen that the overall features cannot be captured but are captured in a wide range. Therefore, it can be confirmed that the results depend on the background. The number of captured features does not change as the number of image data increases.

Fig.60, Fig.61 and Fig.62 show the Grad-CAM results of VGG16+SVM for each dataset, indicating that VGG16+SVM captures more features around the trigeminal nerve than VGG16+SVM. Compared with VGG19, VGG19+SVM tends to capture features in the same way. However, the number of features captured does not change as the number of image data increases. From these results, it can be concluded that the addition of SVM does not improve performance.

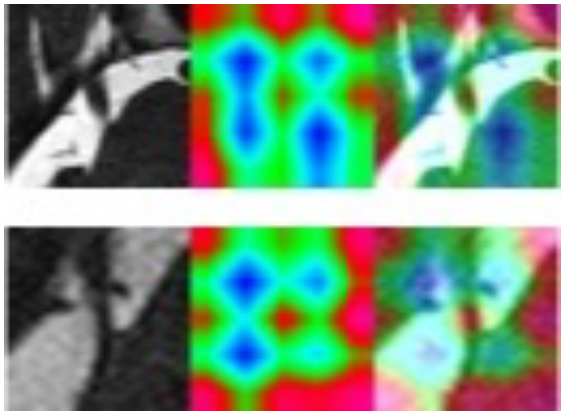

Fig. 57 Grad CAM of VGG16+SVM on A set

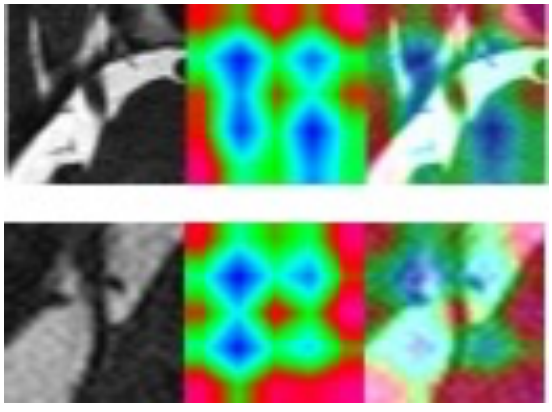

Fig. 58 Grad CAM of VGG16+SVM on B set

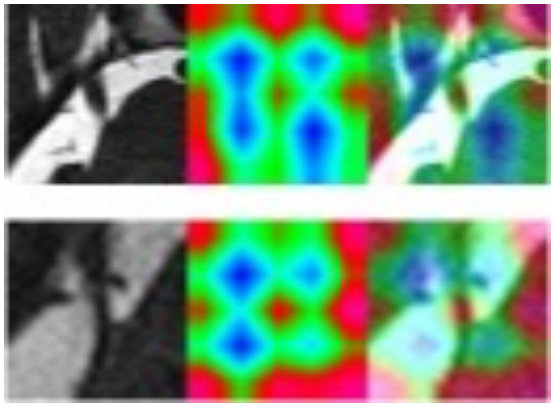

Fig. 59 Grad CAM of VGG16+SVM on C set

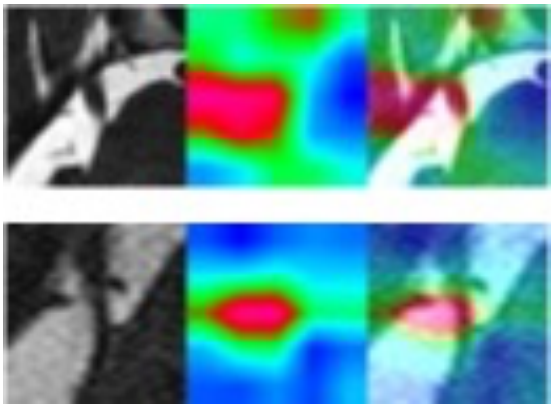

Fig. 60 Grad CAM of VGG19+SVM on A set

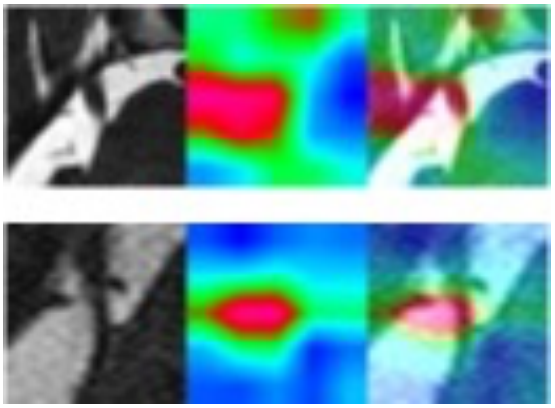

Fig. 61 Grad CAM of VGG19+SVM on B set

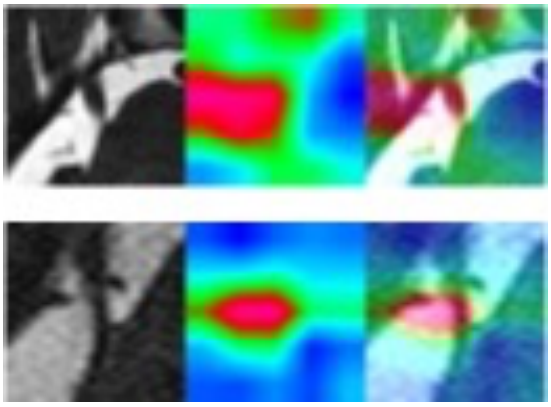

Fig. 62 Grad CAM of VGG19+SVM on C set

5.2 Results after adding Batch Normalization

Evaluate the performance of VGG16 + BN with Batch Normalization added to VGG16 and VGG19 + BN with Batch Normalization added to VGG16, respectively, from test data results for A set, B set, and C set data sets. 2. 6 and compare the results with those of the VGG model in Chapter 3.

5.2.1 Training process

Fig.63, Fig.64, and Fig.65 show the learning process for each dataset of VGG16 + BN. Overall, the learning accuracy of VGG16 + BN after adding Batch Normalization is higher than that of VGG16 + BN on the Validation data. Fig. 65 also shows that there is a significant increase in accuracy in the C set, and there are further indications of this after epoch 180. The same behavior was observed in the training data. However, the training data showed faster learning convergence.

The learning process for each dataset of VGG19 + BN is then shown in Fig. 66, Fig. 67, and Fig. 68. Fig. 66 shows that the learning process for the VGG19 + BN dataset is not convergent, although the learning accuracy improves locally. Fig.67 shows that the B set shows a significant improvement in learning accuracy on the validation data compared to Fig.18. Fig.68 shows that the C set shows a similar improvement to the B set compared to Fig.18, with faster learning convergence, as in the comparison of the B set. Compared to the B set, the C set did not improve the learning accuracy, but it did reduce the variability of the accuracy. Overall, the accuracy of the Validation data was improved compared to the VGG19 data, although the accuracy variability during the learning process was increased. The increase in accuracy variability during the learning process can be attributed to the decrease in learning stability due to the elimination of Dropout. However, after the addition of batch normalization, the accuracy of both VGG16 and VGG19 improved.

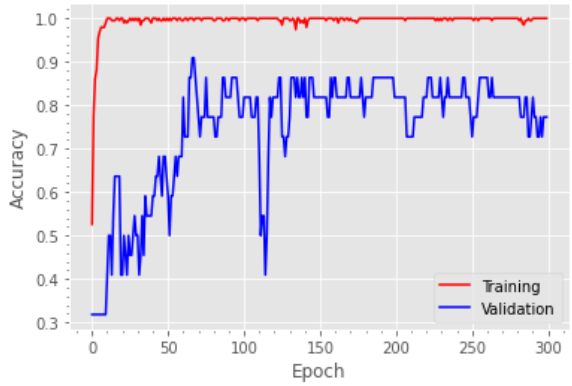

Fig. 63 Learning process with A set

(VGG16 + BN)

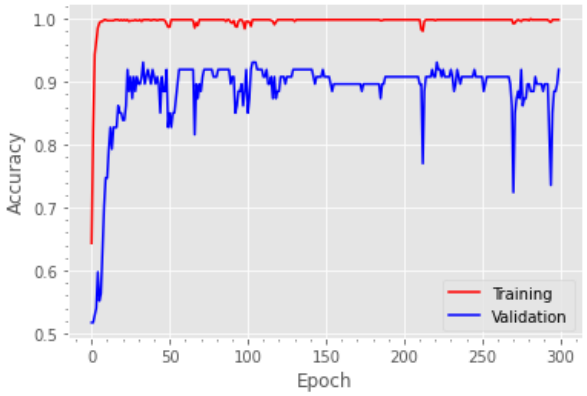

Fig. 64 Learning process with B set

(VGG16 + BN)

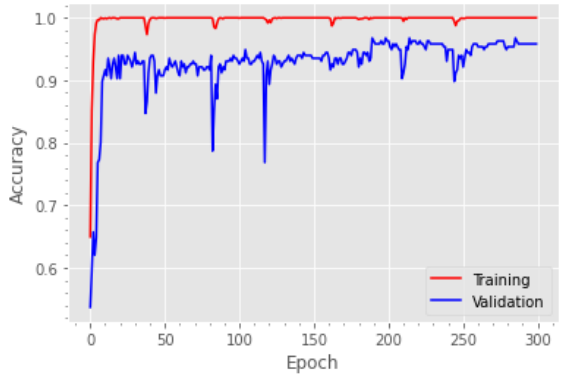

Fig. 65 Learning process with C set

(VGG16 + BN)

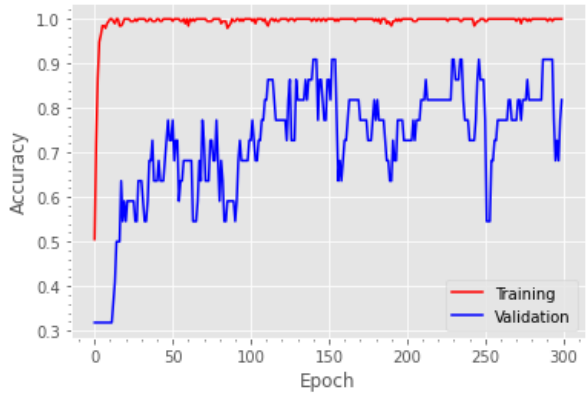

Fig. 66 Learning process with A set

(VGG19 + BN)

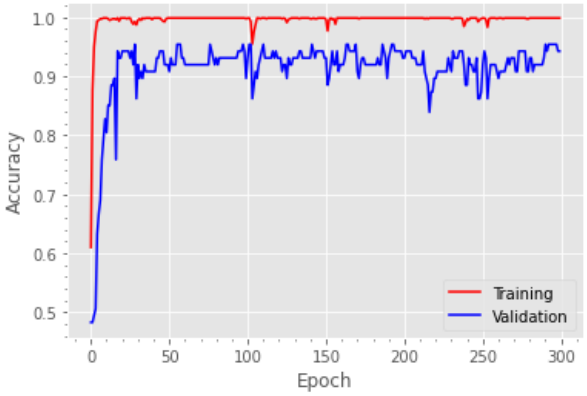

Fig. 67 Learning process with B set

(VGG19 + BN)

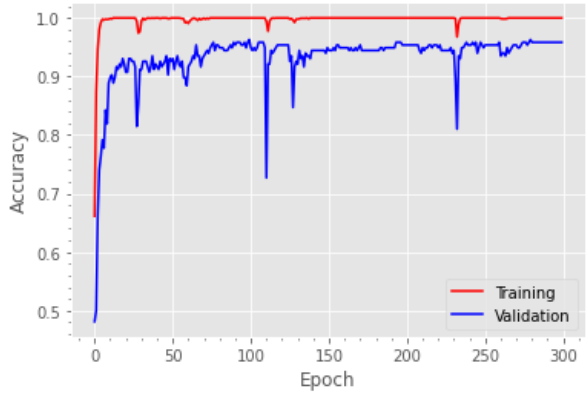

Fig. 68 Learning process with C set

(VGG19 + BN)

5.2.2 Loss function

Fig.69, Fig.70, and Fig.71 show the learning process of the loss function for each dataset of VGG16 + SVM. According to Fig. 70, the B set has the same variation in loss values as the A set, and it is locally large. Fig. 70 shows that the B set, like the A set, has a variation of loss values that is locally large, and in particular, the behavior of loss values after epoch 250 is significant, and there is an increasing trend in loss values. After epoch 180, some loss values are less than 0.2. Compared to Fig. 25, the loss values are generally small and decreasing, although the variation of the loss values is large.

Fig.72, Fig.73 and Fig.74 show the learning process of the loss function for each dataset of VGG19 + SVM. In the B set, Fig. 73 shows that the loss values in the A set are less scattered and the loss values are smaller than those in the same B set of VGG16 + SVM. In the C set, according to Fig. 74, the loss value is less scattered than in the B set, but there are some areas where the loss value is more pronounced. Compared to Fig. 28, the loss values are lower, and there is a decreasing trend from an increasing trend to a decreasing trend. A similar trend was observed in the accuracy of the Validation data for the learning process in 5.2.2. The VGG + BN model shows less learning stability than the VGG model, but the loss values tend to be smaller, and the learning performance improves.

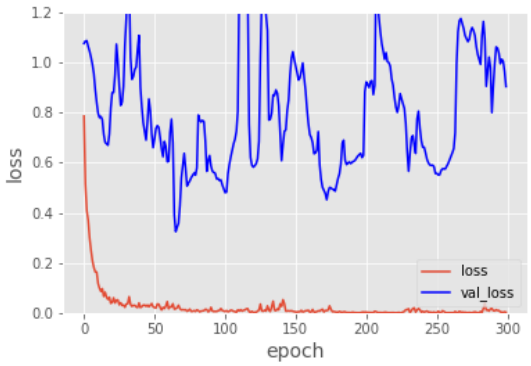

Fig. 69 Loss process with A set (VGG16 + BN)

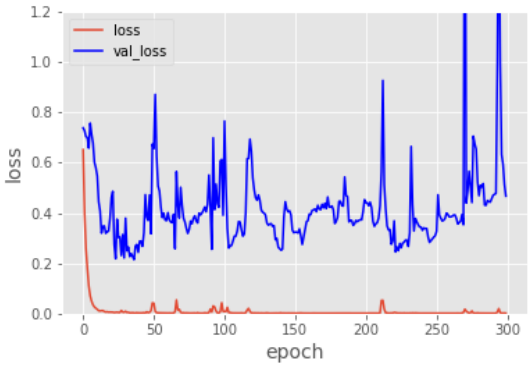

Fig. 70 Loss process with B set (VGG16 + BN)

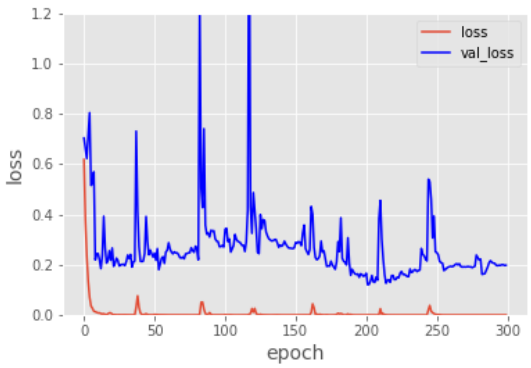

Fig. 71 Loss process with C set (VGG16 + BN)

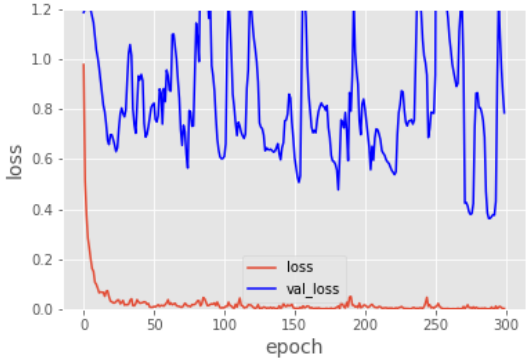

Fig. 72 Loss process with A set (VGG19 + BN)

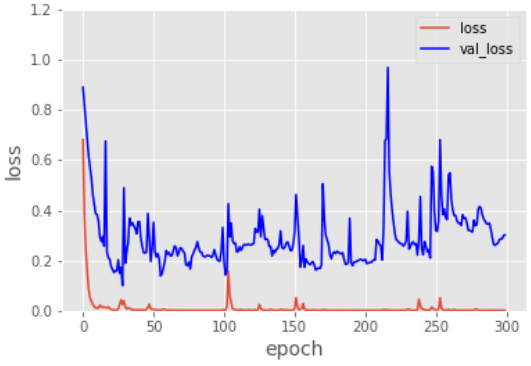

Fig. 73 Loss process with B set (VGG19 + BN)

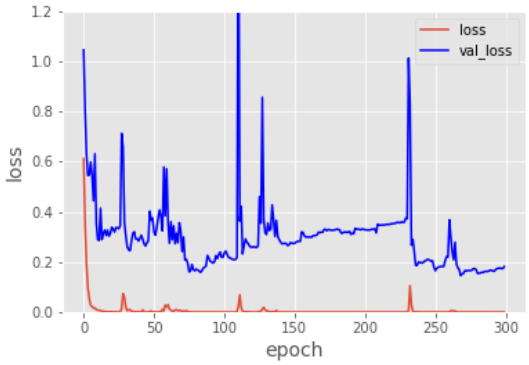

Fig. 74 Loss process with C set (VGG19 + BN)

5.2.3 Evaluation on test data

Tab.8 shows the results (accuracy, precision, recall, F1-score, auc) of each test data in VGG16 + SVM. B set showed a significant change from VGG16. B set improved accuracy 0.052, precision 0.056, recall 0.044, and F1-score 0.052 compared to VGG16. C set improved accuracy 0.012, recall 0.012, and F1-score 0.052 compared to VGG16. A set showed a significant change from VGG16. B set showed a significant change from VGG16. C set showed a significant change from VGG16. B set showed a significant change from VGG16. The C set improved accuracy 0.012, recall 0.047, F1-score 0.013, and auc 0.013 compared to VGG16. Overall, the evaluation increased after the addition of batch normalization. Among them, VGG16 + SVM in the C set showed the highest values in all the evaluations, indicating that the VGG16 model performed the best. The confusion matrix of VGG16 + SVM is shown in Fig. 75. 11 out of 240 test images are incorrectly predicted. Fig. 60 shows that out of the 11 images, 7 are misclassified as false positives and 4 are misclassified as false negatives. The results of the confusion matrix also show that the addition of batch normalization improves the discrimination performance.

Tab.9 shows the results for each test set in VGG19 + SVM. However, the C set achieved a significant improvement in accuracy 0.079, precision 0.090, recall 0.062, F1-score 0.073, and auc 0.027 after the addition of Batch Normalization. The VGG19 + SVM also showed the highest overall evaluation, indicating the effectiveness of data augmentation, as the

evaluation increased as the number of images increased. Fig. 61 shows that out of 240 test images, 5 images are mispredicted, and Fig. 76 shows that out of 11 images, 2 are misclassified as false positives and 3 as false negatives. The results of the confusion matrix also show that the addition of batch normalization improves discrimination performance. Overall, these results indicate that the addition of batch normalization improves the evaluation performance of the VGG model and demonstrates the effectiveness of batch normalization. Among them, VGG19 + SVM in the C set showed high values for all evaluations, and the results of the confusion matrix also showed low misclassification. Therefore, we can conclude that VGG19 + SVM is the model with the best discriminative performance in this study. As with the VGG results in 3.1.3, the effectiveness of data augmentation is confirmed.

Tab. 8 Test result with VGG16 + BN

		Accuracy	Precision	Recall	F1-score	auc
VGG16 + BN	A set	0.792	0.800	0.727	0.762	0.944
	B set	0.917	0.880	0.957	0.917	0.963
	C set	0.954	0.944	0.967	0.955	0.995

Tab. 9 Test result with VGG19 + BN

		Accuracy	Precision	Recall	F1-score	auc
VGG19 + BN	Aset	0.792	0.714	0.909	0.800	0.888
	Bset	0.896	0.875	0.913	0.894	0.951
	Cset	0.979	0.983	0.975	0.979	0.991

Fig. 75 Confusion Matrix of VGG16 + SVM on C set

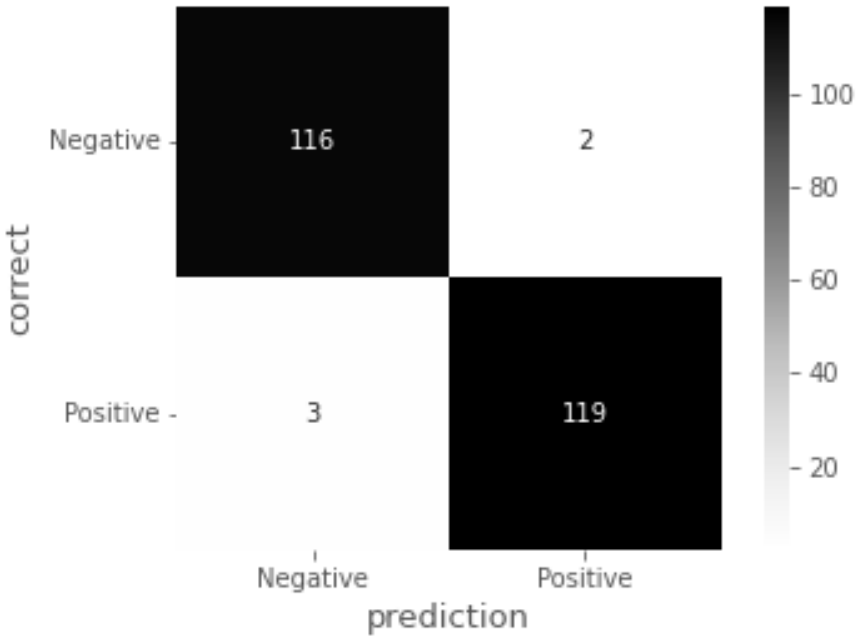

Fig. 76 Confusion Matrix of VGG19 + SVM on C set

5.2.4 ROC curve

The ROC curves for each test data of VGG16 + SVM and VGG19 + SVM are shown. First, the ROC curves for the VGG16 + SVM test data are shown in Fig. 77, Fig. 78, and Fig. 79. In Fig. 79, the curve is almost a leftward curve, and the range of misclassification is narrow and auc is 0.995, indicating that the discriminative performance is high.

Fig.80, Fig.81, and Fig.82 show the ROC curves of the VGG19 + SVM test data, indicating that the VGG19 + SVM discriminates the VGG19 + SVM data more clearly than the VGG19 test data in 3.1.4. In Fig. 82, the curve tends to be more to the left, and auc is as high as 0.99, indicating a significant improvement in performance. Compared to VGG16 + SVM in the same C set, VGG19 + SVM has a wider range of false positives, but fewer false positives. Overall, there was an increase in the ROC curve due to batch normalization. The C set of each model showed high auc and narrowed the range of false positives, indicating clear discriminative performance.

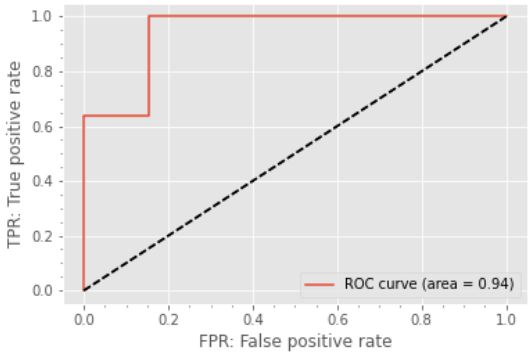

Fig. 77 ROC curve with A set (VGG16 + BN)

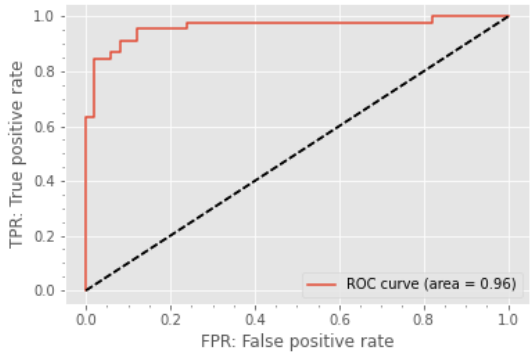

Fig. 78 ROC curve with B set (VGG16 + BN)

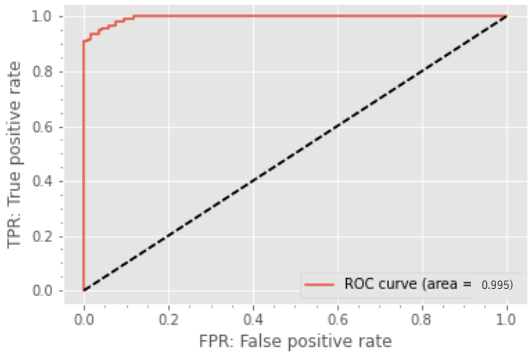

Fig. 79 ROC curve with C set (VGG16 + BN)

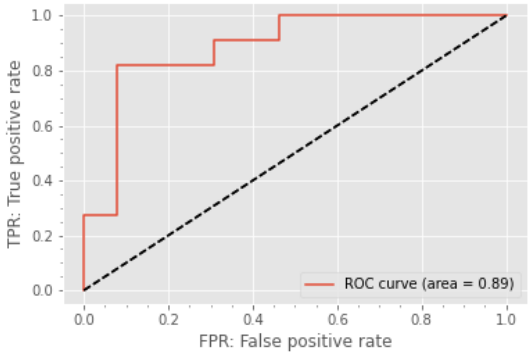

Fig. 80 ROC curve with A set (VGG19 + BN)

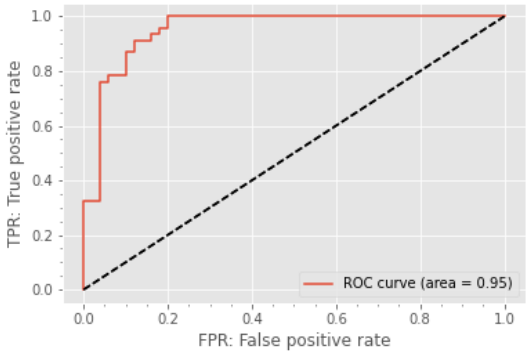

Fig. 81 ROC curve with B set (VGG19 + BN)

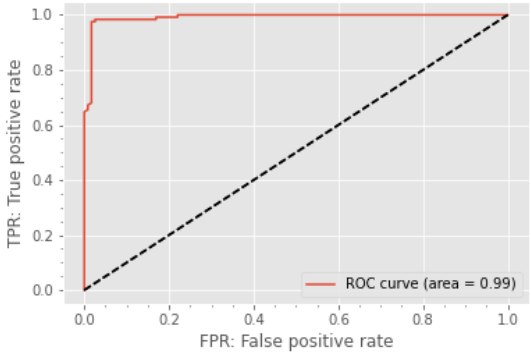

Fig. 82 ROC curve with C set (VGG19 + BN)

5.2.5 Grad-CAM

The VGG16+BN Grad-CAM results for each dataset are shown in Fig. 83, Fig. 84, and Fig. 85. It can be seen that the negative images tend to capture the trigeminal nerve. In the positive image, the trigeminal nerve was not well captured in Fig. 83, but it was captured in Fig. 85. Compared to VGG16, VGG19 and VGG16 are able to capture more features from the images.

The Grad-CAM results for VGG19+BN for each dataset are shown in Fig. 86, Fig. 87, and Fig. 88. In the negative image, the features were not captured in Fig. 87, but they were captured again in Fig. 88. In the positive image, Fig. 88 shows that the trigeminal nerve is captured to a greater extent than in Figs. 86 and 87. Therefore, it can be said that the discrimination performance was improved by the increase in the number of image data. In Fig.88, both the negative and positive images show that the trigeminal nerve is captured in a larger size than in Fig.86 and Fig.87.

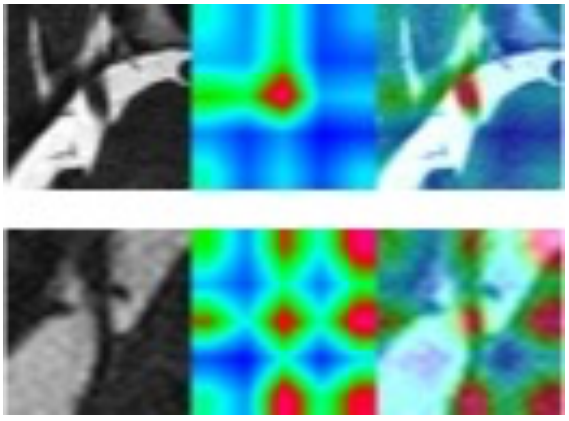

Fig. 83 Grad-CAM of VGG16 + BN on A set

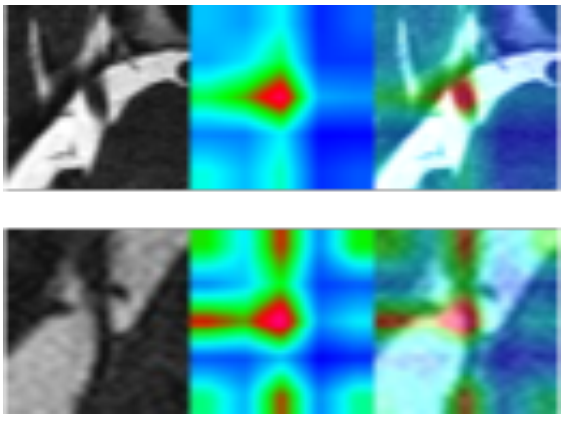

Fig. 84 Grad-CAM of VGG16 + BN on B set

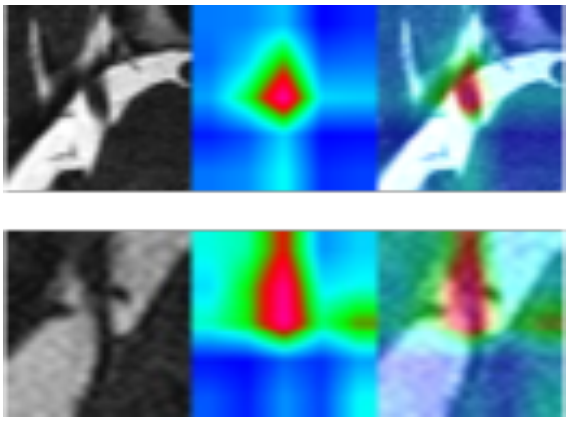

Fig. 85 Grad-CAM of VGG16 + BN on C set

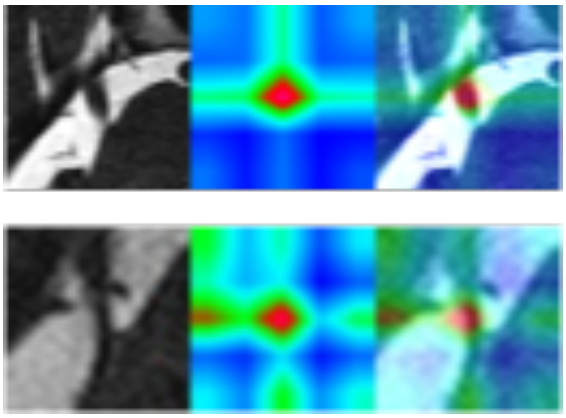

Fig. 86 Grad-CAM of VGG19 + BN on A set

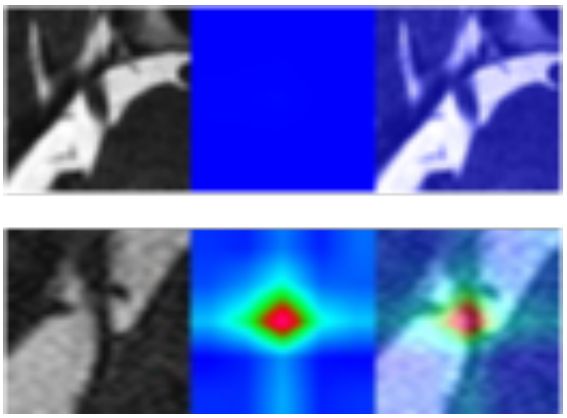

Fig. 87 Grad-CAM of VGG19 + BN on B set

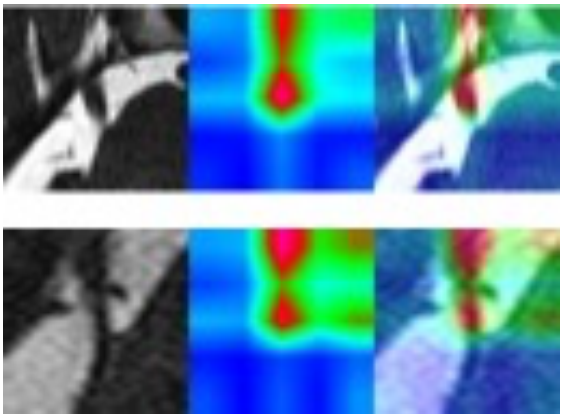

Fig. 88 Grad-CAM of VGG19 + BN on C set

6 Conclusion

In this paper, we examine and propose an AI diagnostic method for image analysis of trigeminal neuralgia using deep learning. In doing so, a series of experiments were conducted with three different classification models, VGG16, VGG19, and ResNet50, which are mainly CNNs, and the performance of each model was compared. The results showed that the best performing model achieved accuracy: 0.942, precision: 0.958, recall: 0.926, F1-score: 0.942, and auc: 0.982 when trained on 80 x 80 images using VGG16 in the C set. CAM also confirms the performance of the VGG model. To further improve the performance of the VGG model, we improved the structure of the VGG model and proposed two methods. The support vector machine did not improve performance, but the addition of batch normalization did, VGG19 achieved accuracy of 0.979, precision: 0.983, recall: 0.975, F1-score: 0.979, and auc: 0.991 in the C set, and the best model in terms of fewer misclassifications and Grad-CAM The model also performed the best in terms of low misclassification and Grad-CAM results. Based on this high classification accuracy, we believe that this study will play a role in the development of an AI diagnostic system for imaging diagnosis of trigeminal neuralgia, which may be useful in the near future in regions where there are medical disparities.

Acknowledgements

We would like to thank Associate Professor Mitsugu TODO, Department of Mechanical Systems Engineering, Interdisciplinary Graduate School of Engineering Science, Kyushu University, and Professor Hechang LIM, Department of Mechanical Engineering, Pusan National University, for their valuable guidance and cooperation in conducting this study. We also thank Assistant Professor Daitake UMEBAYASHI, Graduate School of Medicine, Kyoto Prefectural University of Medicine, for providing many materials. Finally, we would like to thank the members of the Bioenergy Engineering Laboratory and FEEL Laboratory for their valuable advice and cooperation in carrying out this study. The authors would like to express their gratitude.

Reference

[1] 日本神経治療学会ガイドライン統括委員会 B,典型的三叉神経痛 I 臨床像と病態、診断, P769~770,神経治療 Vol.38 No.5(2021)
[2] 日本神経治療学会ガイドライン統括委員会 A,三叉神経痛も分類と診断基準, P767~768,神経治療 Vol.38 No.5(2021)
[3] 日本頭痛学会・国際頭痛分類普及委員会　訳：国際頭痛分類第 2 版　新訂増補日本語版. 医学書院　2007
[4] Katusic S, Beard CM, Bergstralh E et al : Incidence and clinical features of trigeminal neuralgia , Rochester, Minesota, 1945-1984, Ann27:89-95, 1991
[5] MacDonald BK, Cockerell OC, Sander JW et al : The incidence and lifetime prevalence of neurological disorders in a prospective community-based study in the UK, Brain 123 : 665-676, 2000
[6] Muller D, Obermann M, Yoon MS et al : Prevalence of trigeminal neuralgia and persistent idiopathic facial pain : A population-based study, Cephalalgia 32: 1542-1548, 2011
[7] Crucci G, Biasiotta A, Galeotti F et al : Diagnostic accuracy of trigeminal reflex testing in trigeminal neuralgia ,Neurology 66: 139-141, 2006
[8] Dandy WE: Concerning the cause of trigeminal neuralgia. Am J Surg 24: 447-455, 1934
[9] Gardner WJ: Concerning the mechanism of trigeminal neuralgia and hemifacial spasm. J Neurosurg 19: 947-958, 1962
[10] Janaetta PJ : Arterial compression of the trigeminal nerve at the pons in patient with trigeminal neuralgia. JNeurosurg 26: 1159-1162,1967
[11] Baker FG, Janetta PJ, Bissonette DJ et al : The long-term outcome of microvascular decompression for trigeminal neuralgia N engl J med 334 : 1077-1083, 1996
[12] 荒木信夫, 厚東篤夫, 三叉神経痛. 神経内科 29： 126-136, 1988
[13] 日本神経治療学会ガイドライン統括委員会 B,典型的三叉神経痛 IV ブロック療法, P769~770,神経治療 Vol.38 No.5(2021)
[14] Maarbjerg S, Di Stefano G, Bendtsen L et al: Trigeminal neuralgia – diagnosis and treatment, Cephalalgia 37 : 648-657, 2017
[15] 田中博, 医学における人工知能―その動向と最近の進歩. BME, 7 (5), 1-16, 1993.
[16] 山根友絵, et al. 日本における医療分野での人工知能（AI）活用に関する文献検討. *Bulletin of Toyohashi Sozo University*, 25: 61-70, 2021.
[17] 井上悠輔, et al."医療における AI 関連技術の利活用に伴う倫理的・法的・社会的課題の研究." 医療情報学 41.2 56-57, 2021.
[18] Shorten, Connor, and Taghi M. Khoshgoftaar. "A survey on image data augmentation for deep learning." Journal of big data 6.1 (2019): 1-48.
[19] Y. LeCun, L. Bottou, Y,Bengio, and Haffner : Gradiant-based learning applied to document recognition, Proc. IEEE, vol86, no.11, pp.2278-2324, 1998.
[20] Yamashita, Rikiya, et al. "Convolutional neural networks: an overview and application in radiology." Insights into imaging 9.4 (2018): 611-629.
[21] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

[22] He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
[23] Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." *Proceedings of the IEEE international conference on computer vision*. 2017.
[24] Boser, Bernhard E., Isabelle M. Guyon, and Vladimir N. Vapnik. "A training algorithm for optimal margin classifiers." Proceedings of the fifth annual workshop on Computational learning theory. 1992.
[25] Ioffe, Sergey, and Christian Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift." International conference on machine learning. PMLR, 2015.