参赛队员姓名：龚唐跃

中学：北京市十一学校

省份：北京

国家/地区：中国

指导教师姓名：窦向梅

指导教师单位：北京市十一学校

论文题目：Smarter Radiologists: A User-Friendly Algorithm Aiming to Improve the Accuracy and Readability of Intracranial Hemorrhage CT Detection

Smarter Radiologists: A User-Friendly Algorithm Aiming to Improve the Accuracy and Readability of Intracranial Hemorrhage CT Detection

Tangyue Gong
Beijing National Day School
e-mail: gongtangyue@163.com

September 15, 2022

Abstract

Intracranial hemorrhage (ICH) is a serious medical emergency in which blood accumulates within the skull and can be induced by a series of fatal diseases such as trauma, stroke, aneurysm, vascular malformations, hypertension, and coagulation disorders. 50% of patients with ICH die immediately, while others may survive if they receive timely treatments. Therefore, rapid and accurate identification and classification of ICH is essential for applying the correct treatment plan. However, the diagnosis and classification of ICH is a time-consuming task that relies heavily on the experience of radiologists. To relieve the burdens of radiologists and avoid potential human errors, in this study, we designed and proposed a deep learning model that can identify and classify ICH from CT images. In addition, our model is able to demonstrate the precise location in the brain where ICH occurs on CT images. Specifically, our method utilized two image preprocessing methods with an optimized ensemble model that achieved superior performance compared with six baseline models and other ensemble models. Furthermore, we designed a user-friendly app that allowed radiologists to easily access our model and analyze their clinical CT image data in real-time.

Keywords: Intracranial hemorrhage, deep learning, CT images

Contents

1 Introduction

Intracranial hemorrhage (ICH) is bleeding inside the skull. ICH can cause an increase in intracranial pressure, which will crush delicate brain tissue or restrict its blood supply. Because of this, ICH can lead to a number of serious health problems such as trauma, stroke, aneurysm, vascular malformations, hypertension, and coagulation disorders. Statistically, 50% of people die immediately after an ICH occurs, while others can recover if they are properly treated within 24 hours[1]. Therefore, the detection of ICH is truly a matter of life. The faster it is detected, the better the patient's chances of recovery.

Traditionally, to identify ICH, a radiologist should take CT images of patients. The radiologist will then examine the CT scan and diagnose the existence of the ICH. After that, radiologists have to classify the ICH into 5 different subtypes and treat them differently. However, because of the nuances of each subtype and the hundreds of CT images of a single patient, determining ICH is a very difficult task. Thus, the accuracy of the detection depends entirely on the skill of a radiologist with subspecialty training, as there are too many subtypes and too many images to classify. Because of all these challenges, even sophisticated radiologists might make misdiagnoses, especially when they are overloaded. Therefore, there is a need for a rapid and accurate method of diagnosing brain emergencies in terms of detecting and classifying ICH. Recently, computers, with machine learning algorithms have been a good choice for this task. With the development of information technology, more and more computer-aided diagnoses and products have come into being. They are widely used in many industries, especially in the medical field, to predict the structure of proteins[2], genomics[3], identify useful targets for the treatment of cancer[4], and so on. Therefore, in this study, computers also play an important role in detecting ICH.

1.1 Machine Learning

To be general, computer-aided classification of ICH is mainly done in two ways. Compared to the classical way, machine learning (ML) is rather simple and universal. ML is an algorithm, which imitates human's ability to study from the environment. It is widely used in various fields, such as pattern recognition[5], computer vision spacecraft engineering[6], finance[7], and computational biology[8]. Different from traditional algorithms, ML is not solving problems with a finished program, but with a half-finished program, where the method itself will replenish itself by repetition and learning. The process of replenishing is called training[9]. Deep Learning (DL) is a subtype of machine learning. DL imitates not only the basic logic of the learning ability of human beings but also its structure. Many layers are used in the process of deep learning. They together formed a neural network. A neural network is composed of all kinds of layers, including Fully Connected Layers (FCL), Convolutional Layers (ConvL), and so on.

ConvLs are on the top of the model. They are composed of filter (kernel) layers and output feature maps. Each kernel is a small matrix. The number in every pixel of the kernel times with the respective matrix in the input layers. Add all the results of multiple together to form the output layer. If there is more than one input layer, every kernel will filter every input layer. Therefore, the amount of the outcome layer is equal to the multiplication of the number of kernels and the number of input layers. Kernels are responsible for capturing the characteristics of the figure. For example, some kernels are good at capturing the edge of each figure in a

picture, while others are good at determining colors, as shown in Figure 1 After the training process, every kernel can extract a specific kind of feature. FCLs are usually located after the convolutional layers. They, as suggested by their names, are fully connected with each other, which can iterate to the next layer by multiplying and adding the weight and bias respectively. After the calculation of the hidden layer, the output layer contains 6 neurons because there are 6 kinds of ICH. Therefore, we can finally build a tool to detect ICH by using the DL model.

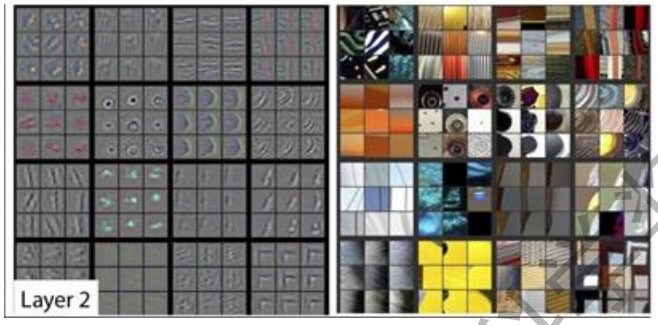

Figure 1: Different characters extracted by kernels. The right part of this figure shows the original image, while the left part of this figure shows the characteristics extracted by the kernels

Applying such a tool to the field of the detection of ICH has several benefits. On one hand, this tool can help beginners of neuroradiology, who have little experience, and neuroradiologists from developing countries to detect ICH as precisely as possible. On the other hand, for experienced neuroradiologists, applying such a tool can make detection faster. Although the model may help neuroradiologists a lot, the final decision of whether the ICH exists will be made by neuroradiologists, while the model will provide valuable suggestions. Though there are many benefits of using the model, many objectors to using DL to help radiologists to classify ICH still exist. One main reason is that the process of DL is a black box. It is hard for researchers to know how the program is optimized and why the program optimizes it that way[10]. In other words, radiologists can't find the direct relationship between the change and the detection of ICH. Such a black box makes DL less convincing and finally leads neuroradiologists to abandon DL.

Though this is an influential flaw for ML, computer scientists have found methods to solve it. In this paper, one of these methods, grad-cam, is applied, and the picture it generates can locate the precise location of ICH.

1.2 Our Contribution

In this paper, a model which can detect ICH and localize ICH was designed. The code of the model was published on GitHub so that the community can try the algorithm on their dataset and improve the model. An APP based on the model was also published. The models proposed in this paper can enhance the efficiency of radiologists, which will, thus, save more lives, and boost the quality of people's life.

2 Related Work

Simply increasing the number of layers will not lead to high accuracy. Instead, it will make the model difficult to train and prone to overfitting. In order to effectively utilize CNN and FCN and optimize their performance, it is crucial to learn different combinations of CNNs and FCNs. There are many effective DL structures, including three most basic ones: AlexNet, VGGNet, and ResNet.

2.1 AlexNet

AlexNet is the earliest network, designed in 2012 by Alex Krizhevesky and his colleagues. As shown in Figure 2, AlexNet contains 5 convolutional blocks. The first block is composed of a convolutional layer (ConvL) and a max-pooing layer (MXL). The ConvL has 96 different kernels with a size of 11*11, and the MXL is 3*3, with a stride of 3. Same as the first layer, the second block has ConcL followed by an MXL. The only difference is that the number of the second layer's kernel is 256 with a size of 5*5. The third, fourth, and fifth ConvL are all in size of 3*3 and have the number of 384, 384, and 256 kernels respectively[11]. Although the model of AlexNet is groundbreaking, the number of hyperparameters is flexible. One of the studies aiming at radiation source-target recognition slightly changed the size of ConvL and FCL, which significantly improved the training speed and classification performance[12].

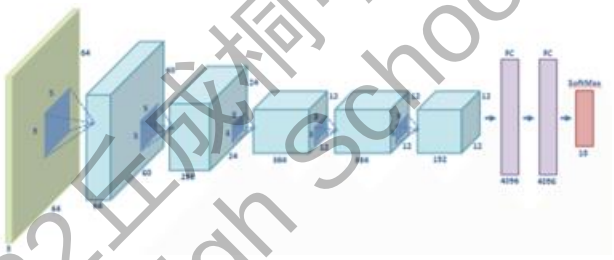

Figure 2: AlexNet has input of $64 \times 64 \times 3$ and output of $1 \times classes$.

2.2 VGGNet

Visual Geometry Group (VGG) was proposed by Simonyan and Zisserman at the University of Oxford. The inventors of VGGNet realized that the performance of a model is closely related to its depth. Guided by this concept, VGGNet was invented. Basically, VGGNet is an assembly of VGG blocks, in which two or more ConvLs are followed by an MXL. There is no specification of the size of kernels and MXL. In fact, VGGNets are classified based on their size and amount[11]. A group of researchers from Guilan University had applied VGGNet to identify abnormalities in cystoscopic images. They finally achieved an accuracy of 63%[13].

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Figure 3: Different types of VGGNet

2.3 Other Architectures

In the paper, a numbers of models are investigated such as Aggregating Nested Transformers[14], Big Transfer ResNetV2[15], and Pooling-based Vision Transformer[16]. The three structures mentioned above are the models that we believe will perform the best. Many structures were incorporated into the above models in order to improve the performance. They have made great progress, but most of them also have limitations. Hyunkwang Lee and other researchers used several convolutional neural networks (VGG16, ResNet-50, Inception-v3, Inception-ResNet-v2), a preprocessing pipeline, an atlas creation module, and a prediction-basis selection module. They finally achieved 98 percent sensitivity and 95 percent specificity. However, all of their data was collected from one institution. Another research was done by D.Venugopal. He proposed FEDL-ICH (FFE-DL for Intracranial Hemorrhage Detection and classification) and achieved 95.65 percent sensitivity, 97.93 percent of specificity, 96.43 percent of precision, and 96.56 percent of accuracy. These performance outreaches other same field models. Besides that, another research conducted by Daniel T.Huff and his team has also proven the effectiveness of DL in the medical field. Daniel T.Huff and his team have summarized both the technical and practical implementation details of model interpretability approaches for deep learning practitioners focusing on medical imaging applications. They have grouped interpretation approaches by their technical similarities, and by their relevance to different stages of the model development process. They have also provided practical advice for choosing between interpretation techniques and implementing them. The approach that this thesis proposed to interpret could be potentially valuable in medical imaging applications where abnormalities or pathologies have a hierarchical relationship. However, some researchers have argued that the commonly employed attribution-based interpretation methods provide unreliable and potentially misleading interpretations.

3 Material and Method

3.1 Datasets

Table 1: The distribution of images in each different type of ICH

Cohorts	number of images
Normal	121700
Epidural	3145
Intraparenchymal	16118
Intraventricular	16205
Subarachaniod	15671
Subdural	27161
Any	78300

The subject of this study comes from a competition held by Kaggle[17] and all data used in this research come from Kaggle's database[16]. The dataset is separated based on different subtypes. As shown in Figure 5, six different types of ICH are mainly classified. The intraparenchymal type is located inside the brain. It has a typical round shape and causes headaches, nausea, and vomiting. The intraventricular type locates inside the ventricle of the brain. It conforms to the ventricular shape and has the same symptom as the Intraparenchymal. The subarachnoid type locates between the arachnoid and pia mater. It tracks along the sulci and fissures and leads to the worst headache of life. The subdural type locates between Dura and arachnoid. It has the shape of a crescent and may be insidious. Finally, the epidural type subtype locates between the dura and the skull. It has the shape of a lentiform and leads to a skull fracture and altered mental status.

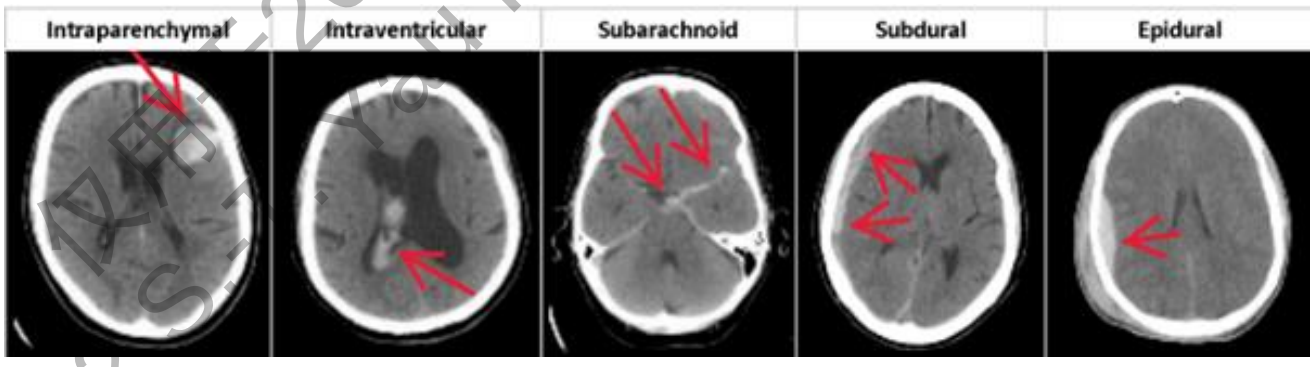

Figure 4: The image of different subtypes of ICH. The red arrows point at ICH that the CT image has ICH

As shown in Table 1, the dataset is rather imbalanced, with the epidural subtype being par-

ticularly small compared to the other subtypes. Such an unbalanced dataset may hinder the identification of the epidural subtypes by the model, ultimately leading to poor model performance and poor generalization ability.

3.2 Image Preprocess

Image preprocessing is used in many fields, including geography, medicine, etc. The purpose of image preprocessing is mainly to improve the efficiency of DL. In this study, image preprocessing can help the model identify ICH faster and more accurately.

3.2.1 Window Preprocess

Window preprocess is one of the most popular preprocess methods being used in DL. In this study, the data downloaded from the Kaggle[17] website is in the form of Dicom, ab international standard for medical images and related information. Values of every pixel denote the radiation density of that spot. Radiation density reflects the ability of electromagnetic radiation to pass through a certain kind of material. The larger the number, the more difficult it is for the radiation to pass through. Therefore, different kinds of tissue are represented by different ranges of values. For example, brain tissue ranges from 0 to 80, while subdural tissue ranges from -120 to 280. By setting the values, which are lower than the lower boundary or higher than the upper boundary of certain tissue, to the lower boundary and upper boundary respectively, other information in the CT scans besides the wanted tissue is dropped. This may help DL by allowing the model to get rid of the influence of the tissues that do not determine whether ICH is present or not, thus improving the accuracy of DL model. In this study, windows for brain tissue with a center of 40 and a width of 80, subdural tissue with a center of 80 and a width of 400, and soft tissue with a center of 30 and a width of 30. At the end of the image preprocessing, windowed images are normalized into ranges 0 to 1 in order to fit the model.

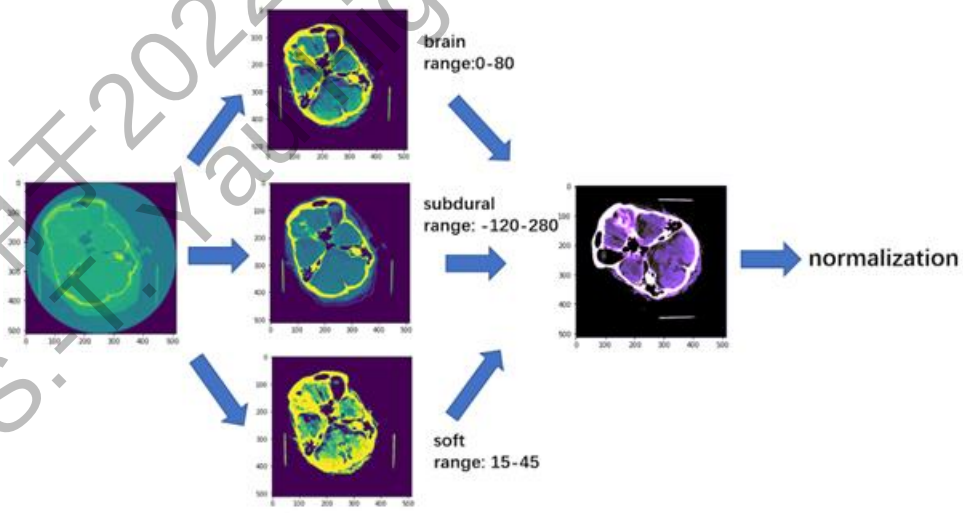

Figure 5: Steps of preprocessing

3.2.2 Adjacent Slice Image Preprocess

Adjacent slice image preprocess (ASIP) is another popular way of image preprocessing. It is mainly used in the medical field. Instead of three windows, ASIP uses only one window, with a range of 40-80. Since the Kaggle[17] data is in the form of Dicom, it is possible to track which patient and its adjacent slices each image belongs to. As shown in Figure 6, in ASIP, the core image, the image before the core image(pre_image), and the image after the core image(post_image) are clipped together into one RGB image. Then, it is normalized. ASIP works because human brains are 3D, while CT images are 2D. ASIP can increase the "thickness" of the CT image. If the pre_image and post_image have ICH, the core image has ICH for a great chance. Therefore, adding pre_image and post_image can help the model better determine whether the core image has ICH or not.

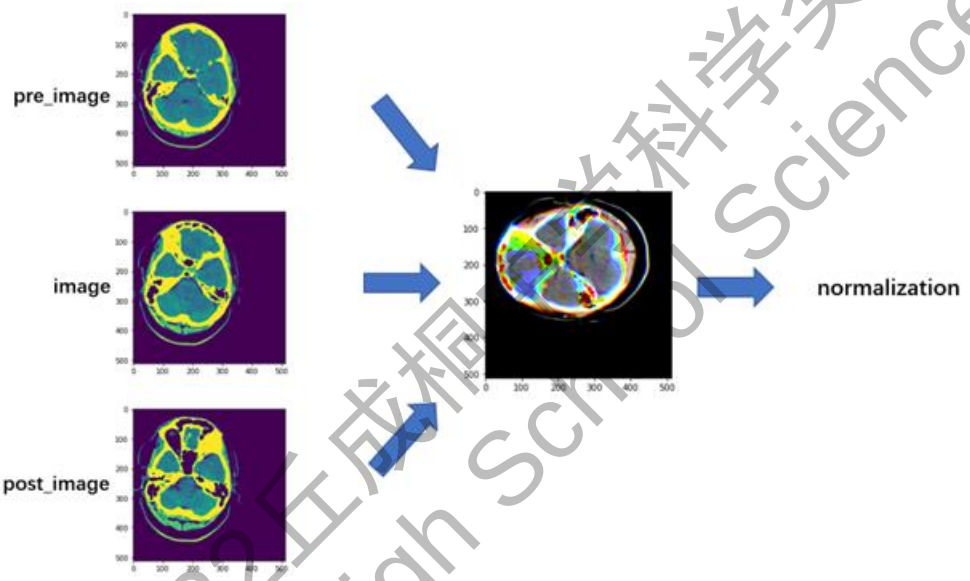

Figure 6: Adjacent slice image

3.2.3 Model Aechitecture

In this research, many architects are employed as baseline models, including Residual Networks (ResNet), EfficientNet, and Densely Connected Convolutional Networks (DenseNet). With the increase of the size of inputs, the width, depth, and resolution should all increase simultaneously since more characteristics should be caught by more kernels. In fact, there is a compound scaling method, scaling widtgetsdepth, and resolution with a fixed ratio. EfficientNet is a family of models applying the same basic concept. In this research, EfficientNetB2 is applied.

3.2.4 EfficientNet

With the increase of the size of inputs, the width, depth, and resolution should all increase simultaneously since more characteristics should be caught by more kernels. There is a compound

scaling method, scaling width, depth, and resolution with a fixed ratio. EfficientNet is a family of models applying the same basic concept. In this research, EfficientNetB2, which is a type of EfficientNet is applied[18].

3.2.5 ResNet

The depth of a neuron network is crucial and greatly determines the ability of the neuron network. The greater the depth of the network, the more it benefits from the depth. However, too deep may lead to the vanishing or exploding gradient. This problem is solved by normalization. As new layers are added, it becomes harder for layers at the bottom to propagate the information from the previous layers, which leads to the loss of the input information, and the degradation problem emerges. ResNet addresses this problem by constructing basic ResNet components, as shown in Figure 8. In the basic component, a passage through which the result of the first layer can directly affect the third layer is created. The third layer is equal to F(x)+x, instead of F(x). If F(x) is redundant, it can be zero. The ResNet model, as shown in Figure 9, is composed of those basic components. In this research, ResNet50, which has 50 layers, is applied[19].

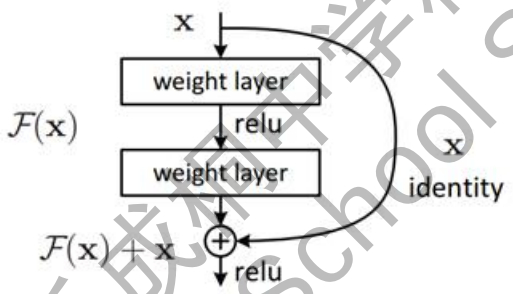

Figure 7: Basic component in the structure of ResNet model

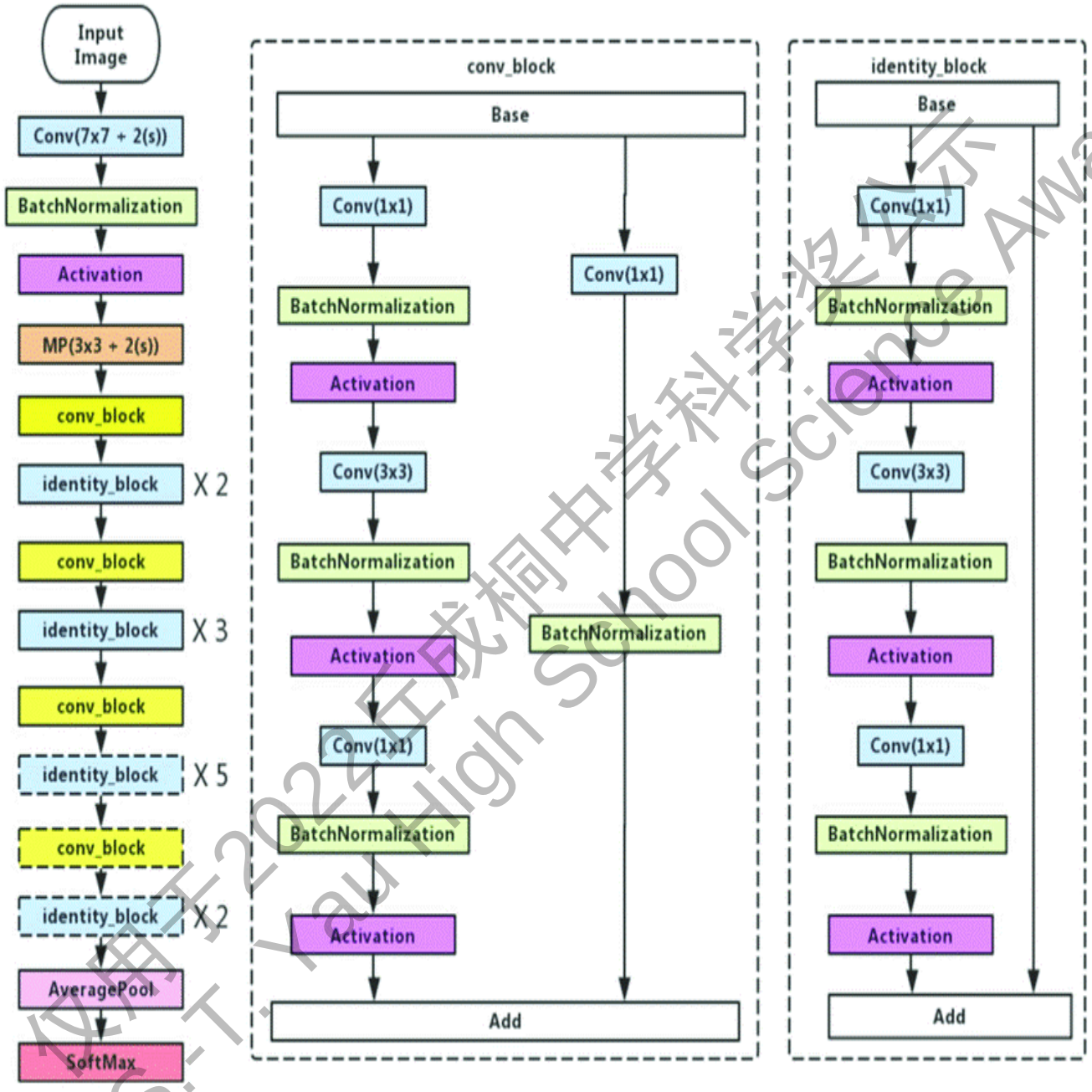

Figure 8: Structure of ResNet model. In the research, the input layer is the size of $512 \times 512 \times 3$, the output layer is 1×6.

3.2.6 DenseNet

Dense net changes the concept of traditional ConvL. The whole architecture is composed of many dense blocks, which are connected with each other just as normal layers. In each dense block, many layers are set and each layer is connected with the others as shown in Figure 9, which means that every layer has a direct impact on other layers in the same block. Such structure brings not only a more direct backpropagation process but also a decrease in the parameter and an increase in computational efficiency. In this research, Densenet121, which is a kind of DenseNet is employed[20].

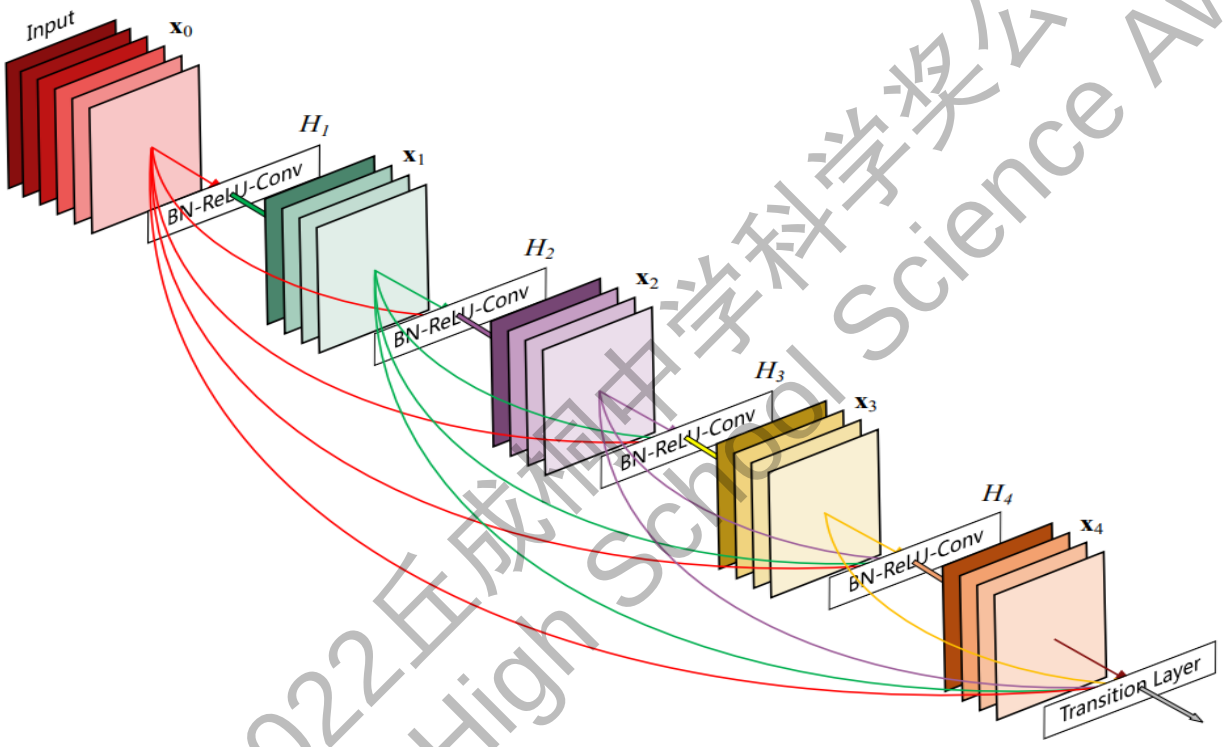

Figure 9: DenseNet model. In the research, the input layer is the size of $512 \times 512 \times 3$, the output layer is 1×6.

3.3 Ensemble

Model ensembles are models that are combined together by manipulating the predicted results of each model on a test dataset. There are many types of ensembles, such as Average ensemble, vote ensemble, and weighted ensemble. Ensemble is effective since models may make predictions based on different things. In addition, some models are especially accurate in predicting a certain kind of subtype. Therefore, ensembles can combine the advantages of all models and make a better output. This paper compares the ensembles of the windowing group, the adjacent group, and all of the models.

3.3.1 Average Ensemble

$$P_n = \frac{A_n + B_n + C_n}{3} \tag{1}$$

In equation 1, P_n is the n^{th} term of the prediction of the average ensemble. A, B, and C are the predictions of three different models. Average ensemble is to add the predictions of all the ensemble models and divide the result by the total number of models.

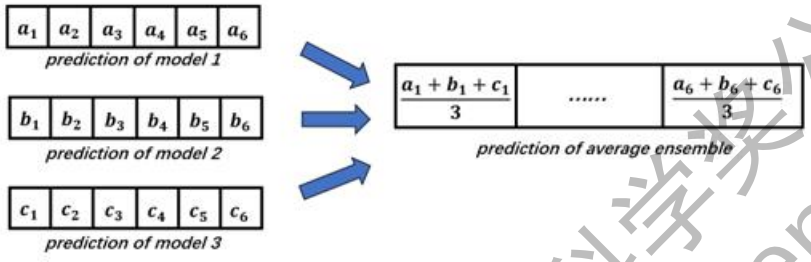

Figure 10: Average ensemble. The array on the left represents the result of the models before the ensemble, while the array on the right is the result of the ensemble of the models on the left.

3.3.2 Vote Ensemble

$$\begin{gathered}\textbf{if} A_n \text{ and } B_n \text{ is } 0 \textbf{ then } P_n \text{ is } 0 \\ \textbf{if} B_n \text{ and } C_n \text{ is } 0 \textbf{ then } P_n \text{ is } 0 \\ \textbf{if} A_n \text{ and } C_n \text{ is } 0 \textbf{ then } P_n \text{ is } 0 \\ \text{otherwise } P_N \text{ is } 1\end{gathered} \tag{2}$$

Vote ensemble is to decide the prediction by the decision of all the models. If most of the models made positive predictions, the result will be positive. Otherwise, the result will be negative. In equation 2, A_n, B_n, and C_n represent the predictions of different models before ensemble. P_n represents the result of ensemble.

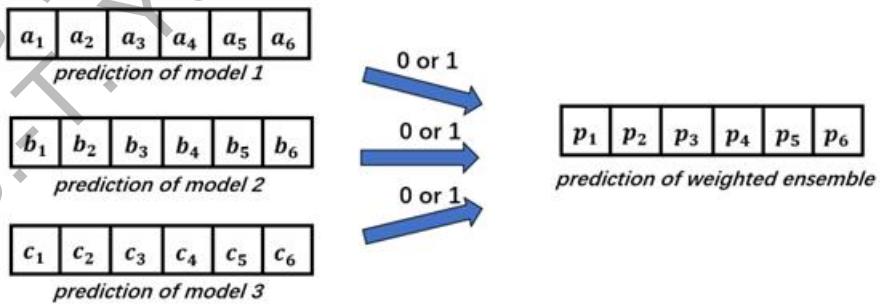

Figure 11: Vote ensemble

3.3.3 Weighted Ensemble

$$P_n = \frac{acc_A \cdot A_n + acc_B \cdot B_n + acc_C \cdot C_n}{acc_A + acc_B + acc_C} \tag{3}$$

Weighted ensemble is to calculate the average of results generated by all the models with the weight decided by their performance on the testing dataset. In this paper, accuracy is the standard of evaluation.

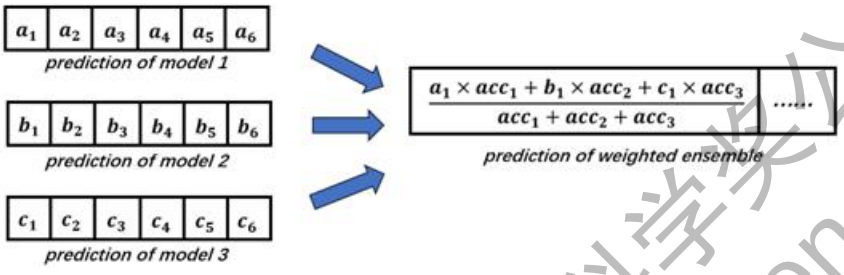

Figure 12: Weighted ensemble

3.4 Grad-cam

Gradient-weighted Class Activation Mapping(grad-cam) is a method that provides evidence for the classification that neuron networks make. In Figure 13, for example, the network classifies the picture as a nurse because of the equipment she uses and the uniform, which are both spotted red[21].

Figure 13: Output of grad-cam

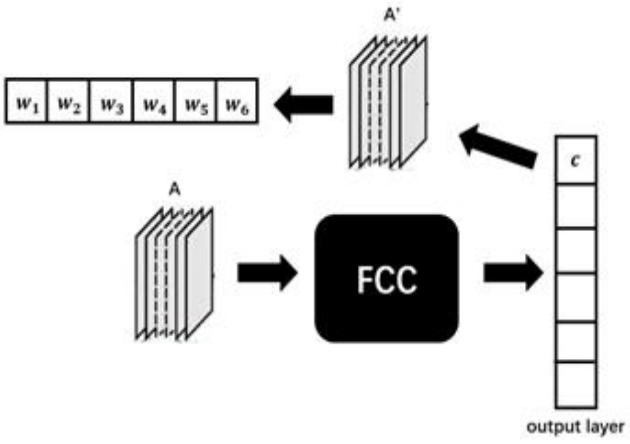

Figure 14: Process of grad-cam. The left-up array is array w.

According to the grad-cam algorithm, map A', composed of the gradient of result C with respect to the last feature map A, is calculated with the equation. The averages of all of the gradients of the same channel are calculated.

$$\alpha_k^c = \frac{1}{Z}\sum_i\sum_j \frac{\partial y^c}{\partial A_{ij}^k} \tag{4}$$

In the equation 4, α is the average weight of subtype c, channel k. Z is the area of the feature map A. y^c is the result that the model predicts of subtype c. A_{ij}^k represents the data of feature map A at positionij.

Put all the α in an array W. Each element in W is timed with the corresponding channel in A, and the average of all channels is calculated. The final heatmap is formed.

$$L^c = ReLU(\sum_k \alpha_k^c A^k) \tag{5}$$

In the equation 5, k is the channel of the feature map, c is the specific subtype.

3.5 Statistical Analysis

3.5.1 Confusion Matrix

Table 2: Confusion Matrix

	Predicted A	**Predicted B**
Actual A	TP	FN
Actual B	FP	TN

TP is the number of the picture which is A and is classified as A by the model. TN is the number of the picture which is B and is classified as B by the model. FP is the number of the picture which is B but is classified as A by the model. TN is the number of the picture which is A but is classified as B by the model. All the other scores are based on the confusion matrix.

3.5.2 Accuracy

$$accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

Accuracy can represent the performance of the model well in most circumstances. However, when the test dataset is imbalanced, accuracy doesn't work well. Image if the number of the picture of a certain class outweighs the other classes, and the model classifies all of the pictures as that class, the accuracy is still pretty high. Nevertheless, this model is not ideal from the perspective of users.

3.5.3 Recall

$$recall = \frac{TP}{TP + FN}$$

Recall measures the percent of correct classification among the positive data. Recall can solve the problem that is mentioned in the previous accuracy part in the case of this research since the positive data is significantly less than the negative data.

3.5.4 Precision

$$recall = \frac{TP}{TP + FP}$$

Precision is the percentage of the correct classified picture among all the pictures that are classified as positive.

3.5.5 F1-score

$$f1 - score = \frac{2}{2 + \frac{FP}{TP} + \frac{FN}{TP}}$$

F1 is the harmonic mean of recall and precision.

3.5.6 AUC

$$TPR = \frac{TP}{TP + FN} FPR = \frac{FP}{FP + TN}$$

TPR represents the correct prediction model made that is positive, while FPR represents the wrong prediction model made that is negative. The program will change thresholds, with higher of which the model will tend to predict the picture as positive. By changing thresholds, a series of TPR and FPR are generated. Plot a graph with FPR as the independent variable and TPR as the dependent variable. AUC is the area between the curve and the x-axis, as shown in Figure 15.

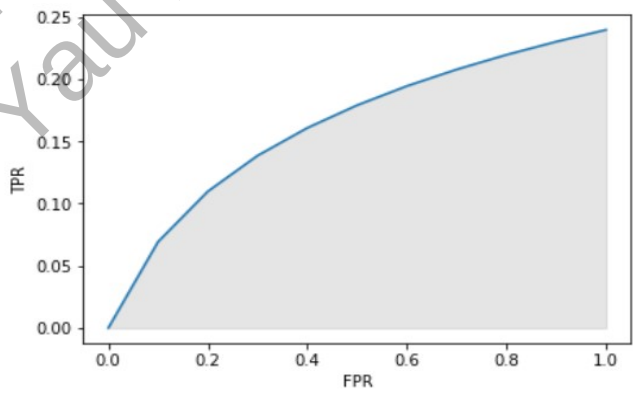

Figure 15: AUC curve. The area in gray is AUC.

3.6 Loss Function

Loss function is the standard that represents the performance of the model. Different from accuracy and recall, the lower the loss function is , the better the model performs. In this research, two kinds of loss functions are applied: binary cross entropy and weighted loss.

3.6.1 Binary Cross Entropy

$$Loss = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i))$$

y is the label and p(y) is the predicted possibility. Binary Cross Entropy is mainly used in classifying things into exactly two classes. In this paper, since a single CT image can have many types of ICH, models predict 1 time whether the image has ICH and 1 time, for each subtype, whether the image has a specific subtype of ICH.

3.6.2 Weighted Loss

In this research, the model aims at determining whether CT images have ICH and classify the subtype of ICH. However, determining whether ICH exists is significantly more important than classifying subtypes. However, although binary cross entropy can accurately measure the loss of each model, it can't find out which class is more important. The solution to this problem is to add weight to binary cross entropy. By multiplying the result with the given weight, the loss with greater weight will affect the final outcome more than others. In this research, type 'any' is weighted 5 while others are 1.

4 Result

4.1 Experimental Design

4.1.1 Datasets

In the paper, cross-validation is used to ensure performance. There are total of 200000 images. 10 percent of the dataset is the testing dataset, while 81 percent is training and 9 percent is the validation dataset.

4.1.2 Hyperparameter

All details are shown in the Table 3. All of the hyperparameters are choosen after experimental trails.

Table 3: Employed models with corresponding parameters.

	Leaning Rate	Patient	Activation Equation	loss	Optimizer	Monitor	Batch Size	Dropout	number of parameter
EfficientNet	0.0005	3	Sigmoid	Binary Cross-entropy	Adam	Accuracy	4	0.125	7949695
ResNet	0.0005	3	Sigmoid	Binary Cross-entropy	Adam	Accuracy	4	0.125	23850758
DenseNet	0.0005	3	Sigmoid	Binary Cross-entropy	Adam	Accuracy	4	0.125	7169478
Optimized EfficientNet	0.0005	3	Sigmoid	Binary Cross-entropy	Adam	Accuracy	4	0.125	7949695
Optimized ResNet	0.0005	3	Sigmoid	Binary Cross-entropy	Adam	Accuracy	4	0.125	23850758
Optimized DenseNet	0.0005	3	Sigmoid	Binary Cross-entropy	Adam	Accuracy	4	0.125	7169478

4.1.3 Device

Our models were trained with a mini-batch size of 4 with a GA104 [GeForce RTX 3070] GPU.

4.1.4 Testing and Training Time

Table 4: Summary of test and training time of windowed pre-processed images.

	Train Time(s)	**Test Time(s/image)**
EfficientNet	46543	0.01
ResNet	17445	0.0096
DenseNet	19030	0.011
Optimized EfficientNet	31212	0.017
Optimized ResNet	21803	0.0115
Optimized DenseNet	33129	0.012

Table 5: Summary of test and training time of ASIP pre-processed images.

	Train Time(s)	**Test Time(s/image)**
EfficientNet	23098	0.027
ResNet	38129	0.022
DenseNet	43847	0.025
Optimized EfficientNet	25323	0.024
Optimized ResNet	49983	0.021
Optimized DenseNet	19834	0.026

As shown in tables 4 and 5, the training time of each model varied greatly since different architectures(refer to the model before fully connected layer) have different numbers of parameters. As expected, more time is required for a model with more parameters to converge. In terms

of test time, the baseline ResNet with windowed pre-processing is the fastest, while the baseline EfficientNet is the slowest. It can also be observed that models with ASIP pre-processing need longer test and training time, which is presumably because they need to find the pre_images and post_images for almost 75,000 images.

4.1.5 Baseline Model

In this paper, EfficientNetB2, ResNet50, and DenseNet121 were applied as the architectures of the baseline models. As shown in figure 16, the model is composed of an architecture, which is either EfficientNet, ResNet, or DenseNet, an average pooling layer, a drop out layer, an fully connected layer(FCC) with a width of 128, an output layer of width 4, and a sigmoid activation layer.

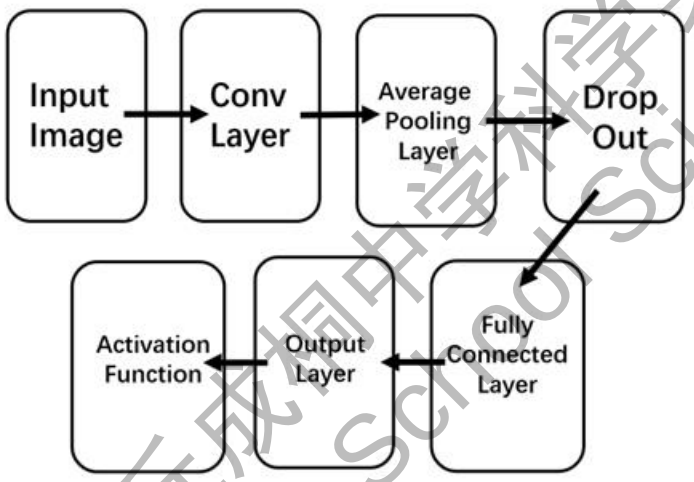

Figure 16: The structure of the baseline model used in this project.

4.1.6 Customizations

The baseline models were customized to fulfill our application. Firstly, since determining whether the CT images show ICH is much more important than classifying the ICH subtypes, weighted loss calculations were used. The output that determines whether a case is ICH positive was weighted by 5, while the output that determines the subtype was weighted by 1. Secondly, data augmentation was applied to enhance the robustness of our models. Since the CT images are not always along the same direction, the models should be able to identify ICH regardless of the image orientation. Hence, two functions, "imgaug.augmenters.Fliplr()" and "imgaug.augmenters.Fliplr()", which mirror the image horizontally and vertically, were applied before the data was input to the models.

4.2 Experimental Result

4.2.1 Performance

We first compared the performance of all six models including the windowed pre-processed and AISP pre-processed baseline models as well as their optimized models, as shown in figures 17 to 21.

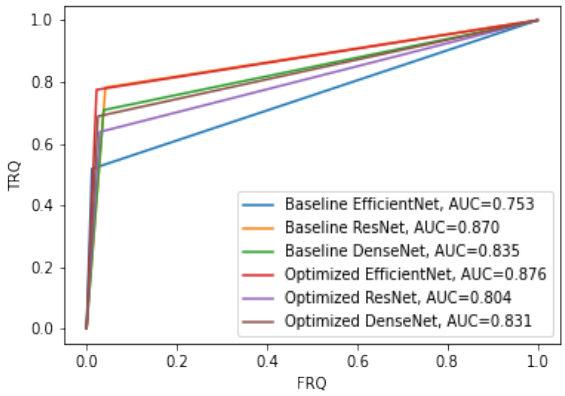

Figure 17: Plot of the TRQ against the FRQ with calculated AUCs of all six models with windowed pre-processing.

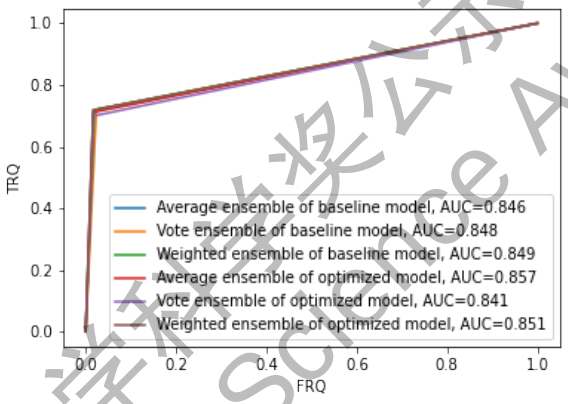

Figure 18: Plot of the TRQ against the FRQ with calculated AUCs of the average, vote, and weighted ensembles of the baseline and optimized models with windowed pre-processing.

Table 6: Summary of the performance calculations of the models with windowed preprocessing

	Accuracy	F1-score	Recall	Precision	AUC
EfficientNet	0.920	0.653	0.518	0.882	0.753
ResNet	0.932	0.771	0.782	0.761	0.870
DenseNet	0.925	0.734	0.709	0.762	0.835
Optimized EfficientNet	0.950	0.813	0.774	0.856	0.876
Optimized ResNet	0.922	0.706	0.637	0.792	0.804
Optimized DenseNet	0.932	0.750	0.688	0.823	0.831
Average Ensemble of Baseline	0.942	0.783	0.711	0.870	0.846
Vote Ensemble of Baseline	0.939	0.775	0.720	0.838	0.848
Weighted Ensemble of Baseline	0.942	0.783	0.719	0.862	0.849
Average Ensemble of Optimized	0.944	0.787	0.711	0.879	0.857
Vote Ensemble of Optimized	0.941	0.775	0.699	0.870	0.841
weighted Ensemble of Optimized	0.944	0.858	0.813	0.909	0.851
best model	0.947	0.813	0.784	0.843	0.880

As shown in table 6, which demonstrates the performance of the models with windowed pre-processing, the EfficientNet performs much better after data augmentation and weighted loss (AUC increases from 0.753 to 0.876). The performance of DenseNet almost remains unchanged after the optimization(AUC decreases from 0.835 to 0.831). However, the baseline of ResNet(AUC of 0.870) significantly outperforms its optimized version (AUC of 0.804). The ensembles outperform most of the models except for the baseline of ResNet and the optimized EfficientNet. All the ensembles of the optimized models perform better than the ensembles of baseline models. Among these ensembles, the weighted ensemble ranks the top. After trying various combinations, we found that the weighted ensemble of optimized EfficientNet and baseline ResNet performed the best, achieving an accuracy of 0.947, an f1-score 0.813, a recall of 0.784, a precision of 0.843, and an AUC of 0.880.

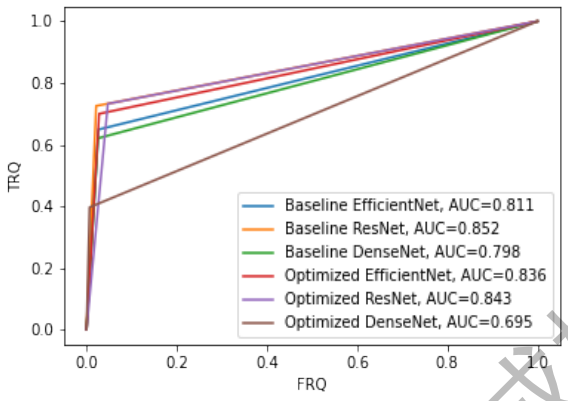

Figure 19: Plot of the TRQ against the FRQ with calculated AUCs of all six models with ASIP pre-processing.

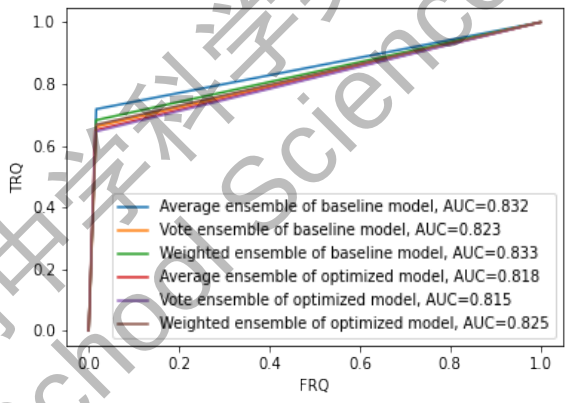

Figure 20: Plot of the TRQ against the FRQ with calculated AUCs of the average, vote, and weighted ensembles of the baseline and optimized models with ASIP pre-processing.

Table 7: Summary of the performance calculations of the models with ASIP pre-processing.

	Accuracy	F1-score	Recall	Precision	AUC
EfficientNet	0.926	0.719	0.649	0.807	0.811
ResNet	0.942	0.784	0.725	0.853	0.852
DenseNet	0.923	0.702	0.620	0.808	0.798
Optimized EfficientNet	0.932	0.751	0.700	0.811	0.836
Optimized ResNet	0.921	0.729	0.732	0.726	0.843
Optimized DenseNet	0.906	0.552	0.396	0.910	0.695
Average Ensemble of Baseline	0.940	0.767	0.681	0.878	0.832
Vote Ensemble of Baseline	0.937	0.753	0.661	0.874	0.823
Weighted Ensemble of Baseline	0.940	0.767	0.681	0.878	0.833
Average Ensemble of Optimized	0.937	0.749	0.650	0.885	0.818
Vote Ensemble of Optimized	0.935	0.744	0.645	0.878	0.815
weighted Ensemble of Optimized	0.937	0.756	0.668	0.872	0.825
best model	0.943	0.788	0.729	0.858	0.854

As shown in table 7, which demonstrates the performance of the models with ASIP, the EfficientNet and ResNet both perform better after optimization . However, the baseline DenseNet performs significantly worse than other models. Thus, unlike the models with windowed pre-processing, all ensembles of the baseline models perform better than those of the optimized models. Similarly, the both the weighted ensemble of baseline and optimized model perform better than average ensemble and vote ensemble of baseline and optimized model. The best model was the weighted ensemble of the baseline and the optimized ResNet and EfficientNet with an accuracy of 0.943, an f1-score of 0.788, a recall of 0.729, a precision of 0.858, and an AUC of 0.854.

Table 8: Summary of the performance calculations of the baseline and optimized models.

	Accuracy	F1-score	Recall	Precision	AUC
Baseline Average Ensemble	0.942	0.778	0.694	0.886	0.839
Baseline Vote Ensemble	0.937	0.751	0.647	0.897	0.817
Baseline Weighted Ensemble	0.943	0.780	0.700	0.882	0.841
Optimized Average Ensemble	0.941	0.771	0.679	0.892	0.833
Optimized Vote Ensemble	0.938	0.751	0.641	0.907	0.815
Optimized Weighted Ensemble	0.942	0.778	0.692	0.888	0.839
best model	0.949	0.814	0.771	0.862	0.875

As shown in table 8, the ensembles of baseline models and the optimized models show similar performance. However, since the models with ASIP generally perform worse than those with windowed pre-processing, the performance of the ensemble of all baseline model and all optimized model are worse than the performance of ensemble of baseline model and optimized model with window preprocess but is better than the ensemble of baseline model and optimized model with ASIP. The best model among all models is the weighted ensemble of the baseline ResNet model and the optimized EfficientNet model with windowed pre-processing, and the baseline ResNet

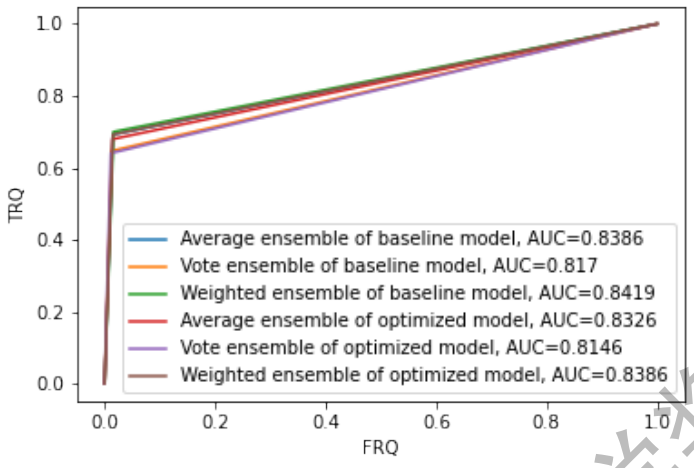

Figure 21: Plot of the TRQ against the FRQ with calculated AUCs of the average, vote, and weighted ensembles of the baseline and optimized models.

with ASIP with an accuracy of 0.949, an f1-score 0.814, a recall 0.771, a precision of 0.862, and an AUC of 0.875.

One possible explanation for the superiority of the windowed pre-processed group over the ASIP group is that the ASIP group discarded many values below 0 and above 80. Those values may be quite important for the classification of ICH. In addition, the ASIP group may not be able to establish a connection between adjacent images.

4.3 Visualization

Grad-cam provides evidence for the classification made by the model.

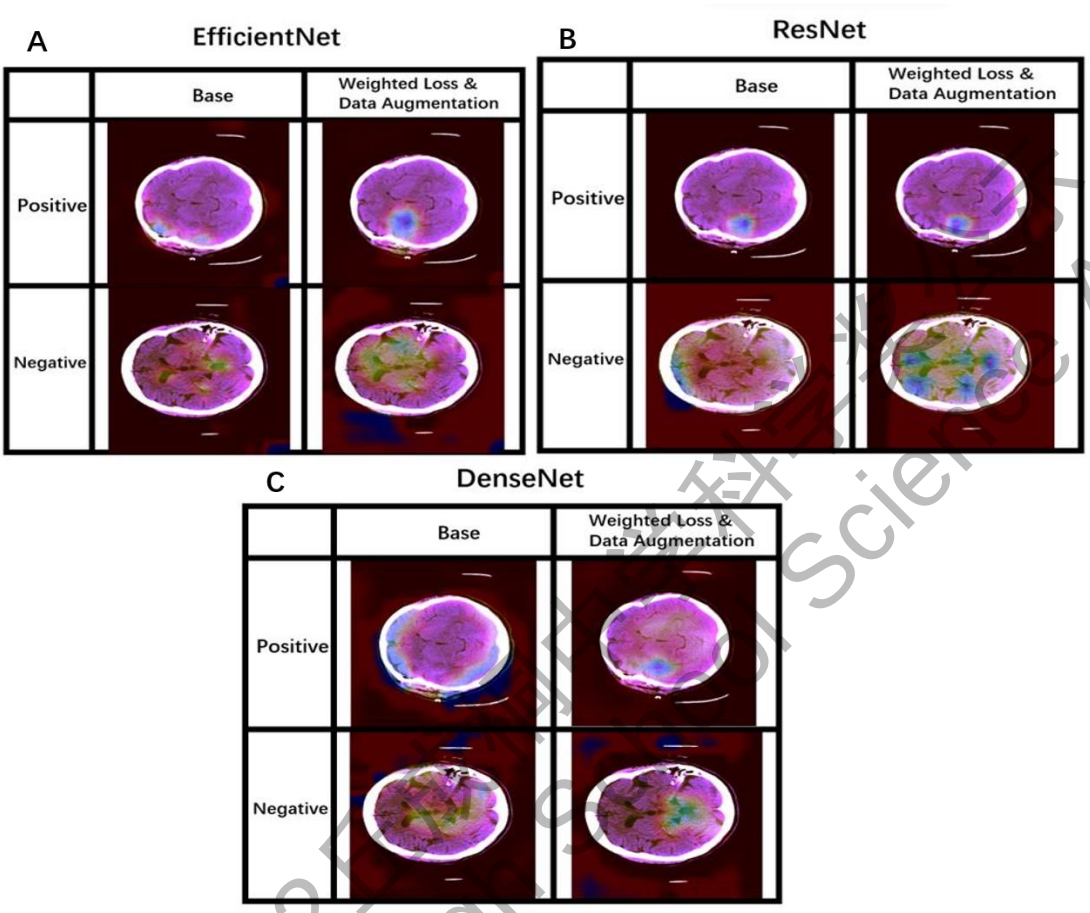

Figure 22: Grad-cam of the output images from (A) the EfficientNet, (B) the ResNet and (C) the DenseNet with windowed pre-processing. The images at the top left are the grad-cam images of the baseline models when the predicted images contain ICH. The top right grad-cam images show the outputs from the optimized models when the predicted images have ICH. The bottom left images are the grad-cam images of the baseline models when the images being predicted do not have ICH. The bottom right images are the grad-cam pictures of the optimized models when the pictures being predicted do not have ICH.

As shown in Figures 17, 18, and 19, for the EfficientNet and DenseNet, optimized models can locate the ICH precisely, while baseline can't, though they can locate to the bottom of the CT image, which is the general position of the ICH. However, both baseline and optimized models of ResNet can find the accurate position of ICH. When the CT image happens to be negative, the model won't locate a specific position. Instead, the color of the image is quite similar or randomly distributed.

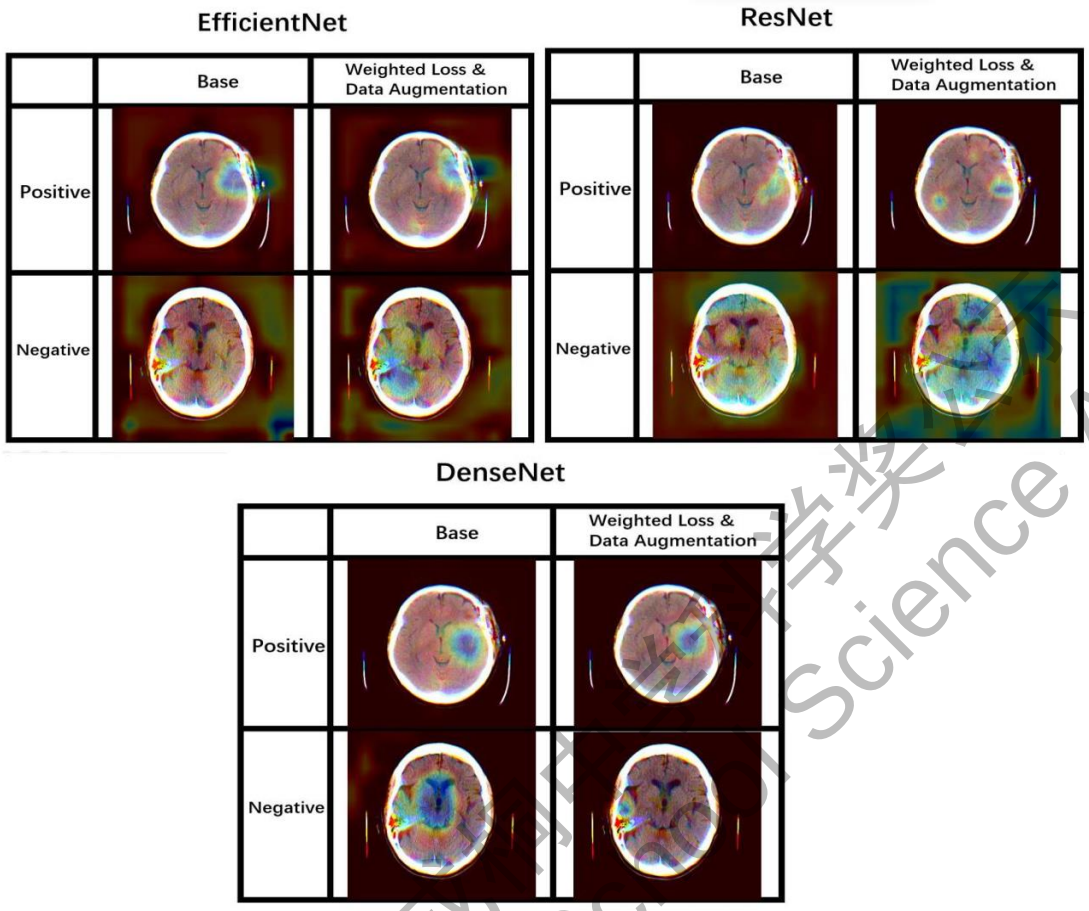

Figure 23: Grad-cam of the output images from (A) the EfficientNet, (B) the ResNet and (C) the DenseNet with ASIP pre-processing. The images at the top left are the grad-cam images of the baseline models when the predicted images contain ICH. The top right grad-cam images show the outputs from the optimized models when the predicted images have ICH. The bottom left images are the grad-cam images of the baseline models when the images being predicted do not have ICH. The bottom right images are the grad-cam pictures of the optimized models when the pictures being predicted do not have ICH.

As shown in figures 23, all models precisely located the location of ICH when predicting images with positive labels. When there is no ICH in the brain, no specific location will be focused and the mappings are random.

5 Discussion

5.1 Overview

In this study, DL was employed to detect the presence and location of ICH with 12 proposed models (baseline, data augumented and weight (optimized) models for EfficientNet, ResNet, DensNet with two pre-processing methods: windowed and ASIP). Their ensembles were also

calculated of which the best performance was with an accuracy of 0.945, an f1-score of 0.890, a recall of 0.813, and a precision of 0.984.

One of the main reasons radiologists are concerned about using DL to help determine ICH is that DL is a black box and it provides no clue as to why it makes such judgments. In this study, the processed CT images were presented through a grad-cam, in which the locations of ICH were labeled with different colors. Such labels clearly showed the key regions by which the algorithm determines if the patient had ICH. In this case, radiologists could simply compare the algorithm-labeled regions with their own judgments to validate if the algorithm truly recognized ICH by the corresponding abnormal features. Furthermore, radiologists could potentially save a lot of time by simply looking at the grad-cam-labeled regions instead of the whole raw image. Hence, our model is expected to improve the diagnosis of ICH in terms of both effectiveness and efficiency.

Instead of directly employing architectures designed by other computer scientists, ensembles, including average, vote, and weighted ensembles of those architectures were applied in our study since they provided better outcomes than any single architecture. In addition, these ensembles also increased the generalization ability of our model , since during the training process, different models may be better at classifying different subtypes.

In addition to determining if a patient has ICH, our model can also determine the subtype of ICH, which may greatly help doctors design the treatment plan. Together with our specially designed user-friendly APP, we believe that our model could provide great assistance to radiologists and doctors who fight against ICH, and therefore save more lives.

5.2 APP Demonstrate

The layout design of the APP is shown in figure 24. User can click the folder icon to select the CT images, which will then be analyzed by our models. When the analysis is finished, the APP directs to the 'further info' interface. In this interface, basic information about the patient, the ICH classification result and the grad-cam labelled CT images are displayed. The APP can also store the previous classification results, which can be browsed in the 'history' interface. Contact details of the developer of the APP can be found in the 'Contact us' interface.

Figure 24: App layouts showing (A) the starting page, (B) the history page, which shows a list of the patient details, (C) the further Infor page that contains the output grad-cam images from our DL training models and (D) the contact us page with the developer's contact details.

5.3 Limitations and Future Work

Despite the merits mentioned above, our models also have some limitations. Firstly, since the data from Kaggle[17] is imbalanced in number, our models may be trained insufficiently in classifying some subtypes. In our further research, more CT images will be added to the training dataset to improve the performance of our models. Secondly, all our tests were carried out on the public dataset, i.e., our models have not been employed in realistic clinical situations. Hence, we plan to collaborate with hospitals to obtain more real-world clinical data to validate the practicability of our models.

Other potential improvements include: 1) we can utilize more complicated and powerful models for better classification accuracy and speed. 2) by performing analysis on the CT images of the same patient at different time points, our model may be able to predict the pathological development of ICH, facilitating early diagnosis and treatment of ICH. 3) We can include more functions into the APP, such as showing potential treatment plans, tracking patient status, and predicting the prognosis of the patient, etc.

6 Conclusion

In this paper, 12 DL models and their ensembles were designed and trained to identify if a patient has ICH by CT images as well as to classify the specific ICH subtypes. The best model reached an accuracy of 0.949, an f1-score of 0.814, a recall of 0.771, a precision of 0.862, and an AUC of 0.875. On top of the best model, we designed an APP that can be installed on mobile devices for the convenient and rapid diagnosis of ICH.

References

[1] J. L. Clarke, S. C. Johnston, M. Farrant, R. Bernstein, D. Tong, and J. C. Hemphill, "External validation of the ich score," *Neurocritical Care*, vol. 1, no. 1, pp. 53–60, 2004.

[2] A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Žídek, A. W. Nelson, A. Bridgland *et al.*, "Improved protein structure prediction using potentials from deep learning," *Nature*, vol. 577, no. 7792, pp. 706–710, 2020.

[3] G. Eraslan, Ž. Avsec, J. Gagneur, and F. J. Theis, "Deep learning: new computational modelling techniques for genomics," *Nature Reviews Genetics*, vol. 20, no. 7, pp. 389–403, 2019.

[4] C. Pan, O. Schoppe, A. Parra-Damas, R. Cai, M. I. Todorov, G. Gondi, B. von Neubeck, N. Böğürcü-Seidel, S. Seidel, K. Sleiman *et al.*, "Deep learning reveals cancer metastasis and therapeutic antibody targeting in the entire body," *Cell*, vol. 179, no. 7, pp. 1661–1676, 2019.

[5] Y. Anzai, *Pattern recognition and machine learning*. Elsevier, 2012.

[6] L. Lucas and R. Boumghar, "Machine learning for spacecraft operations support-the mars express power challenge," in *2017 6th International Conference on Space mission challenges for information technology (SMC-IT)*. IEEE, 2017, pp. 82–87.

[7] M. F. Dixon, I. Halperin, and P. Bilokon, *Machine learning in Finance*. Springer, 2020, vol. 1406.

[8] C. Angermueller, T. Pärnamaa, L. Parts, and O. Stegle, "Deep learning for computational biology," *Molecular systems biology*, vol. 12, no. 7, p. 878, 2016.

[9] I. El Naqa and M. J. Murphy, "What is machine learning?" in *machine learning in radiation oncology*. Springer, 2015, pp. 3–11.

[10] D. Lei, X. Chen, and J. Zhao, "Opening the black box of deep learning," *arXiv preprint arXiv:1805.08355*, 2018.

[11] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C. Van Esesn, A. A. S. Awwal, and V. K. Asari, "The history began from alexnet: A comprehensive survey on deep learning approaches," *arXiv preprint arXiv:1803.01164*, 2018.

[12] X. Xu and C. Wang, "A new deep learning model based on improved alexnet for radiation source target recognition," in *Proceedings of the 2018 International Conference on Data Science and Information Technology*, 2018, pp. 1–5.

[13] E. Kozegar, "Cystoscopic image classification by an ensemble of vgg-nets," *International Journal of Nonlinear Analysis and Applications*, vol. 12, no. 1, pp. 693–700, 2021.

[14] Z. Zhang, H. Zhang, L. Zhao, T. Chen, S. Ö. Arik, and T. Pfister, "Nested hierarchical transformer: Towards accurate, data-efficient and interpretable visual understanding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 3, 2022, pp. 3417–3425.

[15] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby, "Big transfer (bit): General visual representation learning," in *European conference on computer vision*. Springer, 2020, pp. 491–507.

[16] B. Heo, S. Yun, D. Han, S. Chun, J. Choe, and S. J. Oh, "Rethinking spatial dimensions of vision transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 11 936–11 945.

[17] R. S. of North America. (2019) Rsna intracranial hemorrhage detection. [Online]. Available: https://www.kaggle.com/c/rsna-intracranial-hemorrhage-detection

[18] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*. PMLR, 2019, pp. 6105–6114.

[19] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1181–1193, 2018.

[20] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[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," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

Acknowledgement

I became aware of ICH a year ago. The father of my baseball teammate died abruptly last summer because of a stroke, which is likely caused by ICH. Everything happens all of a sudden just like a dream. Since my teammate's parents divorced, the family left only himself and his little sister. Fortunately, many people helped them and they finally get out of the sorrow and keep living positively. From then on, I started to search for some documents about ICH. I want to develop a standard process that every patient can be inspected in minutes. This research is the outcome.

I would like to first express my sincere gratitude to my parents. They are experts in the medical field, and I often discuss health care issues with them. On the one hand, they first evoke my interest in the health care field, which is another important factor for me to do this research. On the other, they provided me with many chances to communicate with experts in this field and paid the expensive fee of renting GPU on the one hand. The research won't conduct as well as it is now without the help of my parents.

I would then like to thank Yunan Wu and Xiangmei Dou. They will answer my question carefully and provide very useful suggestions and professional guidance to make this research better. Besides, Dr. Wu always cheers me up when I was distressed by the performance of the model is not ideal.

Last but not least, I would like to thank Jinzhuo Wang from Beijing University. Dr. Wang provided practical suggestions for the improvement of the essay.