Linköping University | Department of Medicine and Health
Master's thesis, 30 ECTS | Radiation Physics
2019 | LIU-IMH/RV-A-19/001-SE

Classifying Liver Fibrosis Stage Using Gadoxetic Acid-Enhanced MR Images

Yi Cheng Lu

Supervisor : MSc Markus Karlsson
Radiation Physics, Linköping University

Examiner : Professor Peter Lundberg
Radiation Physics, Linköping University

LINKÖPING UNIVERSITY

Linköpings universitet
SE–581 83 Linköping
+46 13 28 10 00 , www.liu.se

Upphovsrätt

Copyright

Acknowledgments

First of all, I am very grateful to my supervisor, Markus Karlsson, for his advising in the past 20 weeks. Without his support, I would not get familiar with the MRI data set that quick. Also, he helped me to coordinate with other staffs in the hospital, which is really helpful when I had any question about the data.

Second, I also thank my examiner, professor Peter Lundberg. He also gave some advice to my project and some essentially support such as helping me to get a new computer with powerful GPU. Without this hardware support, I undoubtedly could not finish all the task in time.

Third, I thanks the Dr. Chunliang Wang from KTH who gave me really practical suggestions during my midterm report, which really guide me a clear path in my second half of the experiment.

At last, I sincerely thank again all of the people that help me to fulfill this master thesis. As an international student, I also thank to their tolerance of my English skill and help me to complete this thesis as well as this report. I will remember all the things that they taught me and using all these skills in my own future works.

Thank you all.

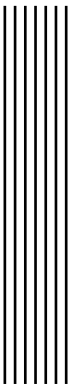
Contents

List of Figures

Abstract

In the very beginning, a method proposed by one Korean group is being examined and trying to reproduce their result. However, the performance is not as impressive as theirs. Since the image used in this thesis actually are different from theirs, it is concluded that the method they proposed is not very general or just not feasible to transfer to different kind of image type. To develope model that fits to our data set, first, their procedure has been adopted to develop a new version of the index. Furthermore, some gray-scale image feature extraction methods are used, and the resulting features are plugged into machine learning algorithms which used as classifiers. Last but not least, the hottest method in recent years - Convolution Neural Network(CNN) was utilized. All of our models are tested under two data sets: one is the same source as our training data and the other one is totally new data set(from different MRI machine and different MRI image parameters) that did not include during the training. So the generality of our models can be tested. On the other hand, one simple method - NLE indicator, is used as our baseline(AUC=0.76).

The result shows that with manual feature extraction, the Adaboost model works pretty well that AUC achieves 0.9. Besides, the AUC of ResNet-18 network - a deep learning architecture, can reach 0.93. Also, all the hyperparameters and training setting used on ResNet-18 can be transferred to ResNet-50/ResNet-101/InceptionV3 very well. Those deeper networks are also trained and examined. The best model that can be obtained is ResNet-101 which has an AUC of 0.96 - higher than all current publications for machine learning methods for staging liver fibrosis.



Introduction

Liver fibrosis is a common sequela of liver injury and an indicator of more severe liver disease such as cirrhosis, which will lead to much more severe liver conditions such as portal hypertension, liver failure and liver cancer.[1, 2] To detect liver fibrosis in the early stage is, therefore, an important task. Currently, the liver fibrosis is detected by liver biopsy which is a "**gold standard**"[3], however, this method is invasive and usually will have inconsistency due to sampling, as well as inter and intra reader variability. Also, some patients might refuse to accept this invasive method and, as a consequent, delay the treatment. Moreover, there actually is some risk of this invasive method including bleeding, infection and accidental injury to a nearby organ[1]. Also, since it is kind of surgery, the patient needs to be taken to a recovery room to rest which is time-consuming. Overall, developing a non-invasive and rapid method to stage liver fibrosis brings many benefits.

Currently, there are two main basic approaches in non-invasive testing: blood test for specific biomarkers, and imaging methods. Blood test needs lots of biology knowledge such as proteomics to develope. On the other hand, blood test is trying to tell us whether there's little fibrosis or a lot of fibrosis but not all the gradient in between, which cannot fulfill our goal since we are focusing on staging the fibrosis.[2] In this thesis, the main focus is on imaging methods.

One common field in imaging technique is *Ultrasonography*. However, this method is not so accurate[4] and depend a lot on the examiner(doctor) who conduct the ultrasound inspection for the patient. On the other hand, this method is usually capable of finding the liver fibrosis in the late stage but not the early stage.

There are some improved methods that are also related to *Ultrasonography* e.g. *sonographic elastography*[5]. However, there is also some disadvantage to this method. *Sonographic elastography* cannot detect the fibrosis stage in cases of inhomogeneous fibrosis or in the cases of obese patients or in patients with ascites.[4] There are still lots of variety of ultrasound-based tests such as acoustic radiation force impulse imaging (ARFI), and real-time shear wave elastography (SWE). All of them essentially evaluate the stiffness of the liver, because the more the fibrosis the patient has, the stiff the liver becomes.[2,3] But they all rely heavily on the examiner. To develop methods that do not rely on human judgement might be interesting.

[1] https://www.mayoclinic.org/tests-procedures/liver-biopsy/about/pac-20394576

[2] https://liverline.com/a-brief-overview-of-non-invasive-testing-in-liver-fibrosis-46901565afac

[3] http://www.mmh.org.tw/gi/index4_8_4.html

Recently, more and more methods are developed depend on several kinds of MR and CT image of the liver. Using the method in *Computer Vision* which includes lots of different ways of image processing and feature extraction. Since the CT or MR image are both grayscale image, most of the method of feature extraction is related to GLCM and GLRLM.[6, 7, 8, 9]. With the development of image processing, these methods seem to be a powerful candidate to detect liver fibrosis. The main reason that these methods work is that the fibrosis liver has a significant texture different which is quite obvious in the image.[10]

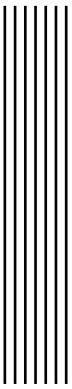
*W*ith the rapid progress in recent years in Deep learning, there are also more and more methods based on the Convolution Neural Network(CNN) [11, 12, 13]. These methods will all be leveraged in this thesis in order to fulfill our work.

Questions to be answered

- Do the formula proposed in the paper[6] also works on our data set? How good or bad is the performance?
- Is there any traditional machine learning model with manual feature extraction that works better on our data set than baseline does? How good or bad is the performance?
- Is there any deep learning network that works better on our data set than baseline does? How good or bad is the performance?
- How general all derived models are?

Report structure

In this thesis report, I will first give a brief introduction to liver fibrosis and MRI image. Second, I will give an introduction to the matrix I used to extract features in the gray-scale image and some basic of CNN include some detail during training. Also, I will briefly introduce the data set used in the thesis. Third, I will present the result and provide some analysis as well as the discussion. Last, I will make a conclusion about the entire process and put forward the future work that is needed in order to bring this concept into practice.



Background

Liver Fibrosis

Liver fibrosis is caused after a person's liver got injury or inflammation and the liver cell want to heal the wound. During the healing, excess proteins accumulate in the liver. Eventually, after several healing procedures, the excess proteins form fibrosis in the liver. More detail of the protein accumulate procedure can be found in [14]. With more and more liver fibrosis, finally, the liver will become cirrhosis. Cirrhosis will later cause **liver cancer**. Liver cancer is the 5^{th} highest death number of cancer and also the 6^{th} highest new cases of cancer among the world in 2018[15, 16]. One of the popular scoring system is the **METAVIR** scoring system. Doctor will assign a score according to the "activity" of how the fibrosis is progressing, and for the fibrosis level itself. The activity grades range from A0 to A3 which is not the focus of this thesis. On the other hand, the fibrosis stages range from F0 to F4:[1,2]

- F0: No fibrosis
- F1: Portal fibrosis without septa
- F2: Portal fibrosis with few septa
- F3: Numerous septa without cirrhosis
- F4: Cirrhosis

Ideally, doctors are expected to find the liver fibrosis as earlier as possible before it becomes cirrhosis. However, it is hard to achieve because the conditions usually don't cause any syndrome in the early stage. If it can be detected at early stage, with proper treatment, the inflammation that happens in the liver can be stopped. The liver with little fibrosis tissue even actually has the opportunity to repair itself and then return to the original liver, or stagnant in the early stage, so as not to deteriorate. On the contrary, if the fibrosis activity did not discovered in time, fibrosis tissue can reduce the liver functionality and impair the liver's ability to regenerate, which will eventually kill healthy tissue and create more scar in the liver. Then, inflammation will continue and lead the liver to develop into more severe fibrosis

[1] https://www.healthline.com/health/liver-fibrosis#symptoms

[2] http://labmed.ucsf.edu/uploads/472/227_Ferrell,%20LiverUpdateOnStagingOfFibrosisAndCirrhosis.pdf

stage.[3] Consequently, to detect the fibrosis in an early stage is very important. In order to achieve the goal, a more convenient and faster examination is necessary.

Magnetic Resonance Imaging

Magnetic Resonance Imaging, known as MRI, is a medical imaging technology. It was first published by Paul Lauterbur in 1973 in journal *Nature* which showed tubes of water. Later in 1997, Peter Mansfield developed a fast imaging technique called Echo-Planar Imaging(EPI), which get rid of the effect of motion during MRI scanning[17]. Now MRI is one of the most important tools used in medical radiology diagnosis.

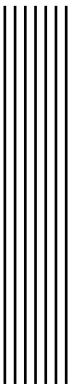
Unlike CT using X-rays or radiation, MRI using a strong static magnetic field, magnetic field gradient and radio frequency(RF) to generate image. Nucleus has its own intrinsic property called *spin*. Because of this "*spin*", the nucleus will form there own magnetic moment just like a small magnet. MRI is the technique that takes advantage of this property. First, using a strong static magnetic field to let the nucleus distribute regularly (parallel or antiparallel to the magnetic field). Then, using RF to disturb the nucleus cause a phenomenon called *precession*(the RF frequency should be the same as precession frequency which is related to the static magnetic field). After the disturbance, the signal of the magnetic moment difference is observed. Moreover, to get the spatial information, the magnetic field gradient is used which let nucleus at different position experience different magnetic field and has different magnetic moment variation. To obtain a 3D body image, there are three gradient coils, each corresponding to one of the three directions - X, Y, Z axis. By analyzing these signals, the profile of our body can be reconstructed. Today, almost all MRIs using hydrogen as the target nucleus, since it's a component of water, which makes up 70% of our body.

All MRIs are produced using a *pulse sequence* which contains *RF pulses* and *gradient pulses*. Both of them have timing values called **TR** and **TE**. The operator will set different **TR** and **TE** to get different image contrast. MRI uses some property of hydrogen to distinguish different tissue. The most important properties are the *proton density* and two characteristic times called *spin-lattice relaxation time* and *spin-spin relaxation time*, denoted $\mathbf{T}_1$ and $\mathbf{T}_2$ respectively. Proton density is related to the number of hydrogen in a certain volume and $\mathbf{T}_1$, $\mathbf{T}_2$ depends on the different tissues. In general, images have different contrast depending on either PD, $\mathbf{T}_1$, $\mathbf{T}_2$.[17]

Gadoxetic Acid-enhanced

Gadoxetic acid(Gd-EOB-DTPA) is one of the contrast-enhanced agent used in MR imaging. A contrast-enhanced agent is a substance works as its name suggest which can increase the contrast of structure or fluids within the body in medical imaging[4]. Gadoxetic acid(Gd-EOB-DTPA) has a higher protein-binding capability and cause shortening of the longitudinal relaxation time (T1) of the liver. It is one of the hepatobiliary-specific contrast agents. In the liver, this hepatobiliary contrast-enhanced agent can improve lesion detection. Because of the characteristic of gadoxetic acid, superior enhancement can be obtained with lower dose compared with other hepatobiliary-specific contrast agents. A nice enhancement image can be acquired about 20 minutes after intravenous injection. Later the gadoxetic acid will be eliminated through the renal and hepatobiliary tracts.[18, 19]

[3] https://www.medicalnewstoday.com/articles/325073.php
[4] https://en.wikipedia.org/wiki/Contrast_agent



Methods

Image feature extraction

Introduction

There are lots of elements can be used as the "features" of the image such as shape, edge, image intensity, histogram, and texture. In this thesis, the focus will be on the **texture features** that accompany the image intensity. The main reason is that only texture features are used in the results of the paper[6] that are attempted to be reconstructed. Also, texture features are obvious for human eyes to observe directly.

Among all the texture feature extractions, the focus is on two kinds of matrices that are dedicated to the grayscale images, called GLCM and GLRLM. Following is a brief introduction of those two matrices. One thing needs to be concerned is these two kinds of matrices are somehow like the summary of the image(inside the Region of Interest(ROI)). Further operations are needed on those matrices to get the "features" that belong to the image. The following section will also briefly mention the further processing required for the matrix to get the final "features" that used in the later experiment.

GLCM

Gray-Level Co-occurrence Matrix(GLCM). The GLCM describes a pixel's relationship to a particular distance or nearby pixel within a particular area. In a more comprehensible view: *GLCM describes the appearance probability of a pair of pixels which arrange in certain pattern in a gray scale image.*

Symbol

The gray level in our image is **Bin size**, denoted as **B**.
The distance between a pair of pixels, denoted as **D**.
The size of a certain area is the **Window size**, denoted as **W**.
The angle used to find the pair, denoted as θ. There are four degrees that will be used: $0^\circ, 45^\circ, 90^\circ, 135^\circ$.

Procedure

- Since GLCM records the number of each kind of pairs of pixels, the size of GLCM is B^2. Each entry in the matrix stores the number of pairs, and it is obvious to observe that GLCM is symmetric since, for example, pair(1,2) is the same as pair(2,1). However, (1,2) can also be viewed differently from (2,1) and then form an unsymmetrical GLCM. Both of them are reasonable.
- The computing of GLCM is simple. First, decide the window size(**W**). Then for every pixel in the window, find the corresponding pixel according to the distance(**D**) that is set beforehand. Then, record the number of each found pair.

Figure 1 shows an example of GLCM in 2($0^\circ, 45^\circ$) directions. With **W**= 5; **B**= 5(assume there are only 5 value in the image); **D**= 1;
Figure 2 shows the 4 directions of GLCM.

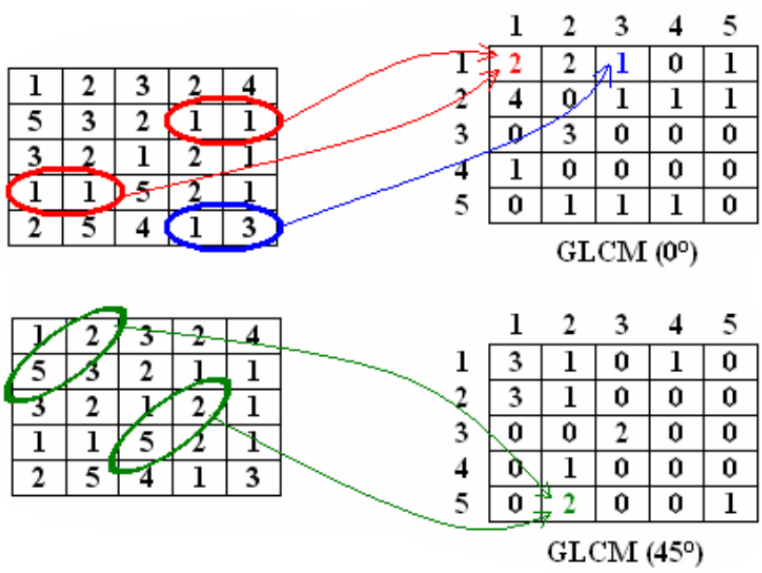


Figure 1: Sample of GLCM[1]

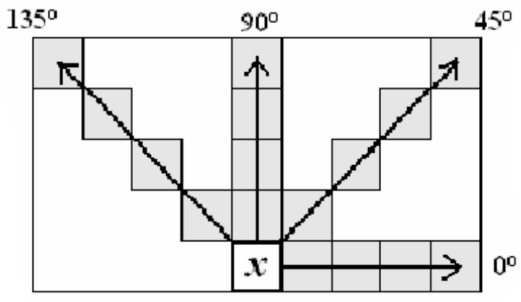


Figure 2: directions of GLCM[1]

Features

After getting the GLCM, some formula is needed to extract the "features". Currently, most of the features got from GLCM is proposed by **Haralick** in 1973[8]. In the paper, he proposed

14 formulas of features (He uses **symmetric** version GLCM, this is also what will be used in the experiment). In the later experiment, those formulas are used to calculate the features. The detail of the formula will not be explained here. For more information, check out the paper for the detail of the formulas.

GLRLM

Gray-Level Run Length Matrix(GLRLM). GLRLM gives the homogeneity of different size of each gray level. This size is the "Run Length" which simply is the size of consecutive pixels appears in the image within a certain window under different concerned directions. Same as GLCM, after deriving GLRLM, further calculations are needed in order to get the true features.

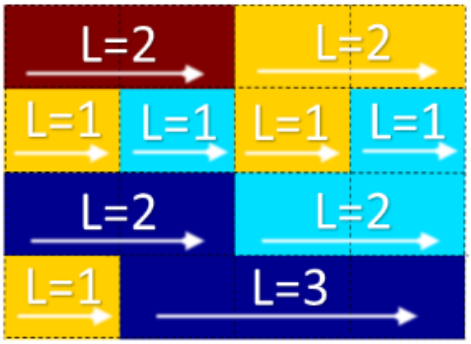
Symbol

Almost same as GLCM.
The gray level that in our image is **Bin size**, denoted as **B**.
The size of certain area is the **Window size**, denoted as **W**.
The angle used to find the pair, denoted as $\boldsymbol{\theta}$. There are four degrees that will be used: $0°, 45°, 90°, 135°$.
But there isn't Distance(D) in GLRLM.

Procedure

- Different from GLCM, the size of GLRLM is $\boldsymbol{B * W}$. The longest length that can be got is the size of the window(**W**). In addition, there is **B** gray level in our image, so the size of the matrix is $\boldsymbol{B * W}$. Each entry of the matrix represents how many sets of certain gray level in certain run length.

Figure 3 shows an example of GLRLM in $0°$. With **W**= 4; **B**= 4; *Notice:* Here the picture doesn't show the last column which represents run length= 4 for convenient. Actually, there should be one more column with all entries equal to 0 as shown in the right side in text format.



	1	2	3
	0	1	0
	3	1	0
	2	1	0
	0	1	1

```
##   1 2 3 4
## 1 0 1 0 0
## 2 3 1 0 0
## 3 2 1 0 0
## 4 0 1 1 0
```

Figure 3: Sample of GLRLM[2]

Features

After getting the GLRLM, some formulas also required to extract the "features". The formulas used are mainly proposed by **Galloway** in 1974[20]. But there still more features proposed after Galloway's work and summarized by **Xiaoou Tang**[21]. In the later experiment,

[1] https://www.researchgate.net/publication/264886592_Computer_Vision-based_Wood_Recognition_System

some of those formulas are used to calculate the features. Also, check out the paper for the detail of the formulas.

Experimental Procedure

In our experiment, 29 features are chosen from those 2 matrices shown in Figure 4.

Figure 4: Features that are being used

The procedure during experiment is as following:

- The parameter of this thesis is **B**= 64; **W**= 15; **D**= 1;
- Since there are four directions, four matrices corresponding to each degree are derived and the features are calculated according to the formulas.
- After that, take the average of the four directions. Then, a single value per features is provided for each pixel in ROI(Region Of Interest). Features of the pixel are not calculated when part of the window, which centered on this pixel, is out of the ROI since it will cover the pixel that doesn't want to include.
- Then, a feature map per pixel in the ROI is acquired, but for further experiments, only one value per features in each ROI is needed, which will make the subsequent experiments easier. In order to get it, the average among all pixels is taken for each feature to obtain a "summary" of features in the ROI.

Machine Learning Model Selection

Since there are lots of traditional machine learning models and parameters can be chosen, the restriction of time makes trying and fine-tuning all of them become impossible. So, at first, some models that are usually of interest are picked. Then, the models of the default setting of the python package **scikit-learn** are tested using 5-fold cross-validation. The result is shown in Figure 5. Models are tested either with or without class weight because our data set is very skewed and therefore the immunity of each model to unbalanced data should be tested. Some good result are marked in the figure. More attention is placed on sensitivity because misdiagnosing patients with advanced fibrosis without fibrosis is more severe than the opposite. In the end, **SVC, MLP and ADA**, these three models are chosen as the models that will be fine tuned later. The reasons are:

- Although the performance of SVC(default with 'rbf' kernel) is significantly bad, the linear kernel SVC performs quite well. So SVC with different kernels is decided to be investigated.

[2] http://joelcarlson.github.io/2015/07/10/radiomics-package/

- RandomForest is similar to Adaboost but with different ensemble method. So the one with higher sensitivity, which is Adaboost, is picked.
- SGDclassifier which is logistic regression is already used in the paper. Although the performance seems remarkable, it is not worth to try the same model again.

Also, the final version of models did not include the class weight because the performance of selected models did not seem to differ too much, and it is easier to implement without class weight.

5 - fold CV				
Default setting classifier	AUC	Sensitivity	Specificity	Accuracy
RandomForset (No Weight)	0.65	67.83%	58.99%	61.31%
RandomForset (With Class Weight)	0.63	62.46%	61.54%	60.00%
SVC(No Weight)	0.62	45.40%	82.91%	72.16%
SVC(With Class Weight)	0.58	54.63%	68.82%	65.77%
LinearSVC(No Weight)	0.69	77.12%	58.11%	62.87%
LinearSVC(With Class Weight)	0.63	80.09%	53.86%	55.45%
MLP(No Weight)	0.65	72.64%	60.43%	63.16%
MLP(With Class Weight)	0.68	74.13%	58.44%	65.29%
ADA(No Weight)	0.6	77.87%	51.86%	59.34%
ADA(With Class Weight)	0.61	58.03%	68.13%	65.68%
SGDClassifier(loss='log') (No Weight)	0.71	70.81%	67.71%	67.98%
SGDClassifier(loss='log') (With Class Weight)	0.62	72.40%	62.90%	60.85%

Figure 5: 5-fold cross-validation of models. SGDClassifier with log loss is equivalent to logistic regression

CNN

Introduction

CNN, which stands for Convolution Neural Network is a class of neural work that becomes hot in recent years because of the huge success in lots of task across from object classification, image segmentation, object tracking etc. CNN was inspired by the connectivity pattern of the neurons in the brain. As the name indicates, the main portion in CNN is convolution operation. The process described as convolution here is just by convention in deep learning. Mathematically, it is a cross-correlation. In computer vision application, convolution is always done between kernel(filter) and the image. The main purpose of these operations are **extracting the features**. The pre-processing required in CNN is much fewer than other traditional methods. With enough training, CNN can learn those filters that were hand-engineered in traditional methods. This main advantage of CNN is that there is no need to extract the matrix and design the formula to extract the features manually. The learnable weights and bias in CNN are a more powerful alternatives than human effort and knowledge in feature design. Figure 6 shows the basic operation of convolution.

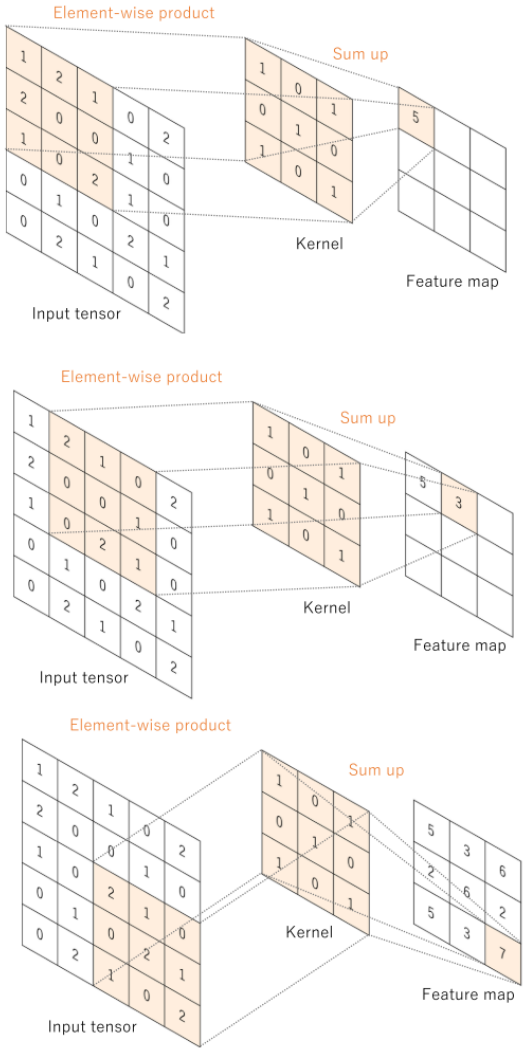


Figure 6: Convolution[22]

In addition to convolutional layer, there are two main kinds of operations also important in CNN which are *Pooling layer* and *Fully connected layer*. Pooling may compute max or average depending on the application. The purpose of the pooling layer is reducing the dimension of the data and might be equivalent to increase the receptive field of neurons in the later layer of the network. Fully connected layers(FC) connect every neuron in previous layer to every neuron in the later layer. It is the same as traditional Multi-Layer Perceptron(MLP). The FC in the CNN is serving as a classifier to classify the features got in the previous convolutional layers(this is just a rough explanation, in recent works, lots of other methods have been proposed which can replace FC and also achieve a great result[1]).

Figure 7 shows the example of max pooling and Figure 8 shows the connection in Fully-connected layer. Figure 9 shows the basic version of CNN architecture.

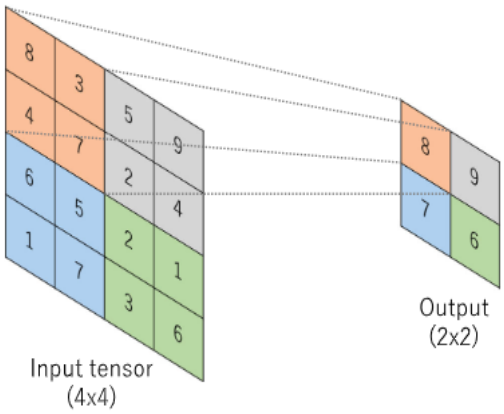


Figure 7: Max Pooling[22]

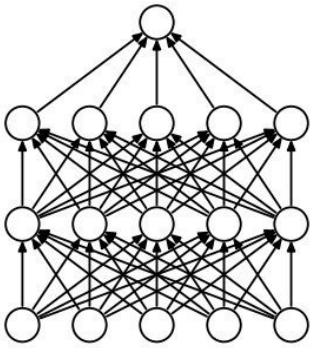


Figure 8: Fully connected[2]

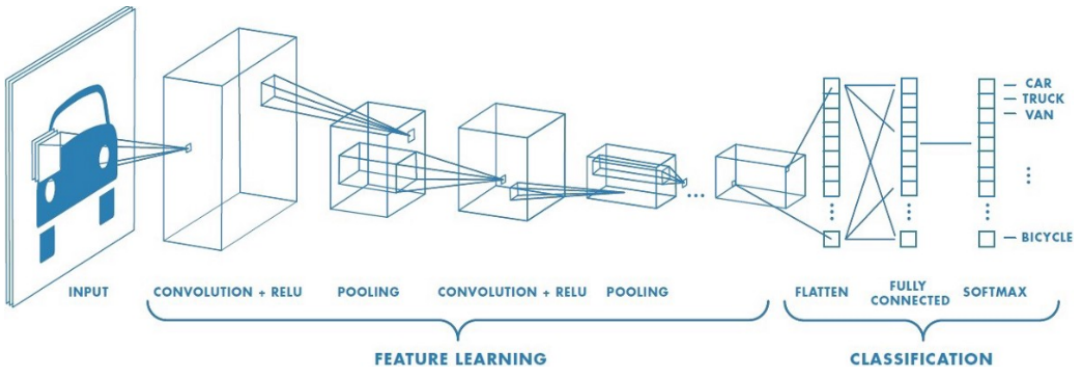


Figure 9: Naive CNN architecture[3]

There are also lots of pieces that are essential to make the network works, such as activation function(RELU and etc) and backpropagation. And there is also some improvement that proposed in recent years including dropout, batch-normalization(BN), 1x1 convolution, depth-wise convolution, and some innovation network architecture. Since CNN is just a tool for the thesis, detail will also not be explained here. Following section will only briefly go through the

network architecture that used in this thesis and mention some main improvement technique.

ResNet

ResNet was proposed Kaiming-He in 2016[23]. The main problem it solved was the degradation problem when a network becomes too deep(too many layers). According to experience, the deeper the network the better it should perform, due to more complex feature extraction the network can conduct. However, the fact is when a network becomes too deep the performance degrade no matter what technique used to prevent over-fitting and gradient explosion/vanish. ResNet architecture uses the shortcut connection (Figure 10) to solve the problem. Due to the shortcut connection, the network will learn *residual* which is easier than learning original features. Furthermore, if the learned residual becomes 0, the network is just doing identity mapping which at least won't hurt the network performance.

In the thesis, ResNet-18 is the architecture mainly used, which only has 18 layers and cannot reflect the true power that shortcut connect want to solve. But since it takes a lot of time to train a deeper network, so the conclusion is to compromise on the shallower network, then using the same setting and training procedure on a deep network.

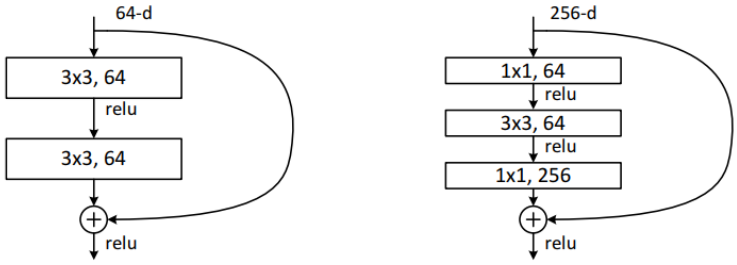


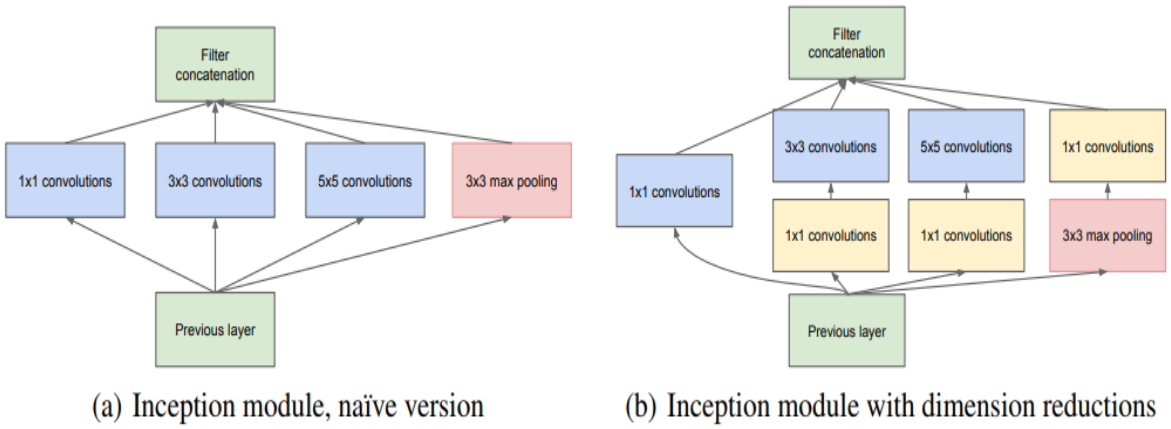
Figure 10: Two kinds of shortcut in ResNet[23]. Left: Shortcut uses in shallow network(18/34 layers).
Right: Shortcut uses in deep network(50/101/152). The number of channels is different and require some strategy to deal with(1x1 convolution or zero-padding).

Inception

Inception network architecture was first proposed by Google in 2014[24]. The main characteristic of Inception network is that it concatenates 1x1, 3x3, 5x5 convolution layer and 3x3 pooling layer in parallel instead of in series (Figure 11). Also, it uses GlobalAveragePooling layer to replace the FullyConnect layer. The goal of these designs is in order to reduce the parameters in the network. With fewer parameters, a deeper network, which can learn more complex features, becomes trainable and available since the hardware resources needed during training is reduced as well as over-fitting extent.

In this thesis, version 3 of Inception network(InceptionV3)[25] is chosen since there is a pre-trained model which can be utilized in Pytorch. The improvement of InceptionV3 is that it introduces "Factorization" as shown in Figure 12. The network factories a big convolution into a small one. For example, 7x7 convolution will be factorized into 1x7 and 7x1 convolution. With this improvement, the network can further reduce the parameters and equivalent to add one more layer which can increase the expression ability of the network.

[1] http://cs231n.github.io/convolutional-networks/
[2] https://cv-tricks.com/cnn/understand-resnet-alexnet-vgg-inception/
[3] https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53



(a) Inception module, naïve version (b) Inception module with dimension reductions

Figure 11: Two kinds of Inception module[24].

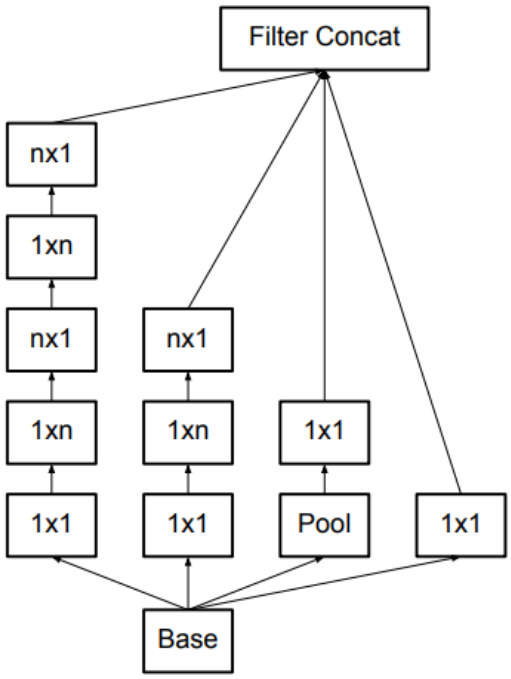


Figure 12: Factorization of InceptionV3[25] All the "Big" nxn convolution is factorized into 1xn and nx1, n depend on the design.

Transfer learning

Transfer learning is a technique that a CNN model training on task A can be transferred to task B. First, the CNN model is training on the dataset for task A which often is very huge and easy to collect. Then, the model will be fine-tuned on the dataset for task B. This technique is useful when the data is hard to collect which is the situation in medical applications.

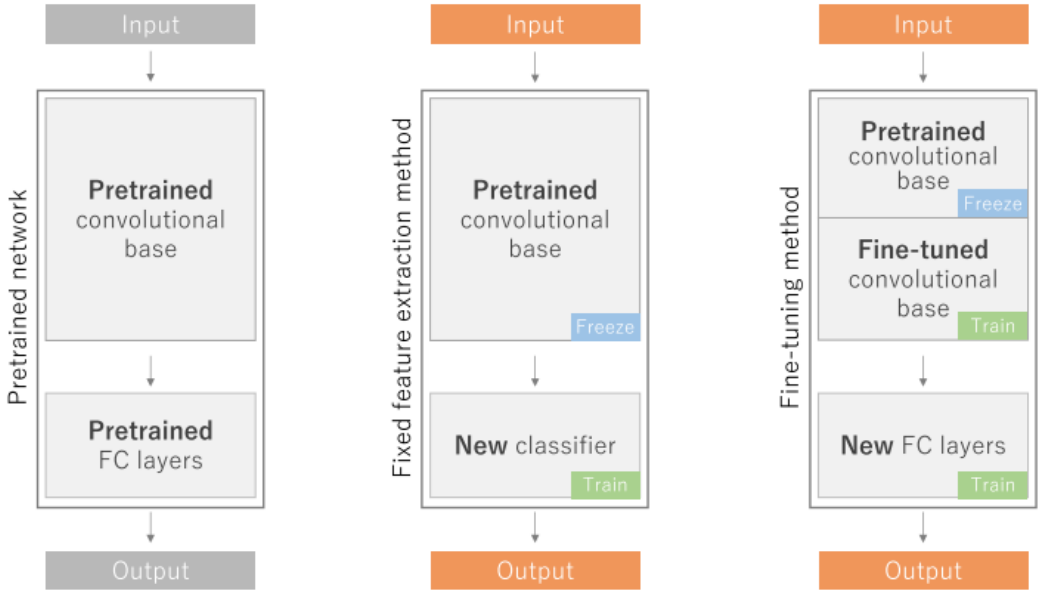


Figure 13: Transfer learning schematic diagram[22] There are two different schema. Middle: We only retrain a new classifier in the network. Right: Apart from the classifier, we also fine-tune the feature extraction methods

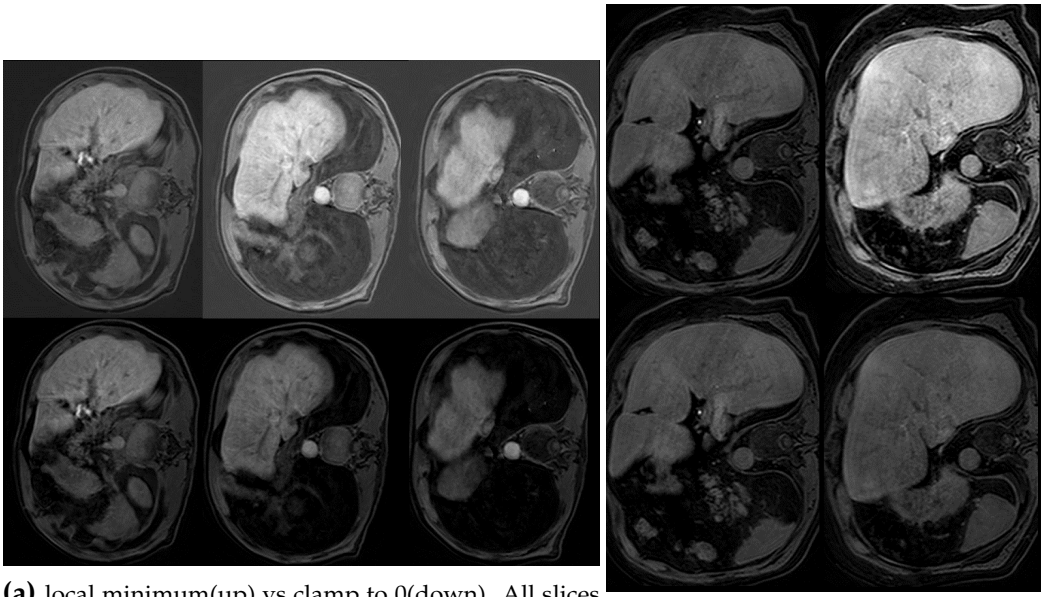
In this thesis, the main modification needed to utilize transfer learning technique is in the final fully connected layer. The output number from the original setting needs to be changed to one since this thesis is a binary classification problem. Also, since the MRI images are quite different from general object images and MRI images are in grayscale, fine-tune the feature extraction methods is also essential. The overall procedure is like the right subfigure in Figure 13

Data Preprocessing

At first, with direct extraction, the value of each pixel in each slice will convert to [0,255] interval according to their local minimum and local maximum since the images are in grayscale. However, the model did not work under this setting(The intermediate result is shown in Appendix D). The reason is that each patient's data is take in a different time with some different processing which is not clear(since doctors won't care about how detail image data was processing). Due to that, the original data distribution varies a lot. Some contains value almost same as [0,255] interval while some contains value in [-200,8000]. Also, each slice has different local minimum and local maximum, and it happens that background is not always the smallest value(since some slice contains pixel value that is minus which the reason is also not clear). In a technical speaking, all slices convert to [0,255] interval with different windowing method and this affects a lot in appearance as shown in Figure 14. To deal with those problems, the minimum value of all the slices are manually clamped to 0(which is supposed to be original background value) and the maximum value is the global maximum of each patient. All slices will later shift to [0,255] interval according to these two values and

following formula 1. So now each slice has an identical background and slices from the same patient has the same maximum value.

$$\begin{aligned} Value_{afterscale} =& \frac{(Value_{origin} - \mathbf{MIN}) \times 255}{\mathbf{MAX} - \mathbf{MIN}} \\ \mathbf{MIN}, \mathbf{MAX} =& \text{The min/max value, determined} \\ & \text{according to the windowing method} \end{aligned} \tag{1}$$



(a) local minimum(up) vs clamp to 0(down). All slices are consecutive slices that come from the same patient

(b) local maximum(up) vs global maximum(down). All slices are consecutive slices that come from the same patient

Figure 14: Appearance of different windowing

Data Augmentation and Training

During training, images will first resize according to which architecture is using(ResNet: 224*224; InceptionV3: 299*299) with batch size 32. Since our data set is pretty skew, samples in each batch are not totally randomly picking but with some weight. The weight of each sample is inversely proportional to the label amount, so the samples belong to the fewer label will more probable to be picked in order to keep label balance in each batch. To further increase our training data and increase generalizability, each slice will randomly process with contrast enhancement/inhibition, brightness enhancement/inhibition, sharpness enhancement/inhibition, blur, flip(45,90,135 degree) and vertical/horizontal transpose or stay the same. The appearance after each processing is shown in Figure 15.

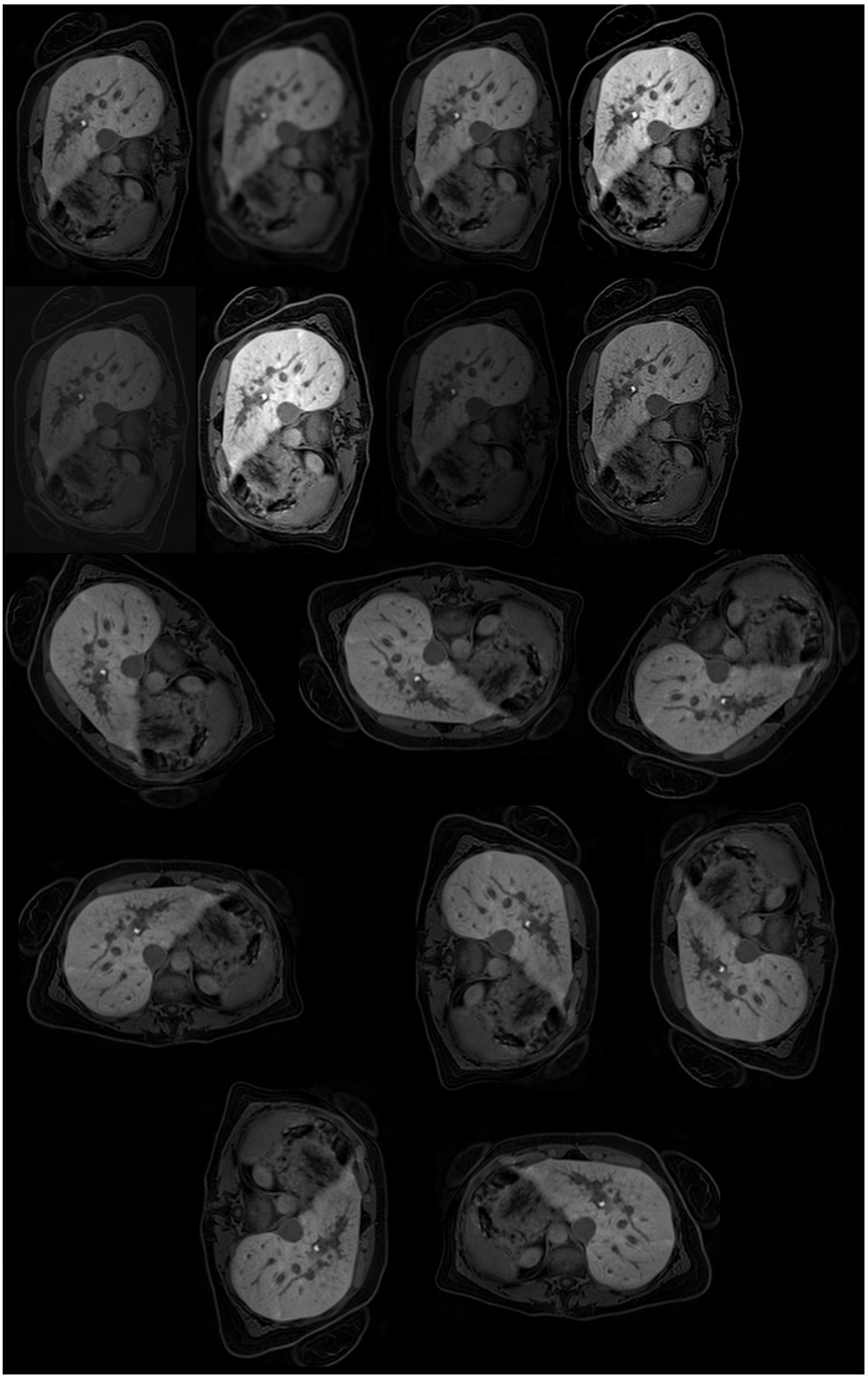


Figure 15: Augment Image. The corresponding augment method from left to right, up to down is:
Origin; GaussianBlur(radius=2); MedianFilter(size=3); Contrast enhance(2)
Contrast enhance(0.5); Brightness enhance(2); Brightness enhance(0.5); Sharpness enhance(5)
Rotate_45; Rotate_90; Rotate_135
TRANSPOSE; FLIP_LEFT_RIGHT; FLIP_TOP_BOTTOM
Rotate_180; Rotate_270

Data set

Dataet-1

The most important part of machine learning and deep learning methods is **data**. We have two data sets one is separate for training and testing(Set1) the other one(Set2) is used to test whether our method is general or not. Set1 is 1.5T MRI T1-weighted image from **PHILIPS** MRI machine with gadoxetic acid-enhanced. It contains 91 patient's data which contains:

- Stage 0: 29
- Stage 1: 16
- Stage 2: 25
- Stage 3: 14
- Stage 4: 7

Since our data set is small, it might be not enough for later use. So, a simple augmentation method is conducted on our data set. Since our image data for each patient contains all the slice of the whole liver (each patient has about 100 slices). In the previous data set, only one slice per patient is taken as train and test data according to the **NLE** data.

NLE stands for Normalized Liver Enhancement. It is a feature that related to liver relative enhancement(image intensity) which also used as an indicator to classify the liver fibrosis stage in [26]. NLE is also used to classify our data set as a baseline(Details are shown in Appendix A). In my data set, NLE is measured by the staff in the hospital and only measure on a certain slice. This is the reason why only certain slices are used before. But in order to increase the data set size, all of the resources in the data set needed to be leveraged. Each slice of each patient is taken, but in case the region of the liver contained in a slice is too small, a liver area threshold is set(threshold= 2500(pixel)) which filter out the slice that contains a too small liver area. The data distribution after augmentation is:

- Stage 0: 2105
- Stage 1: 1105
- Stage 2: 1806
- Stage 3: 954
- Stage 4: 570

Dataet-2

Set2 is a 3.0T MRI T1-weighted image from **PHILIPS** also with gadoxetic acid-enhanced. However, we don't have NLE data for data set2, so we can't analyze the NLE or RFI index in this data set. But this dataset-2 is just aim to test our methods generalizability, it is not a problem that doesn't contain NLE data. It contains 33 patient's data which contains:

- Stage 0: 10
- Stage 1: 7
- Stage 2: 7
- Stage 3: 8
- Stage 4: 1

Same augmentation as set1 is also conducted on this data set and the distribution after augmentation is:

- Stage 0: 778
- Stage 1: 646
- Stage 2: 641
- Stage 3: 680
- Stage 4: 103

Data set for CNN

For the data used to train CNN, each slice is also treated as individual data points just like the above explanation, so enough data is obtained to train a CNN model. The pictures are directly extracted from original slices data also with liver area threshold=2500(pixels).

Results

Reproduce paper result

After all the features have been extracted from each patient, the result index(RFI) can be derived by following the formula proposed by the paper[6]:

$$\begin{aligned} RFI &= \frac{e^a}{1+e^a} \\ a &= -4.3 \\ &\quad + \textbf{NLE} \cdot (-0.24) \\ &\quad + \textbf{GLRLM\_LRLGLE} \cdot (0.93) \\ &\quad + \textbf{GLCM\_MAXP} \cdot (-3.15) \\ &\quad + \textbf{GLCM\_SUME} \cdot 1.21 \end{aligned} \tag{2}$$

Since during the feature extraction, each pixel in the ROI(the ROI of my experiment is the whole liver) has a result RFI, a RFI mapping can be set for each pixel, as shown in Figure 16 (Figure24 shows the original images of these 2 examples)

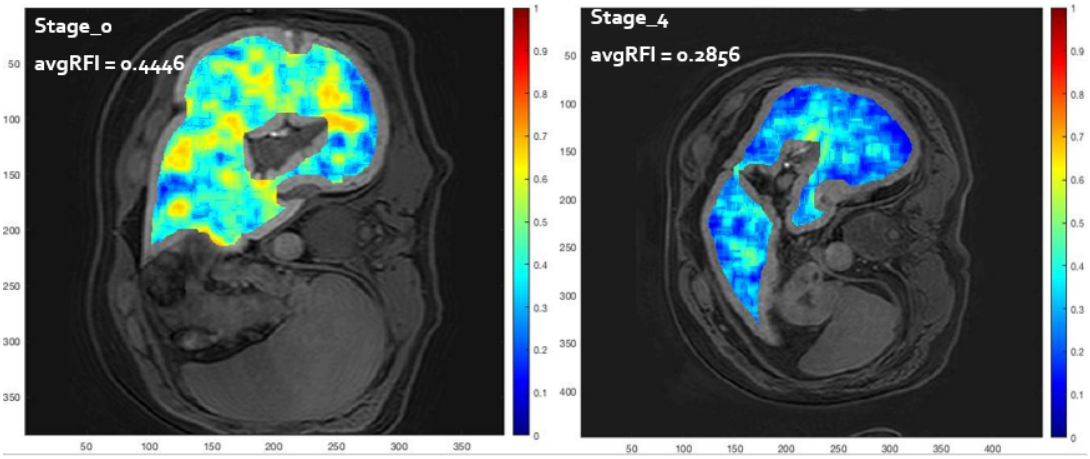
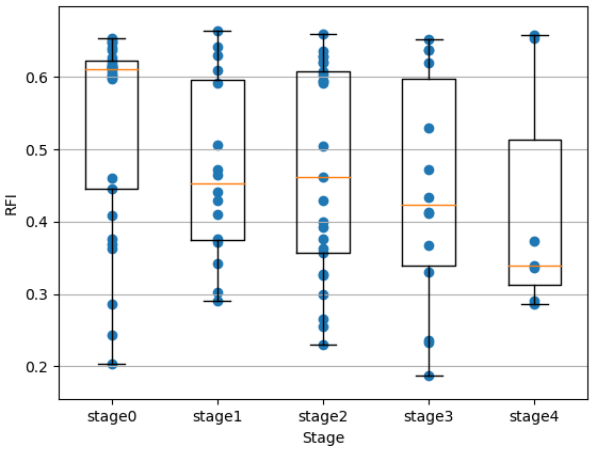
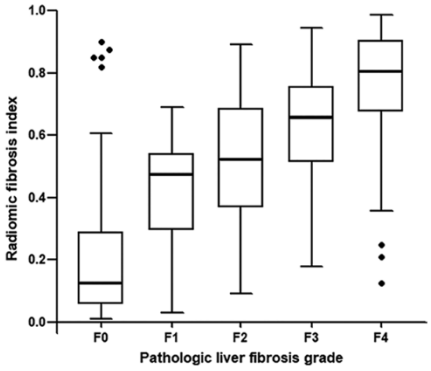


Figure 16: RFI mapping

Similar to the steps described in section Experimental Procedure, the average among the RFI of each pixel is taken to obtain a "summary" of total RFI in the ROI. It is easier to analyze when an image has only one RFI index. The text in Figure 16 shows this averaged RFI of each image. Then, the comparison of the RFI distribution between ours result and the paper is shown in Figure 17.



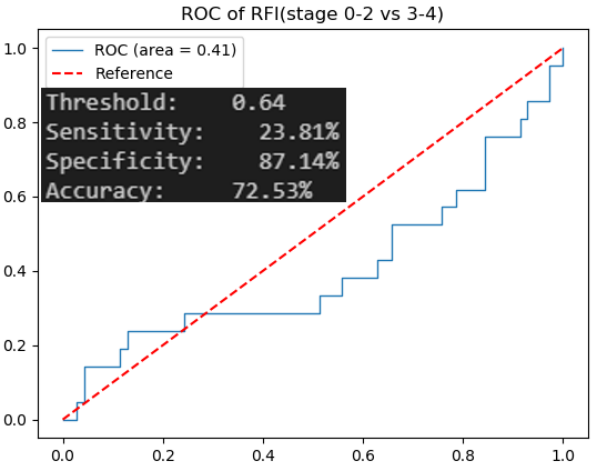
(a) RFI distribution of our result



(b) RFI distribution from paper

Figure 17: RFI distribution comparison

As Figure 17 shows, the resulting distribution of our data set is far from what shown in the paper. Figure 18 shows the performance of our experiment and the result of the paper. Not unexpectedly, our result is much poorer than the paper.



(a) RFI ROC of our result

Advanced fibrosis, stage F3–F4	
AUC	0.88
Threshold*	0.59
Sensitivity (%)	80.3
Specificity (%)	84.0
Accuracy (%)	82.1

(b) RFI AUC from paper

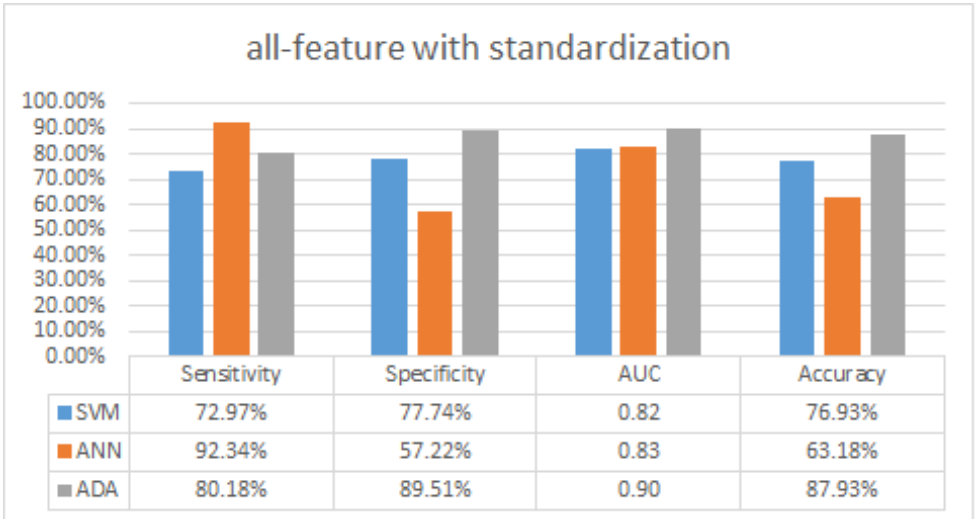
Figure 18: ROC comparison based on RFI

Due to the poor performance that derived by following the paper formula, the next step is finding new models that can work on our data set and evaluate its generalizability on another data set. Also, due to the poor performance of the reproduced result, the baseline of this thesis is turning to the NLE indicator of which the result is shown in Appendix A. The ultimate goal is perhaps to find a method that not only works pretty well on target data set but has high generalizability. First, other conventional machine learning models have been tried.

Notice: In the later section, the data set used transferred to the one that treats each slice from each patient as different samples. Since there is only one NLE data for each patient instead of each slice, in the later experiment only 29 features from 2 matrices, excluding NLE, will be considered as input.

Conventional machine learning method

After following the decision in Machine Learning Model Selection, the models which will be studied in depth are chosen. The grid search is conducted on the hyperparameters of each model. There's also build in function regarding grid search inside **scikit-learn** package to be utilized. The same 5-fold cross-validation method is used on the train set for fine-tuning hyperparameters. Figure 19 shows the final performance on the test set and Figure 20 shows the hyperparameters. The result shows that Adaboost has the best performance with not bad sensitivity and remarkable AUC.



	Sensitivity	Specificity	AUC	Accuracy
SVM	72.97%	77.74%	0.82	76.93%
ANN	92.34%	57.22%	0.83	63.18%
ADA	80.18%	89.51%	0.90	87.93%

Figure 19: Models performance

```
AdaBoostClassifier(n_estimators=55, learning_rate=0.005, algorithm='SAMME.R')
MLPClassifier(solver='adam', alpha=1e-4, hidden_layer_sizes=(25,6), activation='relu',
  early_stopping=True, max_iter=10000,beta_1=0.55,beta_2=0.55,tol=1e-6)
SVC(C=5, kernel='linear', tol=0.0001, probability=True)
```

Figure 20: Models parameter

Then, the generalizability of the Adaboost model is tested on dataset-2. Figure 21 shows the result of ROC. The generalizability of the Adaboost can be observed which is not good.

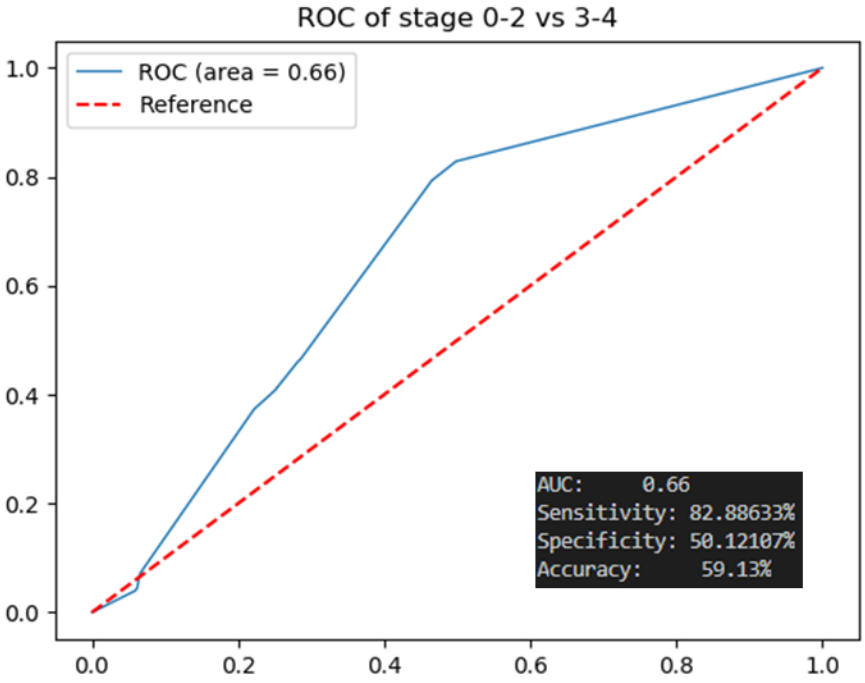


Figure 21: Adaboost generalizability performance on dataset-2

Deep Learning Method

Although the above method reaches a good result on dataset-1, it requires the knowledge about how to extract the features and the implementation. Following the deep learning is adopted which is a hot topic recently and the advantage is it can extract the features automatically.

Following the procedure in Data Preprocessing and Data Augmentation and Training, it ends up that the model using all augment image, **learning_rate = 1e-5 and weight_decay = 1** performs best. Also, the liver area included in the training set also affects the result. The training set includes small liver area will lead to a bad result. The area threshold will be increased to **area_threshold=10000** in the later experiment.

All those fine-tune are done in ResNet-18 which is a more shallow network. After that, the research turned to the deeper networks which theoretically might perform better. Then, the assumption that the training procedure and hyperparameters which experimented on ResNet-18 also worked on deeper networks is made. (All the intermediate result is shown in Appendix D)

Figure 22 shows all the resulting performance including deeper networks and ResNet-18. All deeper networks were training on exactly the same setting as ResNet-18. The generalizability is also tested on all of these deep neural networks. The result is shown in Figure 23. The generalizability of all the models is also poor which merely slightly better than the conventional machine learning model with manual feature extraction.

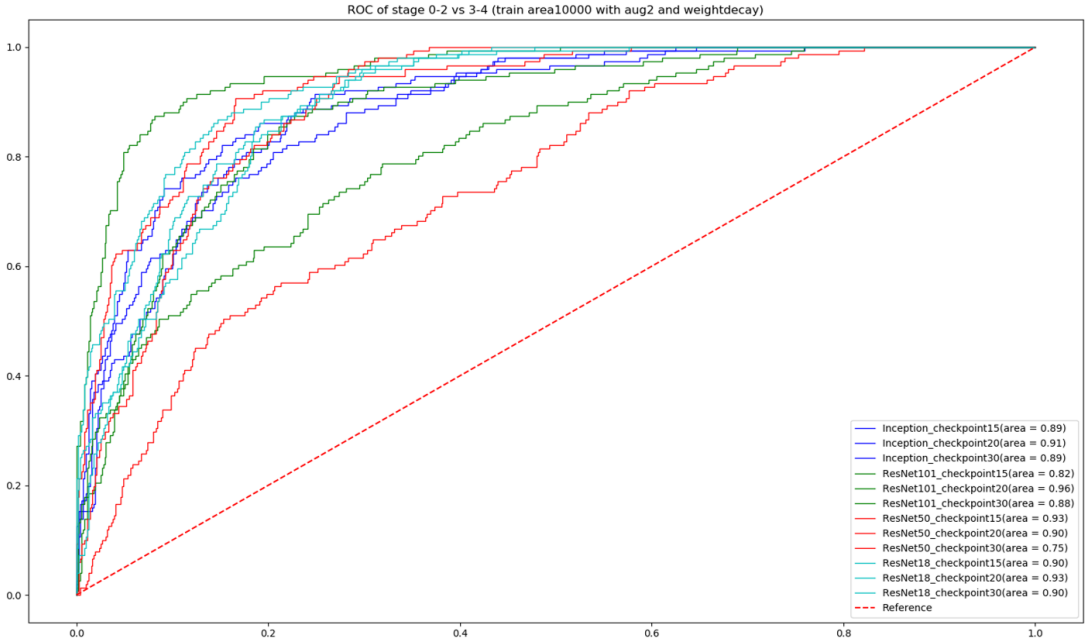


Figure 22: Deep learning result with training set area threshold 10000 on deeper network

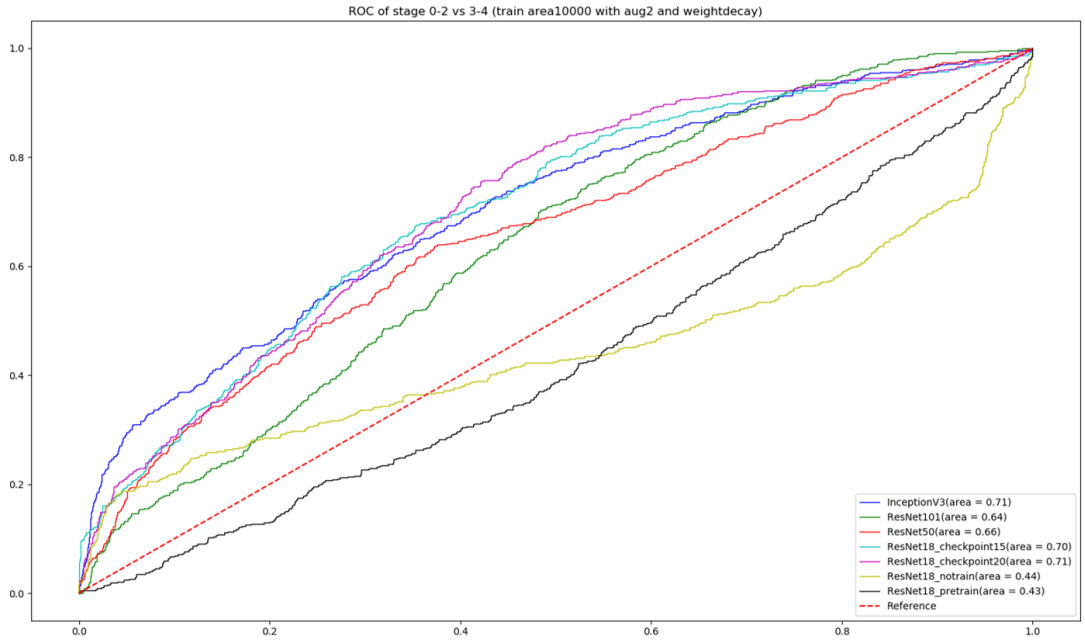


Figure 23: Deep learning generalizability test. *ResNet18_pretrain* means the model that only with pre-trained parameters(on ImageNet) and never train on 1.5T MRI data set

Discussion

Confirmation of published method

As the result shown in Figure 18, our result is far worse than the result from the paper. (Figure 26 also shows the resulting RFI mapping for each pixel from the paper, which can compare with Figure 16) It might be caused by the fact that our data set is different from theirs. For example, our data set is 1.5T MR image but theirs is 3.0T MR image. Also, the different of the machine manufacturer may cause difference between output image and different MRI parameter setting. The patient population distribution is also a potential difference between data set. Overall, there are lots of reason that our data set distribution is different from theirs and the method they purposed is not general enough, so their result is unable to reproduce on our data set. Actually, we can compare the Figure 24 and Figure 25. It is obvious that although the livers are in the same fibrosis stage they show quite different texture appearance between our data set and theirs. This is a common issue in lots of research area, especially, in the field of machine learning. In their paper, they used *logistic regression with elastic net regularization*, which is a kind of learning algorithm. It is common that the algorithm really works on one data distribution but not another. The more reasonable way to take advantage of this paper might be using the same approach they propose to form our own formula. Then, using this formula on our own data set. The result and procedure are shown in Appendix B. But then a model needs to be trained again and again once shifted to new imaging parameters or even just change a machine vendor. This is a pretty miscellaneous and non-general method and also let the user feel unconfident to the model they trained.

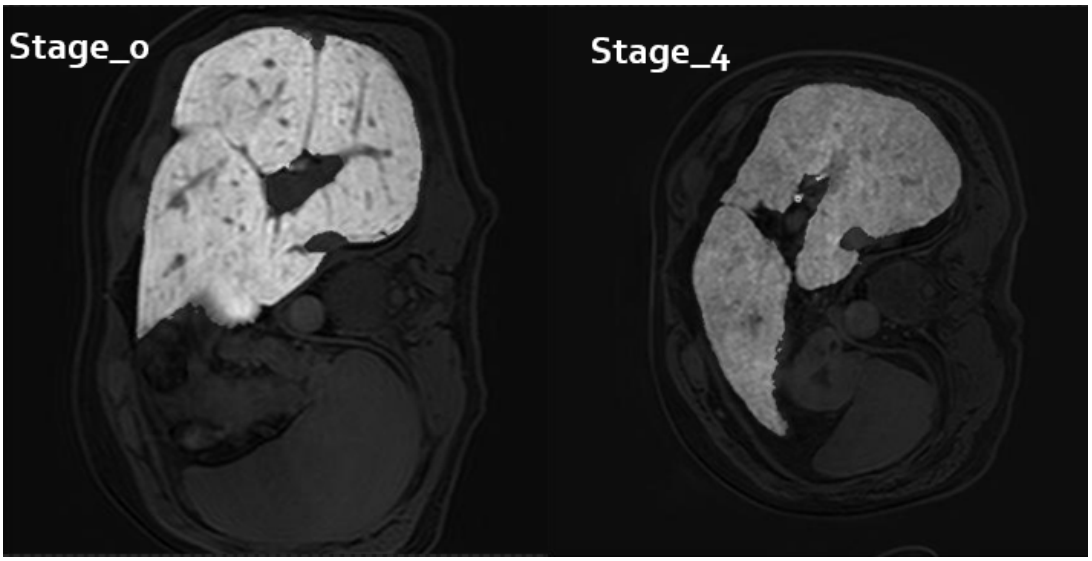


Figure 24: Origin MR image

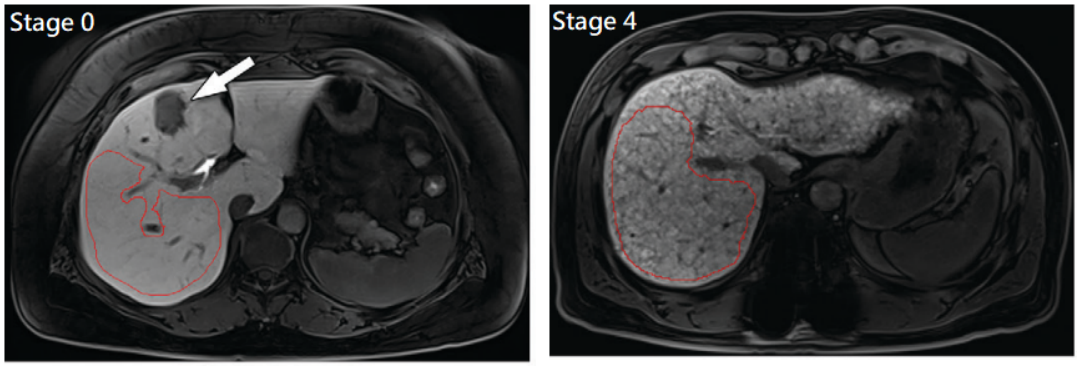


Figure 25: Liver from the paper[6]. Red line indicates their ROI. The arrow in the left image indicate colorectal hepatic metastases, which is not important here

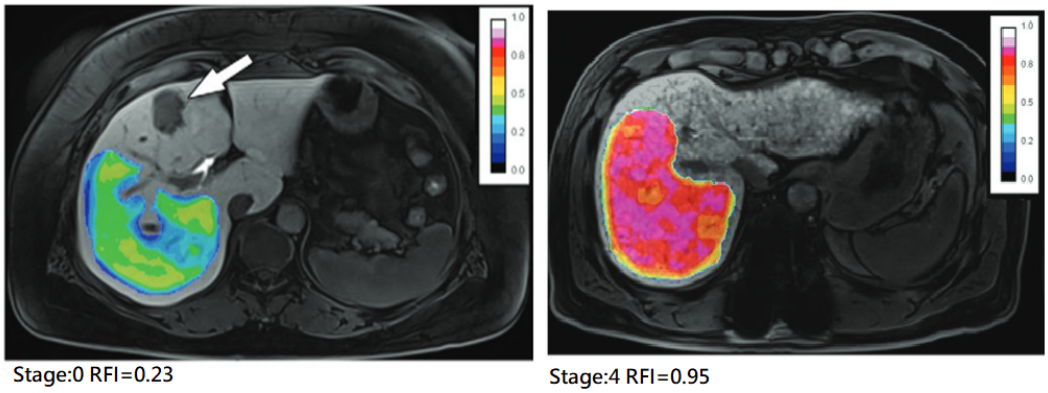


Figure 26: Resulting RFI mapping from the paper

Conventional Machine Learning result

As the result shown in Figure 19, the performance of Adaboost is the best with **AUC=0.9** and outperforms the NLE method with **AUC=0.76** for Stage 0∼2 V.S. Stage 3∼4(which is shown in Appendix A). Some further investigations are conducted on the Adaboost. Figure 27 shows the confusion matrix of Adaboost. Again, in this application, it is more severe to

misclassify the Stage 3∼4 to Stage 0∼2. All those 44 misclassified samples are checked and tried to find out the reason that caused model failure. Figure 28 shows those samples. Most error can be found to occur in patient RH37. Since it is the only patient who has misclassification on almost all the slices, the error may have occurred in the biopsy that is considered as "gold standard". In fact, because the liver fibrosis might not be evenly distributed among the whole liver, there might have biopsy sampling inconsistency which is mentioned in Introduction and also in some papers [27, 28].

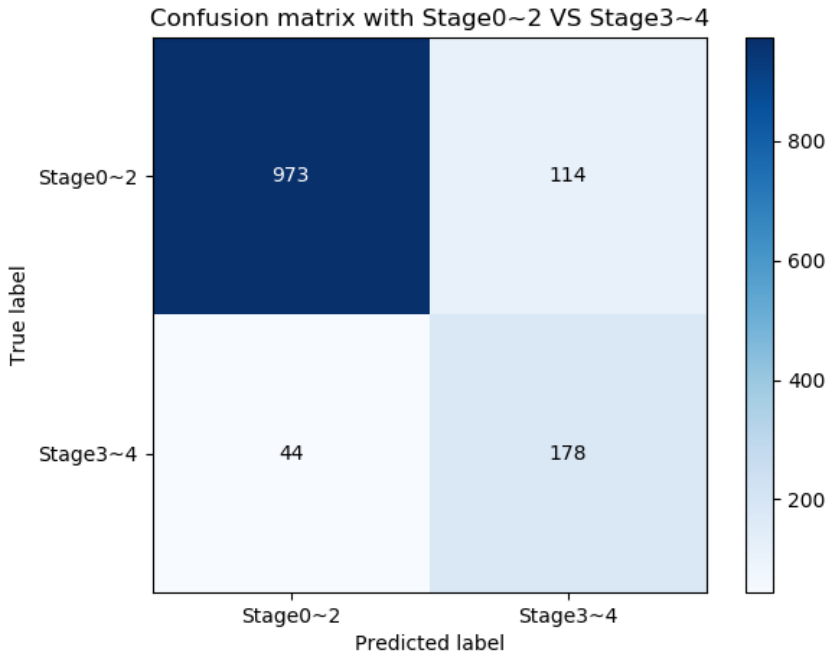


Figure 27: Adaboost Confusion Matrix

PID	STAGE	AREA						
SP55	4	3928	RH37	3	23092	RH37	3	2779
SP55	4	4125	RH37	3	23605	SS48	3	4206
SP55	4	6941	RH37	3	24058	SS48	3	3052
SP55	4	7380	RH37	3	24452			
SP55	4	7888	RH37	3	24794			
SP55	4	8967	RH37	3	25071			
RH37	3	2632	RH37	3	25266			
RH37	3	3218	RH37	3	25381			
RH37	3	4658	RH37	3	25306			
RH37	3	9737	RH37	3	25066			
RH37	3	10874	RH37	3	24641			
RH37	3	11951	RH37	3	24131			
RH37	3	13034	RH37	3	23447			
RH37	3	16906	RH37	3	22536			
RH37	3	17790	RH37	3	21423			
RH37	3	18803	RH37	3	20084			
RH37	3	19659	RH37	3	17080			
RH37	3	21661	RH37	3	15642			
RH37	3	22064	RH37	3	14599			
RH37	3	22572	RH37	3	11560			
			RH37	3	10729			

Figure 28: Misclassified samples

However, the generalizability shown in Figure 21 indicate that the Adaboost model is not fit in dataset-2. This will cause the problem that if someone sees our result and directly gets our model and applies it on their own data set it might fail, especially on 3.0T MRI image. Figure 29 also shows the confusion matrix of the generalizability test. The result shows that the sensitivity seems quite well and specificity is poor which almost half of the negative samples are misclassified. However, the good sensitivity is not very credible, since there are not enough different individual patients in stage 3 or 4 and might introduce some bias to the resulting performance.

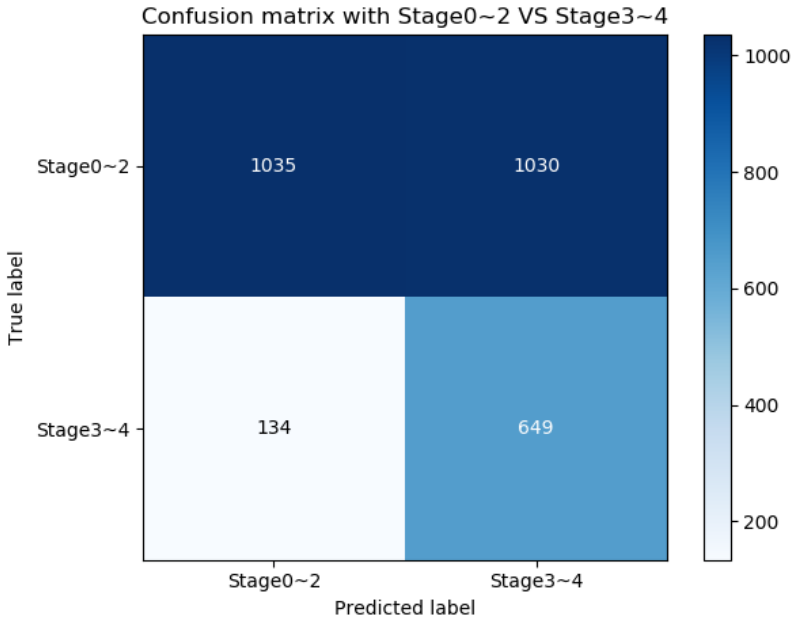


Figure 29: Confusion matrix for Adaboost on dataset-2

Deep Learning result

As the result shown in Figure 22, the best performance is produced from ResNet-101 which achieve **AUC=0.96** - a very impressive number. On the other hand, all the networks were training show a similar over-fitting pattern which happened around 20 epochs (except ResNet-50 might happen in epoch 15 or between 15 and 20). It might be the consequence that all the network were training on exactly the same setting. This is not a big deal but just a clarification that the similar property during training among all the networks might not be just a coincidence. Although the ResNet-101 has best performance on dataset-1, the generalizability shown in Figure 23 is poorer than other architectures. It indicates that the ResNet-101 might be over-fitting on dataset-1, which is not a problem if the concern is only on dataset-1(1.5T MRI).

Besides, there is one big advantage of deep learning methods which is **execution time**. Although, as everyone thinks, deep learning methods do take much longer to train than conventional machine learning methods, the total time deep learning methods cost include data preprocessing is actually shorter in this thesis. The idea is that the manual feature extraction phase takes too much time since features for each pixel are extracted. The situation will definitely be different if another more efficient feature extraction method is adopted, but, at least, there is this advantage on this occasion and might also appear in others. The advantage is much clearer once the training is finished and the time cost only focus on testing. As shown in Figure 30, the execution time of deep learning model is much less than manual feature extraction especially when a slice with a very big liver area is encountered during

testing. If the data preprocessing time is also taken into account, the deep learning methods are not always the most time-consuming methods.

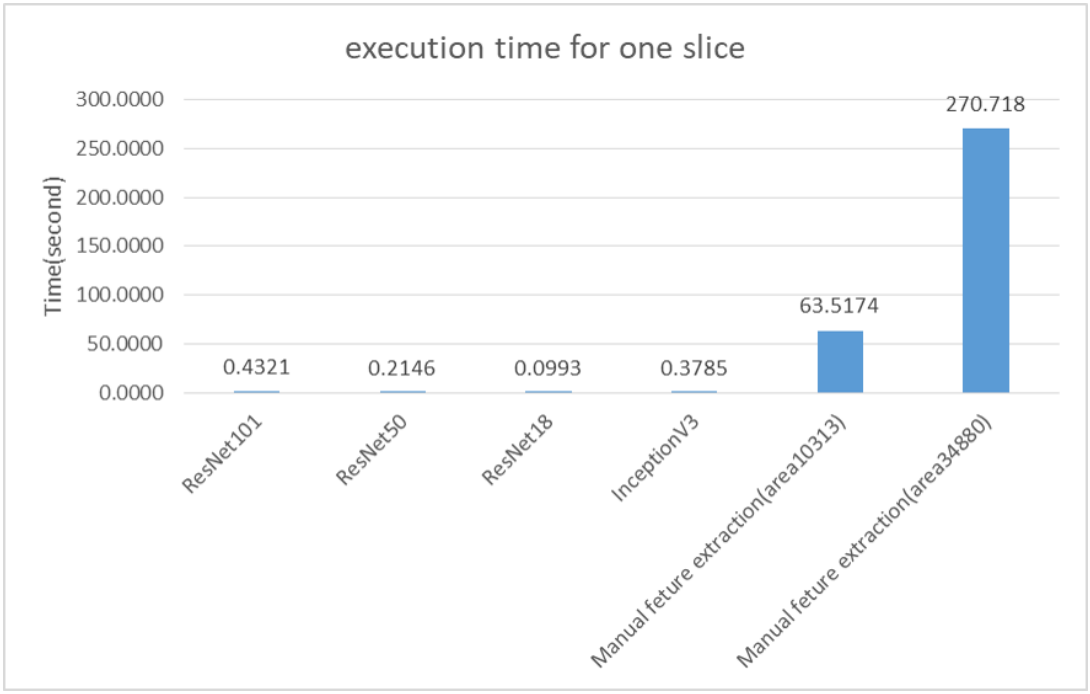


Figure 30: Execution time of each model

Generalizability test

Generalizability is mainly described in this section.

It is difficult to fully trust a new system especially in the medical application in which misdiagnosis is very serious. To check how promising our model is, our model is tested on a totally different data set, but the generalizability of all the models no matter the one proposed by the paper or those used in this thesis is very bad. However, as mentioned in the last sentence in the previous section, this might not be a problem if the focus is only on one kind of data set which is 1.5T MR images in this thesis. Actually, this is the case that will be faced in the common medical applications. One trained model is devoted to only one kind of data set.

In face, conducting a generalizability test on a 3.0T MRI data set with a model trained on 1.5T MRI data set(or opposite) is considered to be not quite fair. It might also not fair to test the paper's formula on our dataset-1, since the paper is trained on 3.0T MRI data, however, ours is 1.5T MRI data set. (The reason why the formula from the paper is not tested on dataset-2(3.0T data set) is that NLE data is not available on dataset-2 as mentioned in Dataet-2.) Actually, the paper has mentioned the generalizability of their formula by comparing the performance of the train set and test set. This is just like a model that just train on ImageNet undoubtedly can't fit into our MRI task, 3.0T and 1.5T MRI should be treated as two different tasks with only limited similarity. All can be done is only what has been done in the thesis - Transfer learning. This is analog to a doctor also need to learn how to read an MR image when he/she never saw before. As shown in Figure 23, the models that train on 1.5T data set still perform way better than *ResNet18_pretrain*(not even trained on dataset-1), which means there is still some similarity between these two data sets. With this similarity, later the model that works on 3.0T can be derived faster and with a smaller data set. To correctly test the credibility of the model, the right way might collect more data from same image category(1.5T or 3.0T MR

images), but from a different manufacturer or from a different time, such as from different years or any time away when the original data set was taken. This might be more reasonable, since the data are from same category with some variance. In this way, although the situation mentioned in Confirmation of published method that performance of the model is still not guaranteed when the imaging parameters are changed, the model performance still can be trusted once the imaging parameters are still consistent(same weighted of MR images) but change to another machine.

Frankly, unlike the paper[6], the purpose of the thesis is not to provide a model or formula that can be easily inserted and adapted (which also failed in [6]) to any data set. Instead of providing formula or model adapted to any data set, a **procedure** is provided that everyone can follow in order to construct their own model to classify their own data set and the result in this thesis shows that the performance is quite promising within the data that comes from the same source. In fact, this will be a more promising method than just provide a fixed formula. Since the data distribution will definitely be quite different among lots of different machines and different MR imaging techniques, to train a model that can fit into all the situations seems impossible. Then, a better way is providing what procedure or which features might work on classifying then let each team train their own model on the data set collected by their own.

Conclusions

In this thesis, in addition to Ultrasonography, some image-based methods are used to stage liver fibrosis. One huge benefit is that the imaging method can examine the whole liver, while biopsy only takes a tiny sample from the liver and might cause misclassification as the result indicated in the Adaboost classifier. The result shows that both traditional methods with manual feature extraction and deep learning methods work better than our baseline. On the other hand, undoubtedly, the decision based on two or more different systems, rather than just one, are more credible. Apart from the other methods based on ultrasound, these methods do not depend on human judgment no matter which models are chosen. This characteristic has both advantage and disadvantage. The advantage is that the fibrosis can be staged automatically and the result can be treated as coming from another aspect which the doctor can take into account and conclude a final diagnosis. But the disadvantage is that the system is working as a black box which has been mentioned before. This problem is more serious in deep learning method, in which the features of the model extraction are not even known. Some techniques such as GRAD-CAM [29] might be helpful for us to *see* which parts the model is *looking at*. Although the intermediate features that the model produce will not be shown, some straight forward interpretation of the model is provided for us.

Future work

In this thesis, a lot of effort is devoted to trying as many models as possible. As far as I am concerned, although the parameters can be varied and the testing method is not quite reasonable, the generalizability of a model can slightly be observed in the results of this thesis. But there still lots of aspect in traditional machine learning model as well as deep learning architecture to improve. Maybe some of the models that have been discarded can reach a nice performance or/and generalizability. Also, as mentioned, more insight into our models is needed in case our system is just a black box. Investigating why our model failed on some samples might let how the model make a decision more understandable. Both of them can help to reveal this black box model and might be helpful to build a more general model. On the other hand, it is not easy to get a liver mask which require expert to label for us. For traditional methods, there is no way to avoid the need for masks. But for deep learning methods, the model should be tried to get rid of the dependence on the mask, which will make the model easier to implement. Actually, preliminary test of the model without mask have been done as shown in Appendix D and the result seems promising. It will be a main

improvement in later works. On the other hand, during the manual feature extraction, there is windowing defect as mentioned in Data Preprocessing. This might not be a big deal since both in RFI or traditional ML methods, the input is the resulting extracted features rather than raw images, just like Deep Learning methods. However, to reexamine the models with adjusting the windowing method might be interesting and the comparison between Deep Learning methods and others might be more equitable.

Overall, there's still lots of space to improve and test. Maybe doctors will still take biopsy as the gold standard in the near future, but these image methods can be used to check the liver fibrosis at a very early stage in a pretty simple procedure, or to become the second reference standard for doctors to check whether the gold standard really works. Since the biopsy can sometimes go wrong, as mentioned. It is definitely useful to have another reference for staging which won't cost a lot to achieve. The ultimate goal is replacing the invasive staging method and like what is mentioned above: The purpose of this thesis is not provide a model or formula in the shelf that everyone just pick and plug in to their data set, instead, this thesis is trying to figure out a procedure or a set of features that might suitable for staging liver fibrosis on any MRI data set with gadoxetic acid-enhanced. The progress of the computer vision technique now is not enough for one model to fit in every data set which might never be possible since there truly is difference between different MR imaging methods. The main thing that has been proven in original machine learning method is that the GLCM/GLRLM features combined with Adaboost model are useful for classifying the liver fibrosis. On the other hands, the deep learning method result shows that transfer learning is really helping and the model converges pretty fast(with only 20 epochs of training). Also, it shows that with proper data augmentation, it seems that many individual patient data is not necessary to achieve a not bad result. In this thesis, only total 73 patients' data are included in training phase which are quite small compared with other papers. ([6] has 329 patients for training; [11] has 534 patients; [12] has 186 patients(but 396 CT examinations)) However, frankly, the demand for more data is still necessary. Even the result of this thesis shows that too much data is not needed for training, more data is still needed for testing. Currently, the amount of the data set used for testing is too few which will essentially lower the credibility of our model. Also, more data is necessary to train a model that can classify each stage from others. The resulting models in this thesis are only dedicated to classify Stage 3$\sim$4 from others, but to have a model that can determine the exact fibrosis stage is necessary for real medical usage, and also truly be a powerful tool that can discover liver fibrosis in an early stage. In medical application, it is hard to obtain more data, so a better way is to cooperate with other hospitals. With sharing data and knowledge, it will become easier to develop a useful model. This thesis is only an early work on developing a liver fibrosis classifiers based on MR images, and more work is required to be done to develop a truly usable system in the future.

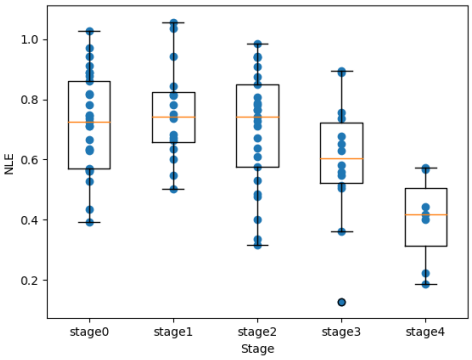
Contact

All the code of this thesis is shown in `https://github.com/eugeneALU` and some information and detail of the code will be included. But the data is not available right now due to the patient privacy issue. If have any question regarding the work, you can leave an issue on github or send an email to `eugenelu49@yahoo.com.tw`.

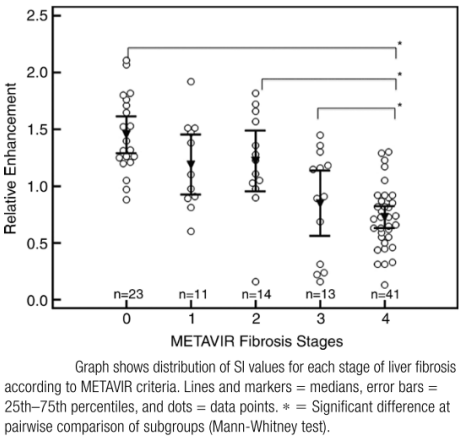
Appendix A

This section shows the performance of NLE index on our dataset-1.

In Figure 31, the distribution of the NLE from our data set and relative SI from paper[26] can be compared. Although the value range is different, the distribution is similar, which let me believe the NLE can work as an indicator in our data set. (The markers are different in two images. Figure 31a using box figure; markers' meaning in Figure 31b show in image)



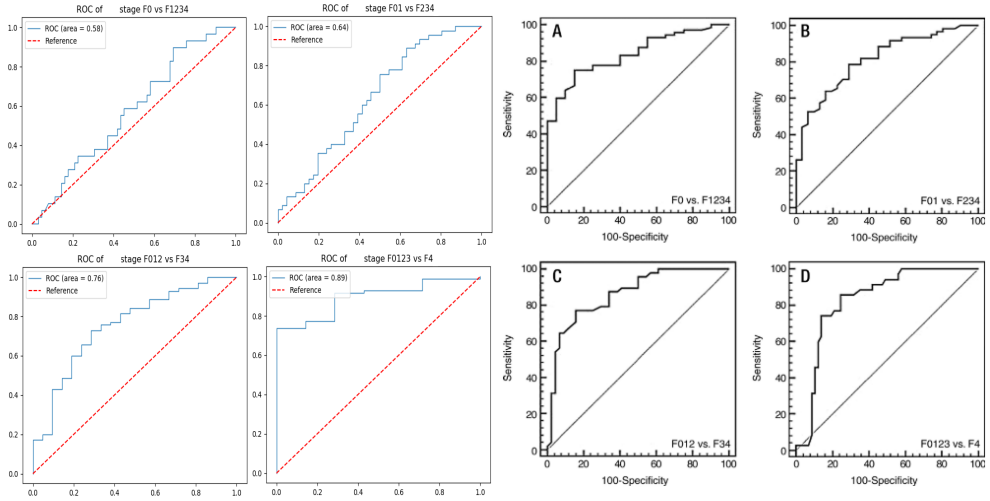
(a) NLE distribution of our original data set



(b) NLE distribution from paper

Figure 31: NLE distribution comparison

The images used in paper is 3.0T MRI images with gadoxetic acid which are different from ours dataset-1(1.5T), but the method is still working. In Figure 33 32 shows the result and ROC of our experiment and paper. Since our experiment focuses on classifying Stage 0∼2 V.S. Stage 3∼4, only that result will be considered later. Because of the small difference between two results(ROC: 0.76 V.S. 0.85), that result are taken as our baseline in this thesis. (One minor thing is that there are only a few data points in stage 4, so the perfect performance of classifying Stage 4 shown in Figure 33a is not so worth believing.)



(a) ROC

(b) ROC from paper

Figure 32: NLE ROC comparison

NLE	0 VS 1-4	0-1 VS 2-4	0-2 VS 3-4	0-3 VS 4
Threshold	0.56088	0.56088	0.62939	0.57675
Sensitivity	30.65%(19/62)	36.96%(17/46)	71.43%(15/21)	100%(7/7)
Specificity	89.66%(26/29)	88.89%(40/45)	72.86%(51/70)	73.81%(62/84)
AUC	0.58	0.64	0.76	0.89
Accuracy	49.45%	62.64%	72.53%	75.82%

(a) result table

Diagnostic Indicator	Fibrosis Stage ≥ F1	Fibrosis Stage ≥ F2	Fibrosis Stage ≥ F3	Fibrosis Stage F4
Relative enhancement cutoff value	≤1.27	≤1.18	≤1.12	≤0.93
Sensitivity (%)	70 (57/81)	75 (51/68)	73 (41/56)	83 (34/41)
Specificity (%)	85 (17/20)	77 (23/30)	87 (33/38)	80 (40/50)
AUC	0.81	0.82	0.85	0.83

(b) result table from paper

Figure 33: NLE result table comparison

Appendix B

This section shows how our own version of RFI constructed and the result.
First, the univariate analysis is omitted. There are two reasons.

- The first reason is that extracting features from two consecutive image sections is needed to get the concordance correlation coefficient before the univariate analysis, which requires drawing ROI manually. The procedure is time consuming.
- Second reason is that to do the univariate analysis first might eliminate some relationship between the features and cause the mistake in the later multivariate regression[1]. The features can still be selected in the later procedure.

Then, Kendall correlation coefficient test is conducted. Figure 34 shows the result of the Kendall correlation coefficient. According to the paper, features with coefficient < 0.2 will be eliminate. However, since negative Kendall correlation coefficient also represent that two variables are related but just negative correlated. In the end, only the features with **absolute** value < 0.2 are eliminated. Also, NLE is not chosen as a feature, because in the later experiment the model is trained on the augmented data set and there is no NLE data for each slice.

[1] https://stats.stackexchange.com/questions/239576/danger-of-univariate-analysis-before-multiple-regression

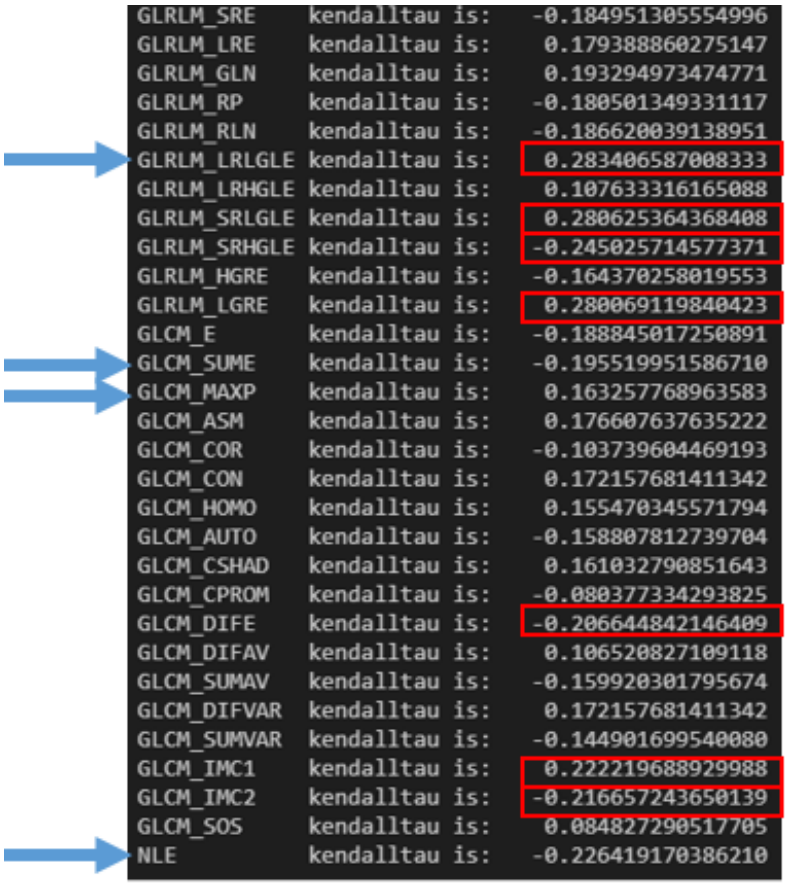


Figure 34: Kendall correlation coefficient. The blue arrow indicates the features used in the paper[6]. The red square indicates the features used in our experiment

Later, those 7 features are used in modeling. The modeling is performed by logistic regression with elastic net regularization, which also used in the paper. However, the paper do not include the hyperparameters they used, so our own hyperparameters are used. Equation 3 shows our own version of RFI(can compared with Equation 2). Figure 35 shows the result of our RFI. The result is almost as good as the paper which is shown in Figure 18b. One thing need to emphasize is that although the performance including sensitivity and specificity don't change significant, but the coefficients in the Equation 3 will slightly shift(shift about $\pm$ 0.1, only coefficient of **GLRLM_SRLGLE** is always 0) in different round of modeling because there is some randomness in our training phase. The reason may be that the model can find lots of different ways to describe our data set. Since our data set is not very big, our data point might be pretty sparse in the feature space, so there's space for our mode to slightly move its classifying criteria.

$$
\begin{aligned}
RFI =& \frac{e^a}{1+e^a} \\
a =& -0.38177752 \\
&+ \textbf{GLRLM\_LRLGLE} \cdot (-0.3240055) \\
&+ \textbf{GLRLM\_LGRE} \cdot (0.5761503) \\
&+ \textbf{GLRLM\_SRHGLE} \cdot (-0.9915529) \\
&+ \textbf{GLCM\_IMC1} \cdot 1.88357532 \\
&+ \textbf{GLCM\_IMC2} \cdot 1.07481857 \\
&+ \textbf{GLCM\_DIFE} \cdot 0.91877409
\end{aligned} \tag{3}
$$

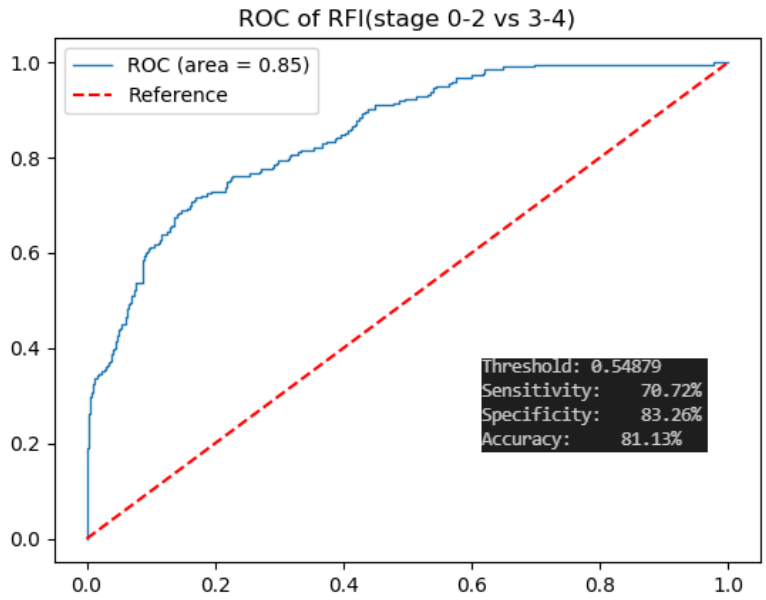


Figure 35: result of our own version of RFI

Although the result seems good, the big different of the two equation show us that this method might not be general. For different data sets, remodeling is needed, although the results after remodeling seem to be satisfactory.

Appendix C

Since the RFI performance is poorer than the paper, an email is sent to the author and ask them some detail about the formula. After several weeks, feedback from the author claimed that some error in the formula has been made by them. The new formula is(compare to original equation 2):

$$\begin{aligned} RFI =& \frac{e^a}{1+e^a} \\ a =& -52.399 \\ &+ \textbf{NLE} \cdot (-2.055) \\ &+ \textbf{GLRLM\_LRLGLE} \cdot (-11.372) \\ &+ \textbf{GLCM\_MAXP} \cdot (52.933) \\ &+ \textbf{GLCM\_SUME} \cdot 13.463 \end{aligned} \tag{4}$$

Then, this formula is tested on our data set again. Figure 36 shows the new distribution and Figure 37 shows the ROC. The distribution is quite different from previous and from paper(Figure 17). However, the performance is very similar. In addition to the threshold, which is highly related to the distribution, others such as AUC, sensitivity, and specificity is same as the previous(Figure 18). The mistake they made is not clear, but the same resulting performance is the evidence that they still stick on the same procedure and data set. The formula contains the same features but with different parameters, which might due to the randomness during training. The tool they used to solve the logistic regression is not clear, but some of the tools include some randomness(what used in Appendix B actually has randomness). As far as I am concerned, the difference in parameters is caused by this randomness, but the outline of the formula is preserved. And this is also why both formulas make almost the same decision on our data set. Because the difference is due to the randomness during training, the overall architecture of the resulting model remains the same. Although the distribution is truly different due to different parameters, the "distribution of decision" remains identical. On the other hand, the result again shows that the method is not general enough, which is consistent with the previous result.

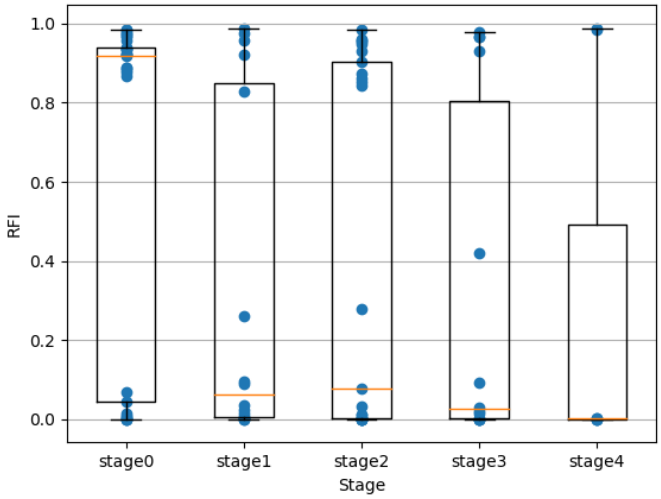


Figure 36: RFI distribution of new formula

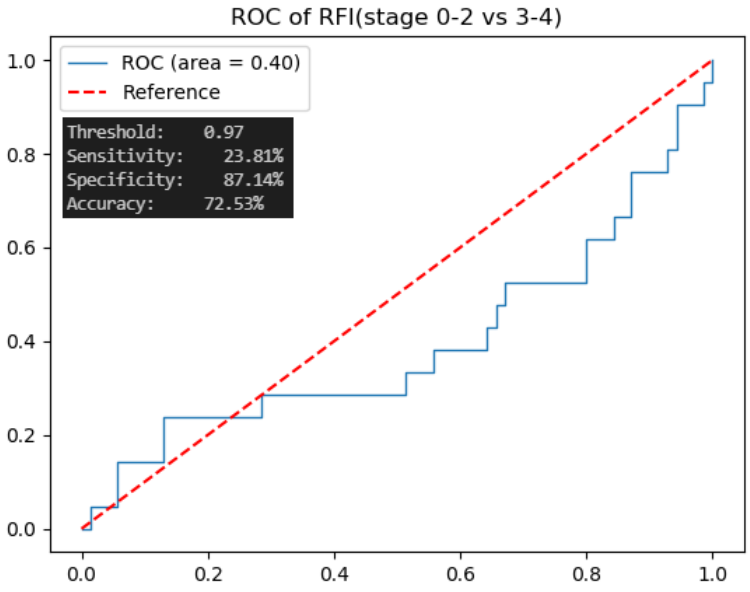


Figure 37: ROC of new formula

Appendix D

This section shows all the detail train/valid result during fine-tuning both before and after using proper windowing method, also, showing how the hyperparameters are decided. Right before fine-tuning, out training set needs to be further divided into train and valid set as shown in Figure 38. The distribution of two sets is(the number in parentheses indicates the number of patients):

- Train
  - Stage 0: 1399 (19)
  - Stage 1: 446 (7)
  - Stage 2: 1390 (19)
  - Stage 3: 688 (10)
  - Stage 4: 240 (3)
- Validate
  - Stage 0: 213 (3)
  - Stage 1: 349 (5)
  - Stage 2: 132 (2)
  - Stage 3: 123 (2)
  - Stage 4: 251 (3)

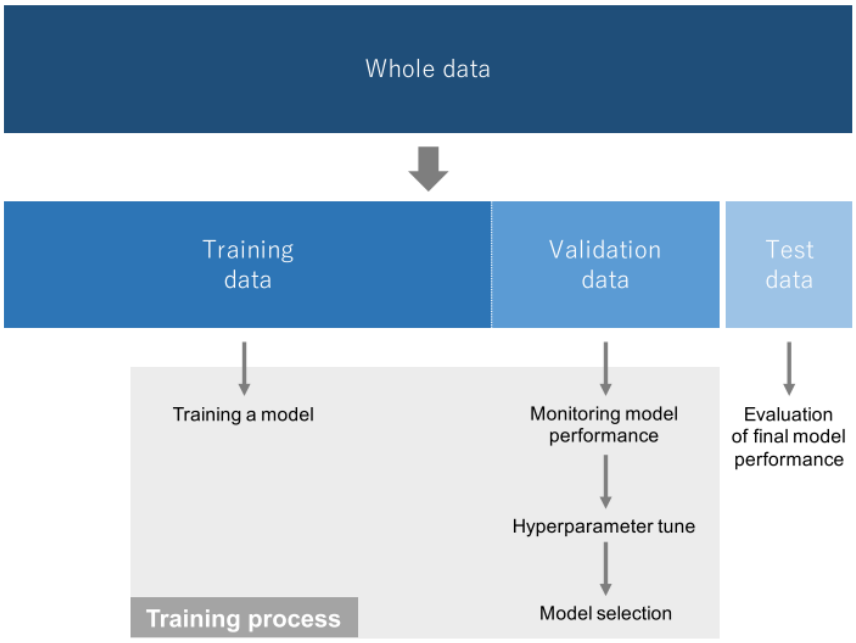


Figure 38: Data split. Later, after fine-tuning, both of Training data and Validation data are used to retrain a model with the previously obtained hyperparameters and the final testing is conducted on the Test data set.

Figure 394041 show the original train/valid loss when the windowing method have not been adjusted. The valid loss is totally not decreasing but goes to a very big value(note the y-axis scale in the valid loss). Lots of different parameters are tested including batch size, learning rate, weight decay, and image augmentation. Different architectures are also tried. The test is also performed on a very shallow CNN, because in the beginning, the problem was considered to be over-fitting of the model immediately after the start of training, which could be solved by using a pretty simple architecture. However, with more and more experiments, training loss even did not decrease sometimes. This is the trigger that makes me go back to check the data set which started to be considered not learnable.

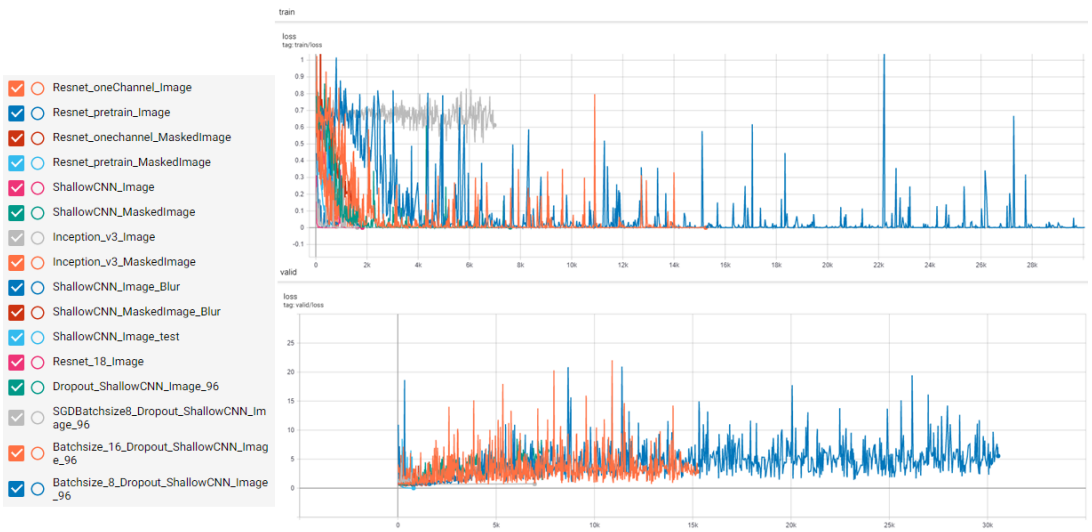


Figure 39: Bad train/valid loss 1

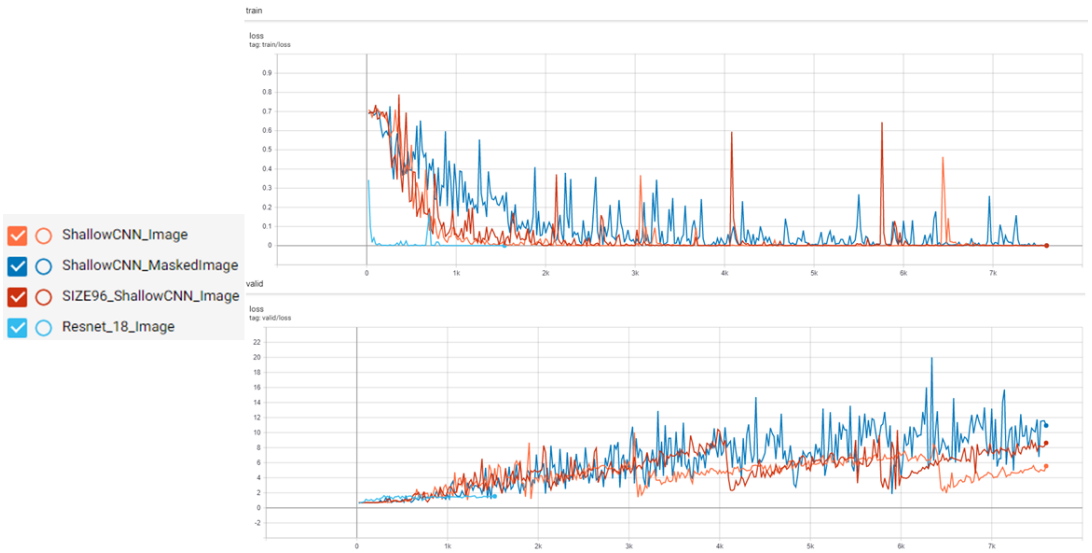


Figure 40: Bad train/valid loss 2

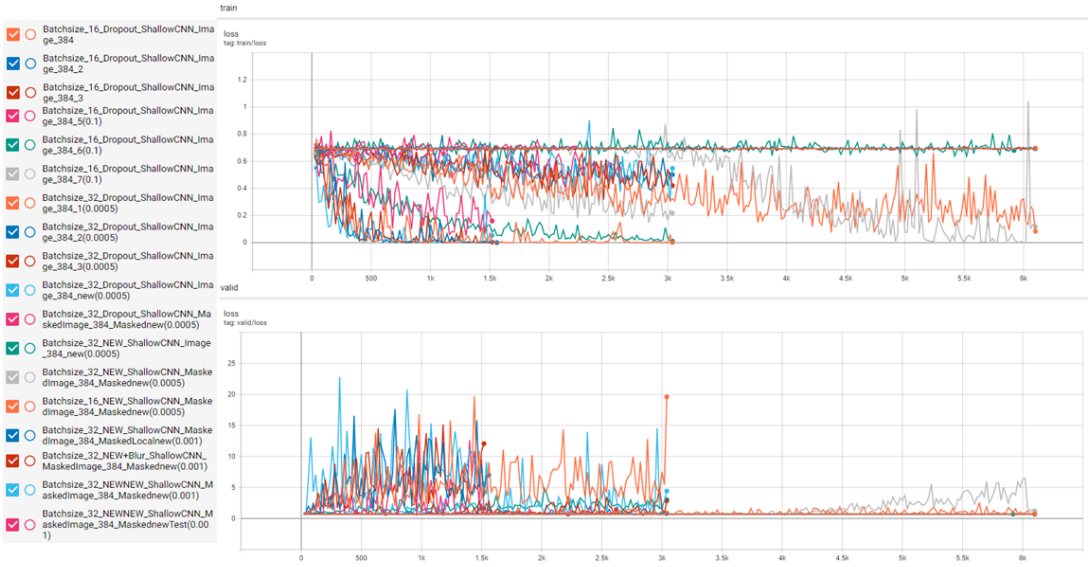


Figure 41: Bad train/valid loss 3

After changing the windowing method, the situation becomes way better as the result shown in the following figures. The following figures show the process and result of fine-tuning.

Figure 42 shows the validation loss and accuracy when training and validating are conducted on the masked image but with different area threshold. A result can be observed that with a bigger area threshold better accuracy can be acquired and also the loss is more stable and smaller.

Figure 42: Validation loss/accuracy with different area threshold

Figure 43 shows the validation loss and accuracy with different windowing method. The two methods considered here are one with the global maximum as MAX and another one with the local maximum as MAX. The different appearance between them is shown in Figure 14b. The result with the global maximum performs better.

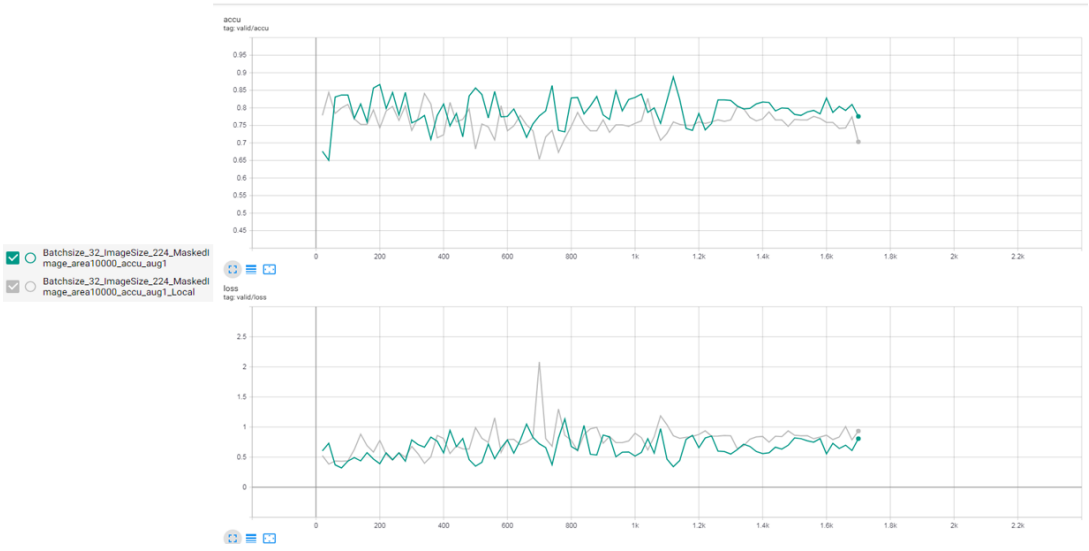


Figure 43: Validation loss/accuracy with different windowing methods

Figure 44 shows the validation loss and accuracy with different learning rate and image augmentation. First, it is obvious that with a smaller learning rate, the loss is much more stable and the accuracy is also better. The validation loss with smaller learning rate decrease in the first few steps and increase later can be observed, which shows the over-fitting really happened and happened very early during training. Second, different augmentations are tried. *aug1* means image augmentation only with the appearance adjust but without any rotation and transposition. The appearance is shown in first 8 sub-figures in Figure 15. On the other hand, *aug2* includes all image augmentation shown in Figure 15. The idea is that every patient actually entered the MRI machine with same body posture, so the cases with rotation and transposition won't happen. However, the result shows here is that the performance is better

with *aug2* though the difference is not very obvious. The reason might be that with rotation and transposition, the model can learn more features which are position irrelevant and can generalize better in the images it never sees before.

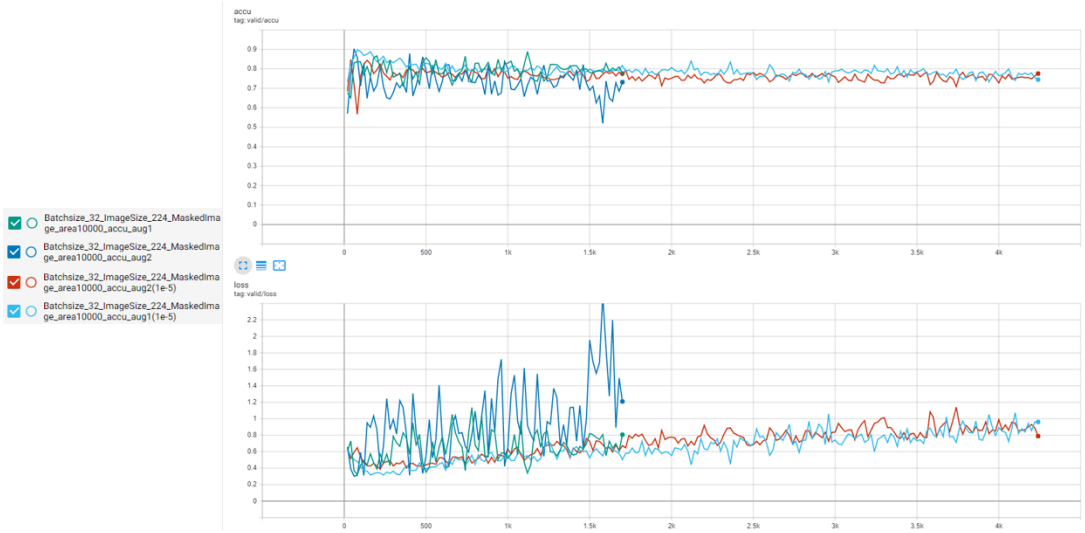


Figure 44: Validation loss/accuracy with different learning rate and augmentation

Figure 45 shows the validation loss and accuracy with raw images or images with liver mask(There are two curves with the same color, so an extra mark is drawn). Training with raw images performs poorer than with liver masks. Actually, it is not surprising since an extra helper is used that helps our model to concentrate on the right area - the liver, that the model is expected to focus on. Since in traditional machine learning method, only features in the liver are extracted, so it is also reasonable to include liver masks in deep learning methods.

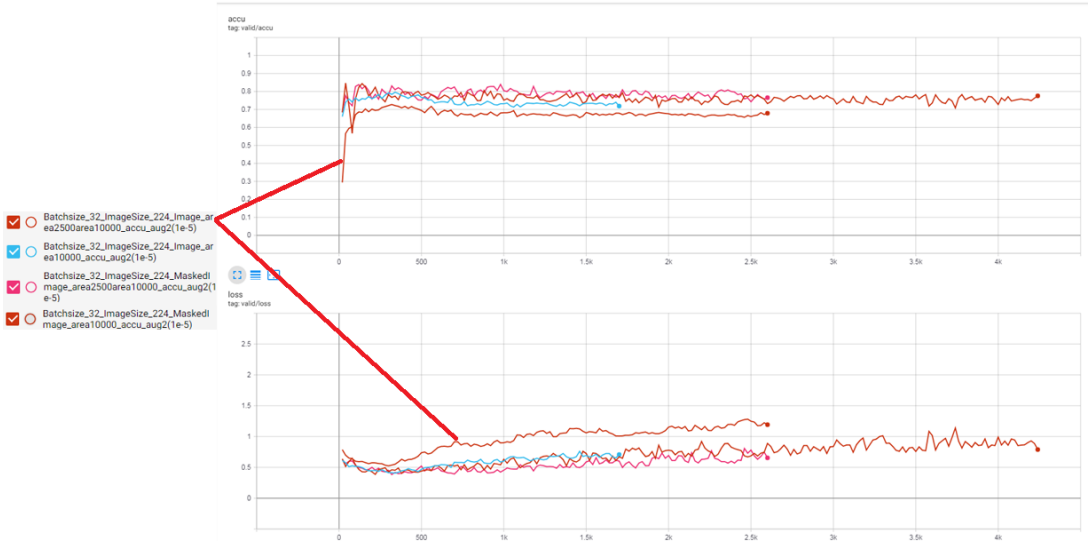


Figure 45: Validation loss/accuracy with raw images or masked image

Figure 46 shows the validation loss and accuracy with different weight decay. A smoothing method is performed on validation loss, so that the over-fitting situation can be observe more easily in different setting. It can be observed that with bigger weight decay (the case -

weightdecay(1)), the lowest point of loss happens later than others which means over-fitting occurred later than others. The final result is also better than the other two setting.

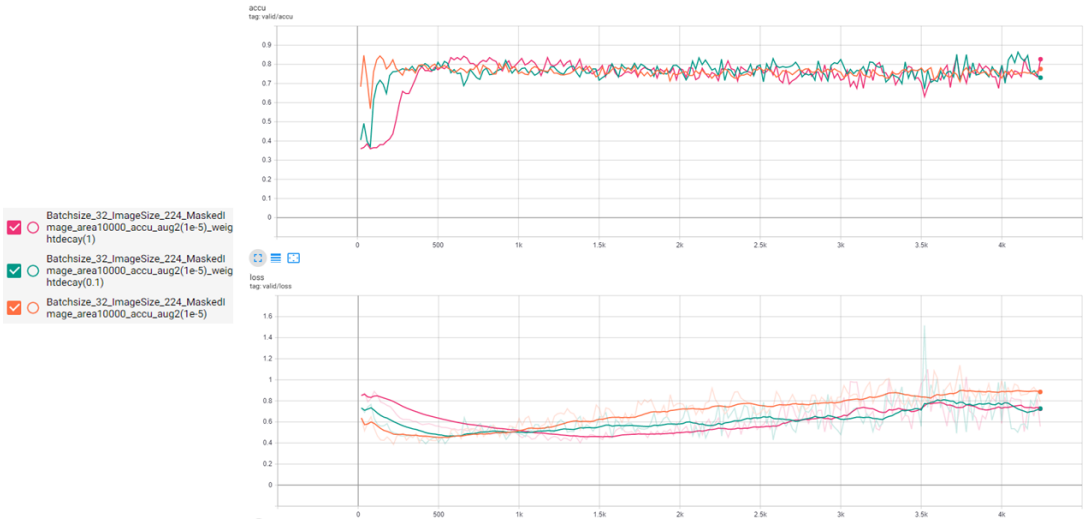


Figure 46: Validation loss/accuracy with different weight decay setting

Figure 474849 shows the classification performance conduct on the validation set using ResNet-18 with different setting and area threshold. The performance is consist with above analysis:

- The bigger the area threshold, the better the model performance.
- With weight decay and proper early stop(equivalent to the checkpoint), the performance will be better.
- To include rotation and transposition in image augmentation(aug 2) really lead to a better result, but just slightly better.

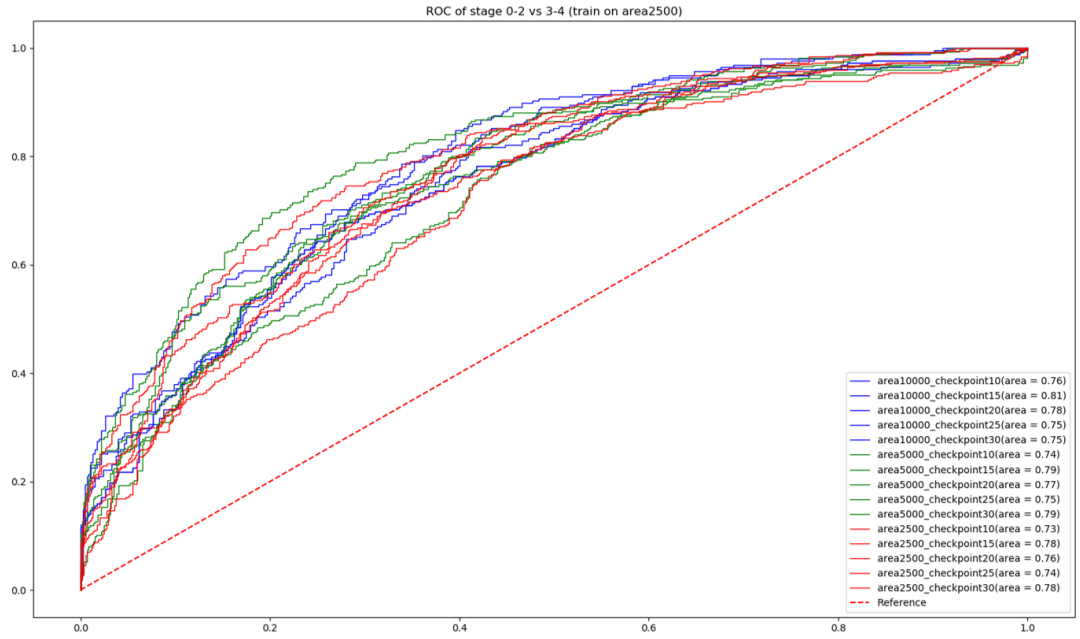


Figure 47: Validation performance with area threshold 2500

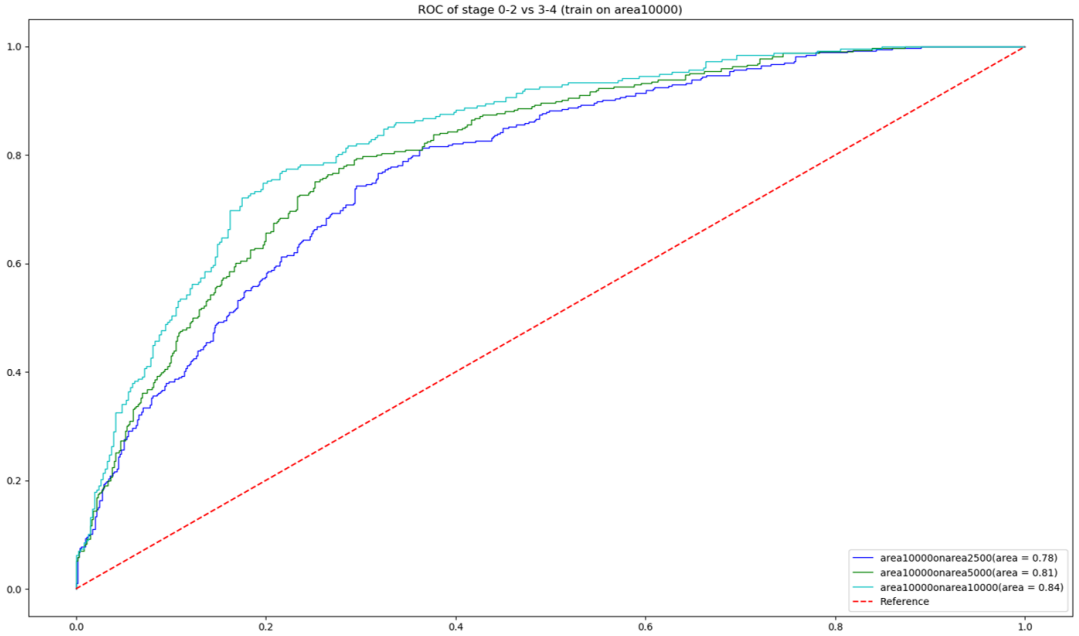


Figure 48: Validation performance with area threshold 10000 without weight decay

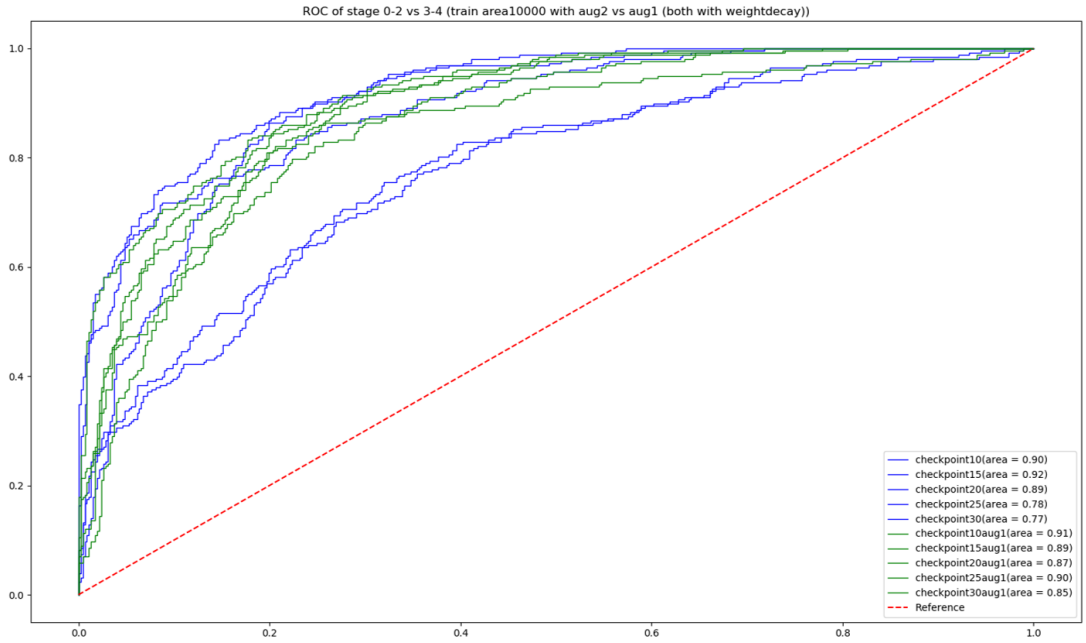


Figure 49: Validation performance with area threshold 10000 and weight decay

Figure 5051 show the training procedure in deeper network(ResNet50/101 and InceptionV3). There is a problem that with those deeper network architecture, fine-tuning on them will cause lots of time. So, in the beginning, what learned during fine-tuning ResNet-18 is just plugged in. Luckily, as shown in Figure 5051, the performance of each architecture is the same as our actual fine-tuned ResNet-18. Those settings are then used during testing.

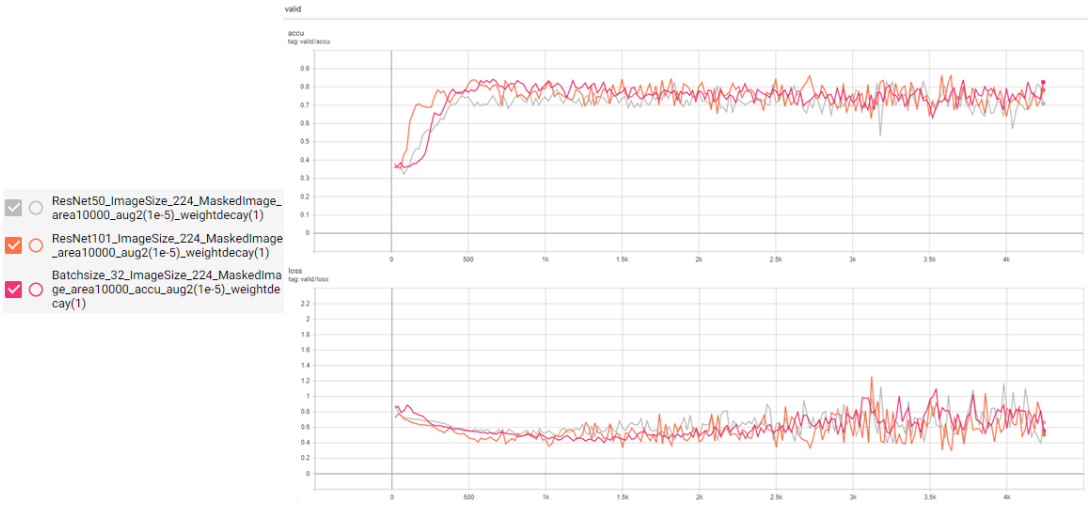


Figure 50: Validation loss/accuracy on ResNet50/101

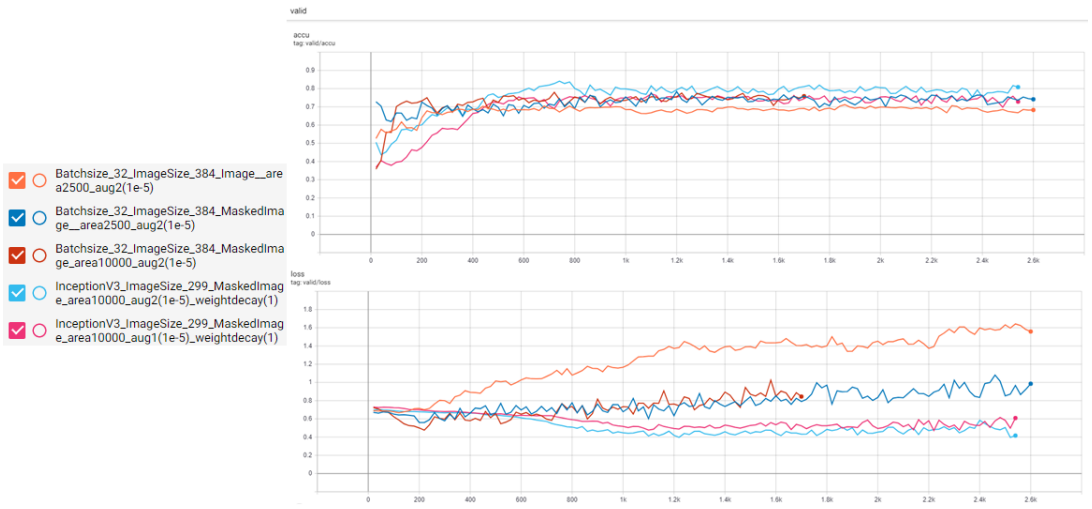
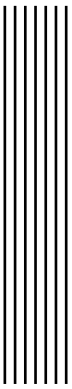


Figure 51: Validation loss/accuracy on InceptionV3



Bibliography

[1] R. Alcolado, M. Arthur, and J. Iredale, "Pathogenesis of liver fibrosis," *Clinical science*, vol. 92, no. 2, pp. 103–112, 1997.

[2] William Rosenberg MD, D.Phil, Julie Parkes MD, "What is liver fibrosis?." `https://www.siemens-healthineers.com/clinical-specialities/liver-disease/what-is-liver-fibrosis`. published in CLI October 2007.

[3] F. M. Sanai and E. B. Keeffe, "Liver biopsy for histological assessment–the case against," *Saudi journal of gastroenterology: official journal of the Saudi Gastroenterology Association*, vol. 16, no. 2, p. 124, 2010.

[4] A. Huber, L. Ebner, J. T. Heverhagen, and A. Christe, "State-of-the-art imaging of liver fibrosis and cirrhosis: A comprehensive review of current applications and future perspectives," *European journal of radiology open*, vol. 2, pp. 90–100, 2015.

[5] L. Sandrin, B. Fourquet, J.-M. Hasquenoph, S. Yon, C. Fournier, F. Mal, C. Christidis, M. Ziol, B. Poulet, F. Kazemi, *et al.*, "Transient elastography: a new noninvasive method for assessment of hepatic fibrosis," *Ultrasound in medicine & biology*, vol. 29, no. 12, pp. 1705–1713, 2003.

[6] H. J. Park, S. S. Lee, B. Park, J. Yun, Y. S. Sung, W. H. Shim, Y. M. Shin, S. Y. Kim, S. J. Lee, and M.-G. Lee, "Radiomics analysis of gadoxetic acid–enhanced mri for staging liver fibrosis," *Radiology*, p. 181197, 2018.

[7] P. Mohanaiah, P. Sathyanarayana, and L. GuruKumar, "Image texture feature extraction using glcm approach," *International Journal of Scientific and Research Publications*, vol. 3, no. 5, p. 1, 2013.

[8] R. M. Haralick, K. Shanmugam, *et al.*, "Textural features for image classification," *IEEE Transactions on systems, man, and cybernetics*, no. 6, pp. 610–621, 1973.

[9] A. Suresh and K. Shunmuganathan, "Image texture classification using gray level co-occurrence matrix based statistical features," *European Journal of Scientific Research*, vol. 75, no. 4, pp. 591–597, 2012.

[10] P. Anthony, K. Ishak, N. Nayak, H. Poulsen, P. Scheuer, and L. Sobin, "The morphology of cirrhosis: definition, nomenclature, and classification," *Bulletin of the World Health Organization*, vol. 55, no. 4, p. 521, 1977.

[11] K. Yasaka, H. Akai, A. Kunimatsu, O. Abe, and S. Kiryu, "Liver fibrosis: Deep convolutional neural network for staging by using gadoxetic acid–enhanced hepatobiliary phase mr images," *Radiology*, vol. 287, no. 1, pp. 146–155, 2017.

[12] K. Yasaka, H. Akai, A. Kunimatsu, O. Abe, and S. Kiryu, "Deep learning for staging liver fibrosis on ct: a pilot study," *European radiology*, pp. 1–8, 2018.

[13] K. Yasaka, H. Akai, A. Kunimatsu, S. Kiryu, and O. Abe, "Deep learning with convolutional neural network in radiology," *Japanese journal of radiology*, pp. 1–16, 2018.

[14] R. Bataller and D. A. Brenner, "Liver fibrosis," *The Journal of clinical investigation*, vol. 115, no. 2, pp. 209–218, 2005.

[15] WHO, "WHO Cancer." `https://www.who.int/news-room/fact-sheets/detail/cancer`. September 12, 2018.

[16] WHO, "WHO Liver Cancer Fact sheet." `http://gco.iarc.fr/today/data/factsheets/cancers/11-Liver-fact-sheet.pdf`.

[17] D. W. McRobbie, E. A. Moore, M. J. Graves, and M. R. Prince, *MRI from Picture to Proton*. Cambridge university press, 2017.

[18] M. K. Seale, O. A. Catalano, S. Saini, P. F. Hahn, and D. V. Sahani, "Hepatobiliary-specific mr contrast agents: role in imaging the liver and biliary tree," *Radiographics*, vol. 29, no. 6, pp. 1725–1748, 2009.

[19] C. Cho, "Gadoxetic acid–enhanced magnetic resonance imaging and contrast-enhanced ultrasonography in the diagnosis of hepatocellular carcinoma," no. 20, pp. 175–91, 2017. DOI: 10.12809/hkjr1716871.

[20] M. M. Galloway, "Texture analysis using grey level run lengths," *NASA STI/Recon Technical Report N*, vol. 75, 1974.

[21] X. Tang, "Texture information in run-length matrices," *IEEE transactions on image processing*, vol. 7, no. 11, pp. 1602–1609, 1998.

[22] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights into imaging*, vol. 9, no. 4, p. 611, 2018.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[24] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[25] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

[26] D. Feier, C. Balassy, N. Bastati, J. Stift, R. Badea, and A. Ba-Ssalamah, "Liver fibrosis: histopathologic and biochemical influences on diagnostic efficacy of hepatobiliary contrast-enhanced mr imaging in staging," *Radiology*, vol. 269, no. 2, pp. 460–468, 2013.

[27] V. Ratziu, F. Charlotte, A. Heurtier, S. Gombert, P. Giral, E. Bruckert, A. Grimaldi, F. Capron, T. Poynard, L. S. Group, *et al.*, "Sampling variability of liver biopsy in nonalcoholic fatty liver disease," *Gastroenterology*, vol. 128, no. 7, pp. 1898–1906, 2005.

[28] P. Bedossa, D. Dargère, and V. Paradis, "Sampling variability of liver fibrosis in chronic hepatitis c," *Hepatology*, vol. 38, no. 6, pp. 1449–1457, 2003.

[29] 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*, pp. 618–626, 2017.