Machine Learning II

Hemorrhage Detection and their Sub-types

Prof. Amir Jafari , PhD

Lilian Sao de Rivera/ Swetha Kalla
5-1-2020

Table of Contents

Table of Figures

Introduction

The need for doctors, health technicians and experts in the medical fields has increased over the years, however we can see the there is a lack of offer for these positions. The problem is higher in rural America and third world countries. Thus, the use of technology in medical diagnostic could be a solution. We can recreate a scenario where dozens of MRI images are scanned by AI and make a prediction about the type of sickness, then only the images that have a low probability are passed down to the clinical eye of an expert. The diagnostics that have a very high accuracy can go directly to treatment. This scenario can help to alleviate the lack of experts in the field, and also can create jobs by allowing more nurse practitioners to work in the field since it requires less time in school.

This project aims to explore the use of neural networks in computer vision and by doing so, participate in the field of medicine and maybe with time and a lot of work help to create these types of scenarios in the places where help is needed.

According to K. Matthews (Sep., 2018), machine learning can improve how doctors can use MRI images. Matthews describes how ML can be used to produce high-quality image with less data by using enhancing algorithms, making it easier for doctors to make a diagnosis. She also mentions the use of these types of algorithms to minimize breakdowns in the MRI's machines. This equipment needs to be calibrated in such a way that the exact amount of energy can be induced in the person. The better the machine is calibrated the better the results will be. ML algorithms can do a better job calibrating this complicated equipment. Other use is to aid better decision making by presenting results on MRI's based on data compiled from several years to predict the extent of a tumor. This problem not only can help the diagnosis for an outpatient consultation but also in the operating room. This last problem is the path we chose to understand and implement a neural network to aid diagnosis in the detection of the type of brain tumors.   Our research question is as follows:

"Classify the type of tumor in a MRI based on the history of annotated images presented in a dataset provided by a Kaggle competition"

Problem

The problem was presented by a Kaggle competition on November 2019. The aim of the competition was to detect the type of intercranial tumor in MRI images. There are five type of tumors: Intraparenchymal, Intraventicular, Subarahnoid, Subdural, and Epidural. The aim of the solution is to detect if an image presents a tumor and what type of tumor it is. There are several problems associated with the images. The first one is that the tumors are spread inside the brain in very particular ways, thus sometimes is difficult to distinguish what type of case we have at hand. Other problem is that the tumors are blood clots that are difficult to separate from the brain matter. The images that we are using are monochrome images that need to be modified to clearly identify the different densities in the brain.

EDA

Labels

The dataset is from Kaggle competitions and it comprises two sets of data A CSV File with 2 Columns

- ID
- Label

ID Contains the name of the image plus the type of hemorrhage and an extra row with the suffix "any". if any is marked in label as one, means that the image presents some kind of the types of hemorrhages. Label contains a zero or a one which indicates the type of hemorrhage presented in the image

This means that each image is repeated 6 times in the dataset. This is an example :

ID | Label |

ID_63eb1e259_epidural | 0 |

ID_63eb1e259_intraparenchymal | 0 |

ID_63eb1e259_intraventricular | 0 |

ID_63eb1e259_subarachnoid | 0 |

ID_63eb1e259_subdural | 0 |

ID_63eb1e259_any | 0 |

Images

The images are embedded in DCOM file, which contains images of 512x512, monochrome which means that is only one channel. The metadata of the images is also saved in the DCOM file. The images are saved in a standard form so they should be translated in Hounsfield scale, so the algorithms can detect the brain matter from the subdural matter. The images will be resized to fit the pretrained algorithm. In Fig.2 below we can observe the different visualization of the brain using the scale, which seems to be a good approach for our problem.

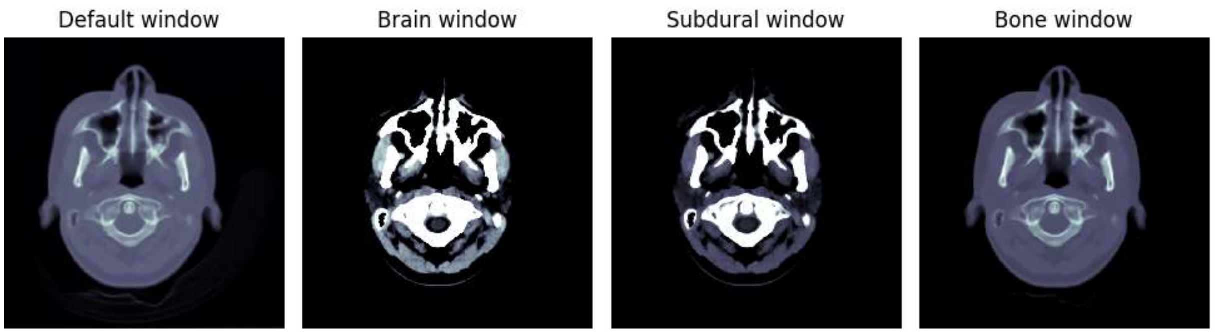


Figure 1 Brain Images with Hounsfield Scale

The distribution of the labels shows that the dataset is highly imbalanced because we are using under sampling; with a pretrained model we will be using a more balanced dataset. In Fig. 3 below we can see the distribution of the images, and that we have also many images that have more than one annotation which means that we have a multiclass classification problem. In the graph on the left you can observe the quantity of images with no annotations, one annotation and on the left with more than one annotation.

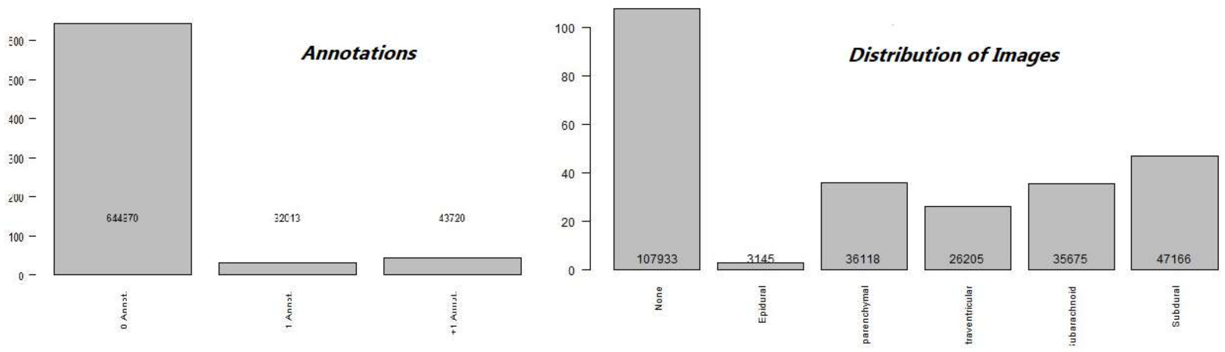


Figure 2 Distribution of Images

Methodology

Due to the complexity of the problem we decided to divide the problem in two parts. The fist part aims to build all the tools necessary to measure the results of our model. The other purpose is to have a working model without the complexities of a large dataset. The second part consists of a more complex model with a well-defined dataset, still we chose under sampling because we have a reduce time to train the model to deliver results. We can see in Fig.3 below the configuration of the execution of our project. The result of the first part is a viable product that we can convert into a more complex tool.

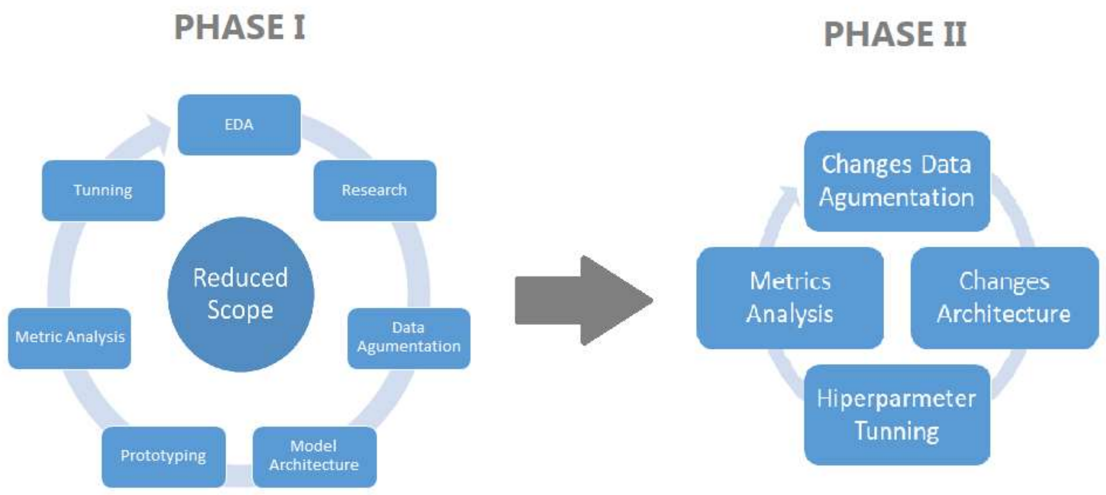


Figure 3 Project Planning

Based on these assumptions we decided to use transfer learning. Our problem is related to feature detection in an image, the features are the different types of tumors, thus the best approach is to use a pretrained model.

Architecture

These types of problems are widely studied in the field of medicine and machine learning so we researched what was the best approach to solve the problem, and based on studies suggested by experts we could modify, learn an build novel solutions. However, our approach was to learn techniques that are already in place to research and understand better our problem. An interesting approach was suggested by A. Esteva, B. Kuprel , et al.(2017) which suggested that “a CNN trained on a finer disease classes perform better than trained directly on several classes”. The study had three layers of leaning first understanding if there was a sickness or not, then training what type of sickness one at a time and then they follow a multilabel classification problem. We decided to go for two paths, first implementing a multiclass problem, one annotation per image, and then train a second model which will solve a multiclass problem for our images.

The other problem that we had was the augmentation of the images. The position of the tumors in the brain is important to identify the type of tumor, thus moving around the images did not seem like a good approach for this problem. Instead we found a paper by M. Jadergerg, K., Simonyan , et.al. (2016) developed in Google DeepMind in UK that presented a Spatial Transformation Network which receives an image and generates a transformation matrix to transport the image to a better position for the following layers of the network. According to this paper “networks that include Spatial Transformation Networks can transform regions to a canonical, expected pose to simplify recognition in the following layer.” This type of approach was what we were looking for, a canonical position for the brain, which means a centered image.

Based on our research we designed the architecture of our network.

Model I

The fist neural network as you can see in the Fig. 4 down below shows a Spatial Transformation Network (STN) at the beginning of the model followed by a CNN with three layers, the fist layer is a pretranning model , VGG16, we chose this model for our first viable product because is a fast network, it only has 16 layers. The best model would be inception_v3 which we found out was use widely in medical image recognition but it is a very complex model, thus we decided to include this model for the second phase when every element was correctly ensembled. The figure shows that layers one, two, four and five are the ones that are trained in this stage. The purpose of this model is to train the model with only one annotation.

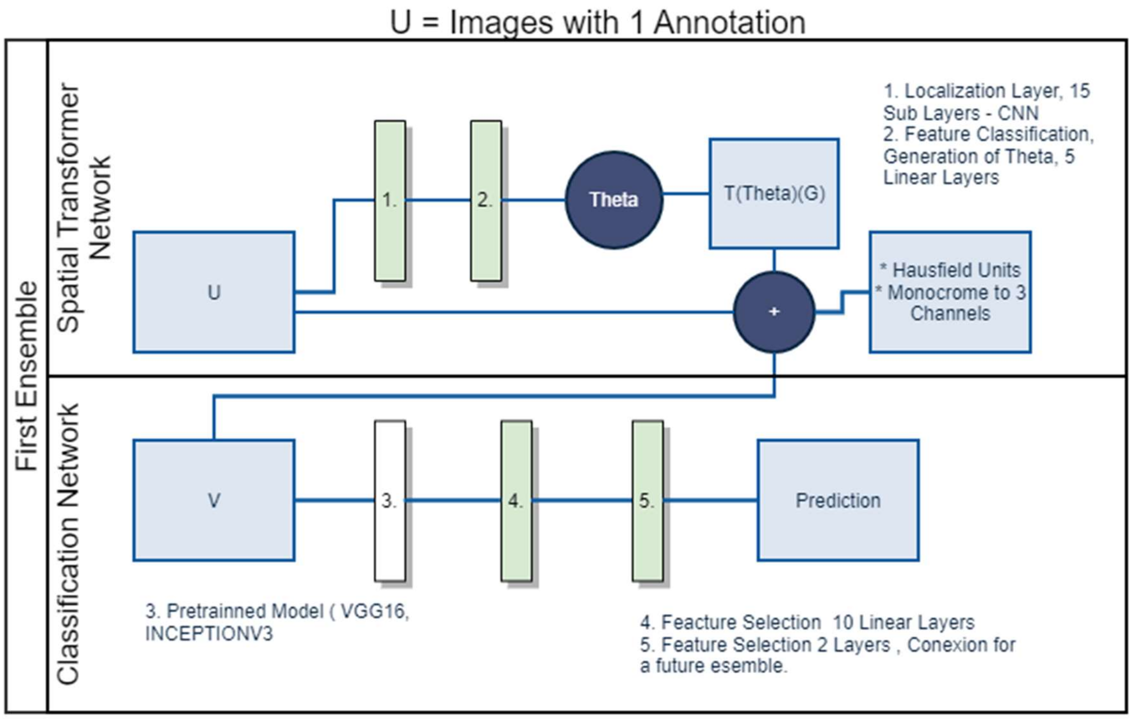

U = Images with 1 Annotation
First Ensemble
Spatial Transformer Network
Classification Network
U
1.
2.
Theta
T(Theta)(G)
+
* Hausfield Units
* Monocrome to 3 Channels
1. Localization Layer, 15 Sub Layers - CNN
2. Feature Classification, Generation of Theta, 5 Linear Layers
V
3.
4.
5.
Prediction
3. Pretrainned Model (VGG16, INCEPTIONV3
4. Feacture Selection 10 Linear Layers
5. Feature Selection 2 Layers , Conexion for a future esemble.

Figure 4 Model 1 , STN plus a Multiclass Classifier

Model II

The second model is a multiclass classifier, however we are using our first model as a pretrained model, from our first model we are changing the last layer which in our model can be depicted as layer one, layer two and three are trained and can be extended as much as we want so the model is better trained.

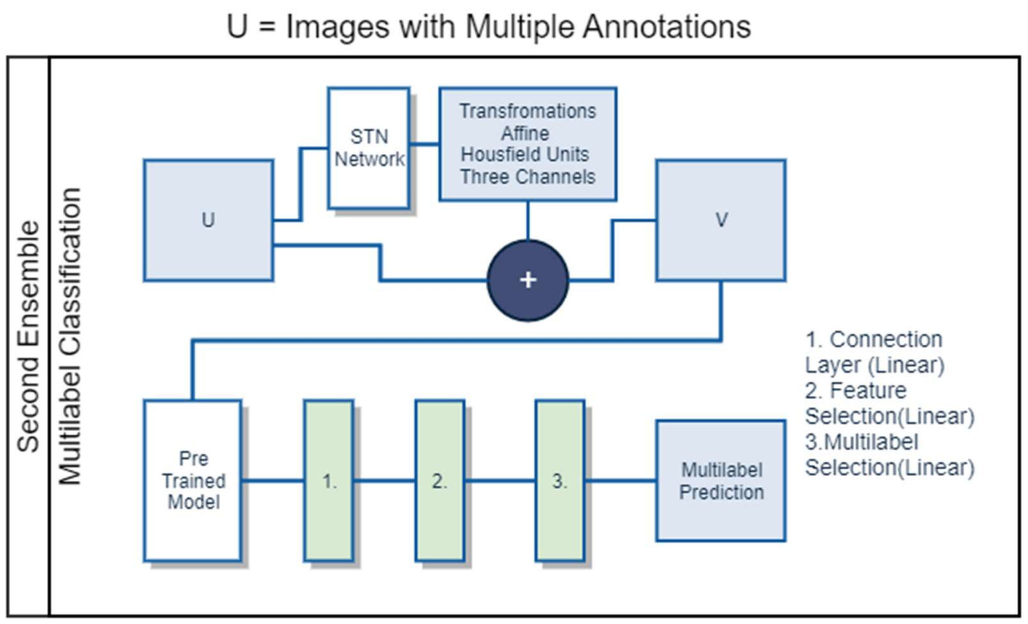

U = Images with Multiple Annotations
Second Ensemble
Multilabel Classification
U
STN Network
Transfromations Affine Housfield Units Three Channels
+
V
Pre Trained Model
1.
2.
3.
Multilabel Prediction
1. Connection Layer (Linear)
2. Feature Selection(Linear)
3.Multilabel Selection(Linear)

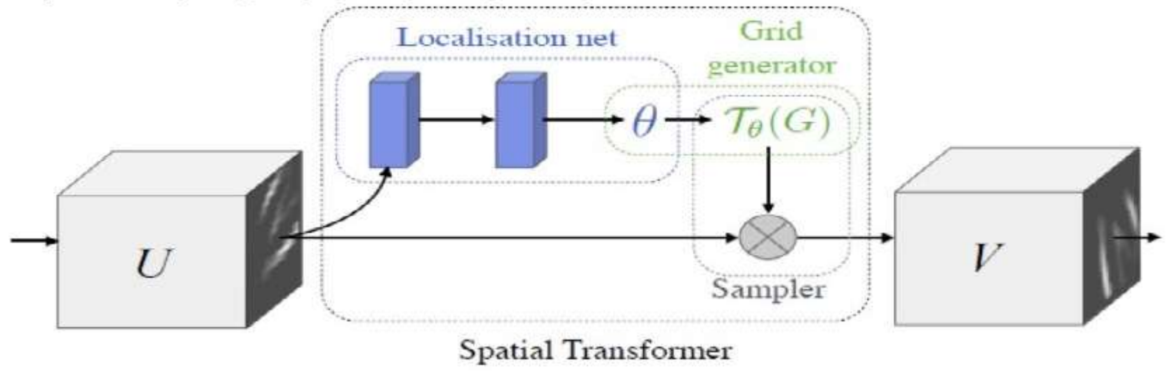
Figure 5 Mode2, a Multilabel Classification Network

STN

The Spatial transformation network is included at the beginning, but in other for it to work it has to be connected to a fully functional network. Our first attempts failed because the following network was a simple linear network, once we understood that the purpose of the network is to help the main network to achieve its purpose the SNT started to work. We have some interesting results in our fist phase but in the second phase we could see images that were actually centered. This feature modification clearly helped the prediction process.

The purpose of the STN is to create a transformation matrix that will project our original image into a better position in the plane or Grid(G). The important part here is that we are using and affine transformation. These types of networks can be used for attention, cropping, rotation, as well as non-rigid deformation. We found an example in the blog for Torch developers which cover exactly what we needed. But further experiments need to be done to understand the many capabilities of STN.

Ref: Review STN-Spatial Transformation Network by Sik-Ho Tsang , Towards Data Science, (Jan, 2019) Orignal paper by Google DeepMind, London UK.



$$\begin{pmatrix} x_i^s \\ y_i^s \end{pmatrix} = \mathcal{T}_\theta(G_i) = \mathtt{A}_\theta \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \end{bmatrix} \begin{pmatrix} x_i^t \\ y_i^t \\ 1 \end{pmatrix}$$

Affine Transformation

$$V_i^c = \sum_n^H \sum_m^W U_{nm}^c k(x_i^s - m; \Phi_x) k(y_i^s - n; \Phi_y) \qquad \forall i \in [1 \ldots H'W'] \ \forall c \in [1 \ldots C]$$

Figure 6 Spatial Transformation Network - Affine

Data Augmentation

As we have stated in the previous section, we decided to under sample our dataset and instead of moving around the images we decided to use STN to look for canonical positions of the image. In phase one we decided to use small amount or records to generate our viable product.

- 500 images per class
- 1000 images annotated
- Balanced dataset
- 75% training 25% validation
- 100 images per class and 200 images with tumor and no tumor for testing

For the second phase we still used a small sample since we did not have enough time to train the models. Even with the reduced number of images the second phase used 8 hours for the first model and 6 hours for the second model to train. Our dataset for the second phase look like this:

- 1000 images per class
- 2000 images annotated
- Balanced dataset
- 65% training 13.5% validation 13.5% testing

Metrics and Parameters

For the first phase we used

- Sigmoid, BWLoss, Adam
- Validation Loss, Training Loss, Accuracy
- LR on plateau starting with 0.015 on validation loss
- Early stopping on validation loss after 5 epochs with no improvement.

For the second phase we used

- Sigmoid, BWLoss, Adam
- Validation Loss, Training Loss, Accuracy
- LR on plateau starting with 0.015 on training loss
- Roc_AUC , F1_Score
- Early stopping on validation loss after 5 epochs with no improvement.

Technical definitions

- We choose to implement data loaders for our model. We use the approach by A. Amidi and S. Amidi presented in their blog from Stanford University. This approach can be implemented very quickly and the transformation functions can be included, making the use of memory very efficient and the transformations very flexible.
- Pytorch was used to design the networks. Since It was the first time that we used this framework it was laborious to build all the controls, metrics and mechanics of the algorithms for training, validations and testing.
- The application was divided y three phases for each model.
    - Training and Validation
    - Testing
    - Metric processing for dashboards
- A function to extract the Affine Matrix was included in the fist model, which purpose is to test the proper functioning of the STN model.
- CNN were used in the definition of the STN, and fully connected Linear layers for label classification at the end of each model.

Conclusions and Recommendations

- Batches of 15 or 10 images yield better results.
- The prototype shows two models are a good approach
- The use of the function loss in the validation phase as a tool to choose the best model might not best the best option
- The accuracy and other metrics are necessary to evaluate the model.
- The number of images use in “Any” category shows a good metric for sampling the other classes for the second phase.
- It is important to include feature enhancing in the model
- The prototype allowed us to build a technical working infrastructure to continue with the second phase which includes:
    - Data Augmentation
    - Sampling

- Fine Tuning the models
- Include Inception V3.

Results

From the graph below we can see that the first model shows declining leaning process, whereas the validation phase shows a different behavior. These results are from the first phase where the sample is very small. The behavior of model II is mor stable in the validation, which means that having two models for the process shows stability to the process.

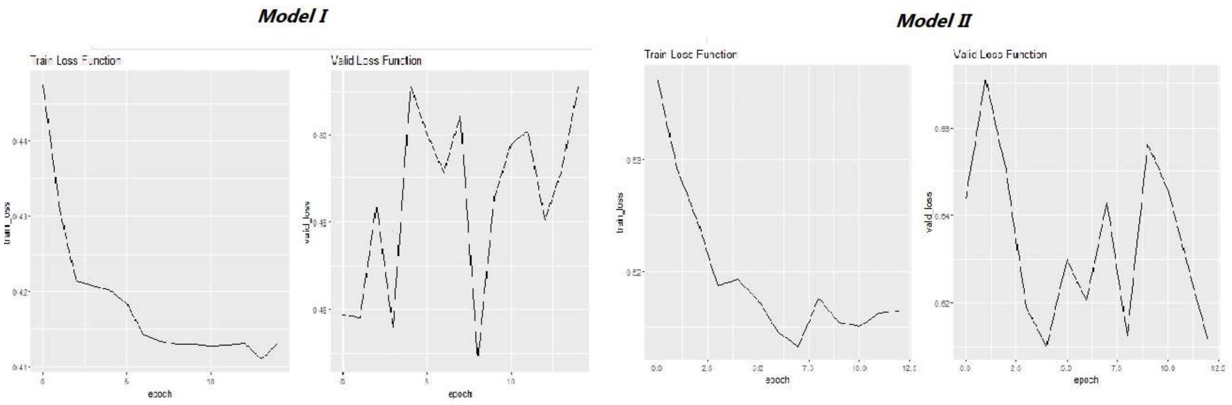


Figure 7 Result Val Loss Train Loss

The AUC in model I in the first phase shows an a good result for the "any" levels, whereas the other labels are very close to the chance lines. You can see these trends in the first graph below. Fig.8. But the AUC on the left shows improvement. That means that what we learn in the first model makes a huge difference in the second model.

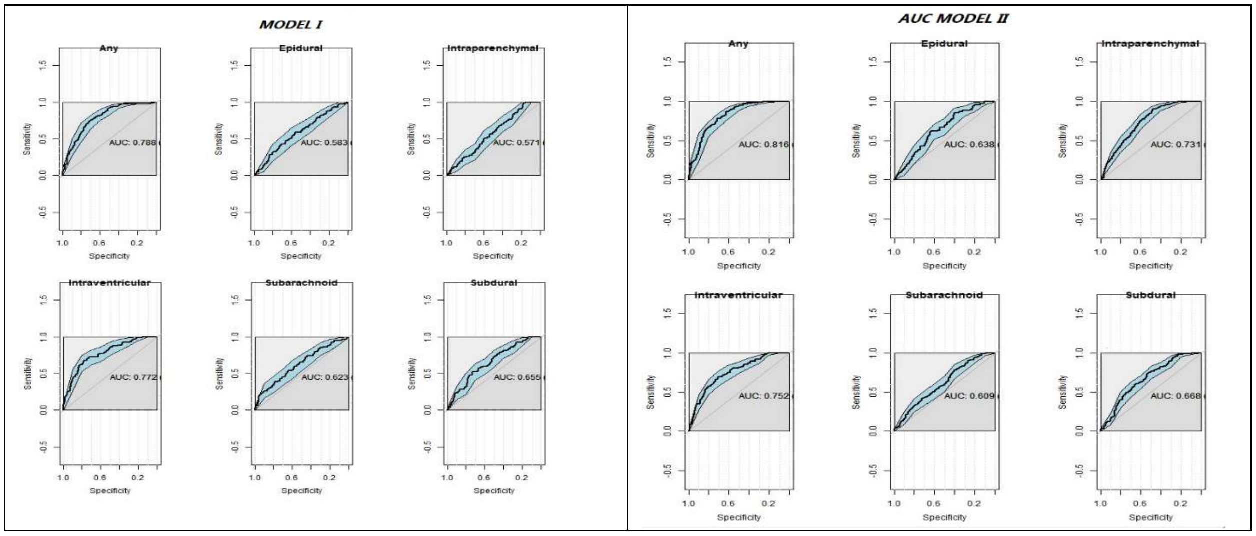


Figure 8 AUC For Model I and Model II in Phase I

The results for phase two shows a better AUC in general in model I and Model II

For Model I we got the following results

Test set: Loss: 0.382323

Accuracy : 13.741497

AUC : 0.723737

f1 : 0.177195

For Model II we got the following results

Test set: Loss: 0.468402

Accuracy : 16.244898

****AUC : 0.748952**

****f1 : 0.425590**

We can see in the second model AUC general of 0.74 whereas an F1 of 0.42 for 19 epochs with three thousand images per class. What we are observing in this experiment is that our model can improve greatly with more time to process and more images.

Another success that we obtain from this experiment was the transformation, the images were transformed during the process to a better position. We can see from the figure below that image on the left shows the nose bone to the left , whereas in the image of the right is aligned to the center. The transformed image on the right shows a shape more centered in the grid, the canonical position. This result was obtained in the second phase.

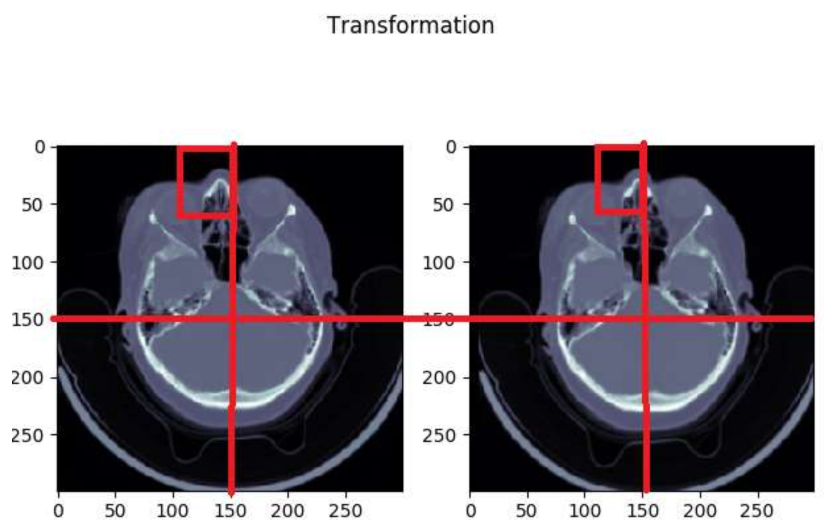


Figure 9 Transformed Images in phase II

Conclusions

- The second phase exhibited a significant improvement in terms of valid_loss, train_loss, validation loss and accuracy.
- AUC and F1 score were included in the second phase with an average of 70% and 17%d. F1 still does not yield a good score, however the images can be increased and more epochs can be included in the training.
- The substitution of VGG16 for Incenption_V3 led to this significant improvement of the results.
- The inclusion of AUC and F1 score made evident the behavior of the training process.
- Each epoch lasted about half an hour for the fist model and twenty minutes for the second model, which shows that we need to budget more computer processing time for more images.

Recommendations

- More time in training is necessary and more images which will require more time, so the planning for the training is important.
- Feature enhance only included using Hounsfield scale for the images, using brain window for channel two, and epidural window for channel 3, however we have seen studies posted of competition of Kaggle that using an RGB color space for each window might help the model to enhance the features. This would be an interesting experiment to try.
- The study from A. Esteva, B, Kuperl, et al. (2017) shows three stages of evaluation, we only use two, thus trying this approach will be also an interesting experiment to try on our dataset.

References

A.Amidi, S.Amidi (2019). A detailed example of how to generate your data in parallel with PyTorch, personal blog.

A.Esteva, B.Kuprel, et al.(Jan, 2017). Dermatologist-level classification of skin cancer with deep neural netowors. Letter. Page. 115-125.

H.Ye, F.Gao, Y.Yin (2019). Precise diagnosis of intracranial hemorrhage and subtypes using three-dimensional joint convolutional and recurrent neural networks.

Kaggel (Nov, 2019) Hemorrhage Tumor detection Competition.

M.Jaderberg, K.Simonyan(Feb, 2017). Spatial Transformation Networks. Google DeepMind, UK.