

# Brain Aneurysm Classification from MRI Images using MONAI Framework

**Vivek Gupta**  
Stanford University  
vivekkrpg@stanford.edu

**Navya Veeturi**  
Stanford University  
nveeturi@stanford.edu

**Abhishek Akkur**  
Stanford University  
abhakk01@stanford.edu

## Abstract

Detecting brain aneurysm accurately based on the Computerized Tomography Angiography (CTA) or Magnetic resonance imaging (MRI) scan is a challenging and time intensive task for humans, mainly due to its smaller size (ranging from 2mm to 10mm). In this work, we use Convolution Neural networks (CNN) classification model using DenseNet architecture to detect aneurysm from MRI scans. To achieve this goal, we used medical imaging data provided by VasoGnosis Inc [15], collected from various hospitals along with open source imaging data. Because of limited availability of positive aneurysm data, we have experimented with techniques like transfer learning, drop out, learning rate update, data augmentation and K-fold cross validation to achieve the best result with an accuracy of 95.54%.

## 1 Introduction

A brain aneurysm is a bulge or ballooning of a blood vessel in the brain and it can leak or rupture, causing bleeding which may also lead to brain hemorrhage. It is an emergency medical situation that needs immediate neurological evaluation and testing for survival. [19] The size of aneurysm is less than 1% of the size of the MRI scan and hence it is challenging to identify and segment the aneurysm using Deep Learning. In this task, we explore how Deep Learning Neural Networks help solve the classification of brain aneurysm from the MRI scans. The input to our model are 3D images, the scans from hospitals and open source images without aneurysm. We are using DenseNet network architecture and MONAI (Medical Open Network for AI) [12] platform, an open-source PyTorch-based framework to train and improve the accuracy of the classification task. We have collaborated with a startup, VasoGnosis Inc., an AI-powered diagnostic and surgical planning software platform provider for cerebrovascular disease [15], for the data and their guidance through this project.

## 2 Related work

Several researchers have been working for early and quick diagnosis of Brain Aneurysm using Deep learning. In [8][4], researchers explored using Deep CNN for classification. In [5][1], techniques have been explored to do multi-class classification to identify different types of brain aneurysms. Inspired by object detection algorithms, [3] used Faster region CNN for detection of location of aneurysm. Other works [2][9][6][7] have attempted to automatically segment brain aneurysm. In [7], a comparative study was done between UNet and DeepMedic CNN to highlight limitations of UNet for segmenting aneurysm. Similarly, [6] used DeepMedic CNN to segment aneurysm. Others [2][9] have used a 3D Dense UNet models to achieve good results with segmentation.

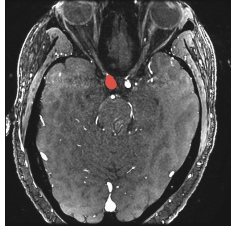
## 3 Dataset and Features

Our dataset consists of 3D MRI images in NIfTI format from 2 different sources. VasoGnosis Inc.[15] collected positive aneurysm scans with dimension (512, 512, [100-150]) from various hospitals and provided the segmentation masks for the same and we also used open source IXI dataset with brain MRI images from Brain Development Org[14]. Distribution of data provided in the below table:

Dataset (60/20/20)				
Class	Training	Validation	Test	Total by Class
Positive	228	77	76	381
Negative	240	80	80	400
Total by Distribution	468	157	156	781

Table 1. Dataset distribution details

Below are the sample images of the data that we have received for positive and negative cases.



**Figure 1.** Aneurysm Positive



**Figure 2.** Aneurysm Negative

These images usually have aneurysm size  $<1\%$  size of the image and rest of the image contains no information for the model to learn from. Due to this high class imbalance and sparsity of the aneurysm data per image, we have used random cropping and sampling mechanism based on mask to extract more positive slices from the image during training.

### 3.1 Data Augmentation

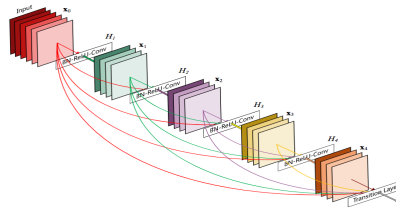
We used augmentation techniques such as, random cropping, scaling image intensity ranging between 0 - 1, changing orientation of 3D image. The 3D orientation method performs left to right on first dimension, posterior to anterior for the second, inferior to superior for the third image. We used MONAI platform provided classes for performing the above mentioned processes. For random crop, we used a crop size of (64, 64, 64) on raw and segmented images and then sample to 20 per input image with positive to negative label ratio of 1:1 for training. This method also crops random fixed sized regions with the center being a foreground or background voxel based on the positive and negative ratio. Class balanced random cropping solves the problem of class imbalance and also reduces MRI image source distribution specific information. After cropping, we generated the classifier label for the image on summing the values of the sample, thereby increasing the overall labelled data for better accuracy. During this process we set the classifier label as 1, if any of the voxel in the cropped segmentation mask image is non-zero. This cropping mechanism increased the number of positive labeled data and which also addressed the class imbalance problem.

## 4 Methods

We used DenseNet network for aneurysm classification. Previous work by VasoGnosis has used ResNet architecture similar to [4] that has produced sensitivity and specificity of upto 85%. Due to very compelling advantages like, strong gradient flow, higher computational and memory efficiency with DenseNet, we chose this architecture. We used DenseNet with 121 layers as a starting point and then explored bigger networks DenseNet169 and DenseNet256. We tuned the hyperparameters like dropouts, learning rate, sampling size, mini-batch size and number of epochs and compared the results. We then tried transfer learning technique based on the trained model on the DenseNet121 architecture and performed cross validation using train-test split and k-fold cross validation methods to evaluate our model.

### 4.1 Architecture

DenseNet is an improvement to ResNet architecture, where each layer receives feature maps from all preceding layers by using concatenation of output layers instead of summation. For  $L$  layers, there are  $L(L+1)/2$  direct connections. This encourages feature reuse throughout the network and leads to more compact models. Each layer has direct access to the gradients from the loss function and the original input signal, leading to an implicit deep supervision. This helps training of deeper network architectures and furthermore the dense connections have a regularizing effect, which reduces over-fitting on tasks with smaller training set sizes.[10]



**Figure 3.** DenseNet with 5 layers

DenseNet is composed of Dense blocks and Transition layer and within each block, the layers are connected to all their previous layers. Each layer consists of Batch Normalization, ReLU activation and 3x3 Convolution. The transition layers between two dense blocks assure the down sampling and is made of Batch Normalization, 1x1 Convolution and Average pooling. The bottleneck layers are added to reduce the number of inputs for later layers in the network. Because of concatenating feature maps, the channel dimension is increasing at every layer. The  $l^{th}$  layer will receive all previous feature maps:

$$x_l = H_l([x_0, x_1..x_{l-1}])$$

If  $H_l$  produces  $k$  feature maps every time, then we can generalize for the  $l^{th}$  layer:

$$k_l = k_o + k * (l - 1)$$

This hyperparameter  $k$  is the growth rate and this regulates how much information is added to the network each layer.

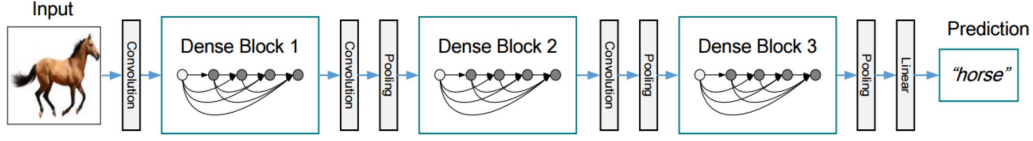


Figure 2. A deep DenseNet with three dense blocks. The layers between two adjacent blocks are referred to as transition layers and change feature map sizes via convolution and pooling.

Figure 4. Dense Blocks from paper

In our project, we used DenseNet with 121 layers with initial feature size to be 64, a growth rate of 32, a multiplicative factor for number of bottle neck layers as 4 and number of layers in each block to be 6 , 12 , 24, 16.

## 4.2 Loss Function

We have used the cross entropy loss function for training our model along with Adam optimizer with learning rate of 1e-5. The loss function as given by PyTorch[18] is:

$$loss(x, class) = \log\left(\frac{\exp(x[class])}{\sum_j \exp(x[j])}\right) = x[class] + \log\left(\sum_j \exp(x[j])\right)$$

## 4.3 Evaluation Metrics

We evaluated our model based on the sensitivity and specificity of test performed on the validation and test dataset. We also captured the sample and total MRI accuracy of our model and plotted the confusion matrix to understand the F1 score and recall. The sensitivity is the proportion of samples that are actually positive that produce a positive result using the test in question and also referred to as the true positive rate (TPR) The specificity is the proportion of samples that result in negative using the test in question that are actually negative. This is also referred to as the true negative rate (TNR).

# 5 Experiments/Results/Discussion

As a starting point, we augmented the data using random cropping to generate balanced positive-negative samples, scaled intensity and performed 3D orientation. We continued to improve the model by tuning the hyperparameters, implementing transfer learning and k-fold cross validation approach for which the details are provided in subsections.

We generated the classifier label based on the provided segmentation mask to identify each sample to be aneurysm positive/negative. We trained DenseNet(121/169/264) models on the cropped samples by splitting data between training, validation and test to avoid data leak and used dropout to avoid over-fitting.

## 5.1 Hyperparameter Optimization

In the process of fine-tuning the hyperparameters, we captured the training and test results post any updates and compared them to evaluate the best possible values. We used our best model, which was DenseNet121 architecture with cross entropy loss function, Adam optimizer at learning rate of 1e-5 and mini-batch size of 2 and 5 epochs. We made step by step updates to understand the impact of each on the final test accuracy. Few optimizations include, reducing the sample size from 30 to 20, adding dropout of 0.2 to 0.3, learning rate of higher and lower values.

## 5.2 Transfer Learning

Due to limited number of medical imaging data we attempted Transfer Learning technique for making further improvements. We used our best trained model ( 373 scans) and applied transfer learning and retrained the model with more data ( 781). Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. In our case, the task for image classification was still the same but it slightly improved the training and validation accuracy.

## 5.3 K-fold Cross Validation

K-fold Cross Validation (CV) is a well-known strategy in machine learning and deep learning areas to achieve more accurate and more stable outputs. In this technique, we first did the split of entire data into training(80%) and test(20%) and then further divided the training set into 5 folds and trained 5 models with every 4 fold data. This ensures that each fold is used as a testing set at some point thereby reducing the over-reliance on the accuracy obtained from one test set. For evaluating the

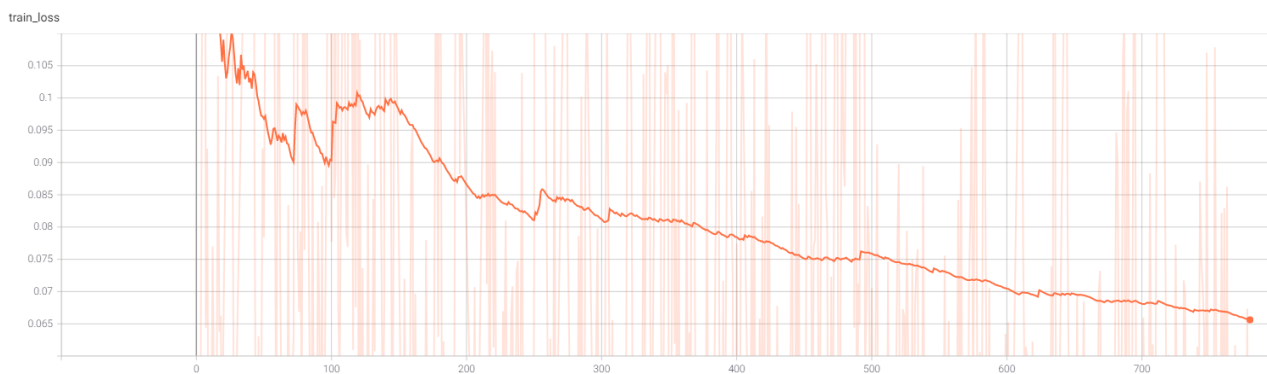
accuracy with the new test data, we ran the tests against all 5 models and picked the best performing model. CV technique used by MONAI [16].

Below is the table comparing the results of various models we trained as part of this project.

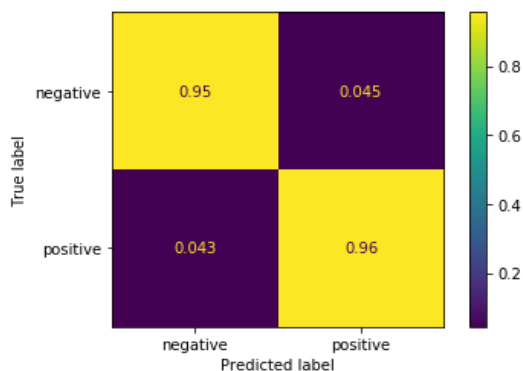
Experiments and Results Chart											
Models Trained	Dataset	Data Split	Train Crop Method	Test Crop Method	Sample Size	Learning Rate	Drop out Rate	Epoch	Accuracy	Sensitivity	Specificity
<b>DenseNet121 (Model-1)</b>	373	80/10/10	Random Crop PosNeg (64,64,64)	Random Crop PosNeg [24] (64,64,64)	20/200	1e-5	0.0	5	Sample 96.34% MRI 89.20%	99.10%	81.50%
<b>DenseNet169 (Model-2)</b>	781	80/10/10	Random Crop PosNeg (64,64,64)	Random Spatial Crop [25] (64,64,64)	30/120	1e-4	0.3	5	Sample 98.06% MRI 76.92%	100%	70.00%
<b>DenseNet121 (Model-3)</b>	781	80/10/10	Random Crop PosNeg (64,64,64)	Random Spatial Crop (64,64,64)	80/120	5e-6	0.2	2	Sample 93.29% MRI 48.7%	100%	13.04%
<b>DenseNet264 (Model-4)</b>	781	60/20/20	Random Crop PosNeg (64,64,64)	Random Spatial Crop (64,64,64)	20/200	1e-5	0.0	5	Sample 98.01% MRI 90.44%	91.82%	88.30%
<b>DenseNet121 (Model-5) Transfer Learning</b>	781	60/20/20	Random Crop PosNeg (64,64,64)	Random Spatial Crop (64,64,64)	20/200	1e-5	0.0	5	Sample 98.06% MRI 95.54%	98.48%	93.41%
<b>DenseNet121 (Model-6) 5-Fold CV</b>	781	60/20/20	Random Crop PosNeg (64,64,64)	Random Spatial Crop (64,64,64)	20/200	1e-5	0.0	5	Sample 98.12% MRI 90.44%	98.73%	94.87%

**Table 2.** Model Comparison Chart

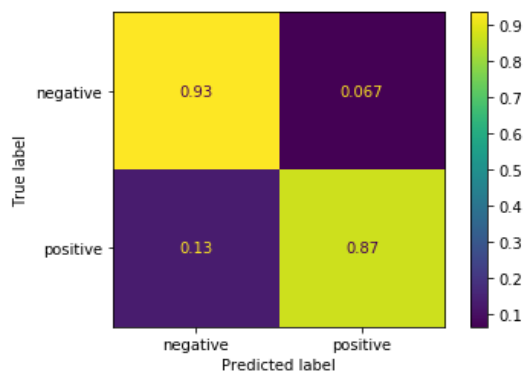
The error loss and train loss plots are created using TensorBoard [22] and are provided for our best performing models i.e. Model-1, Model-5 and Model-6. The confusion matrix is plotted using SciKit Learn [23].



**Figure 5.** Training Loss for Model-5



**Figure 6.** Confusion Matrix for Model-5



**Figure 7.** Confusion Matrix for Model-6

## 5.4 Learnings

### 5.4.1 Data distribution

In early part of the project, we used data augmentation techniques like Scaling Intensity, resizing the image to (64,64,64). We achieved an accuracy of 100%, which was misleading since resizing the image (lower resolution), reduced size of aneurysm and hence the model learnt only distribution of MRI source and did not effectively learn to detect aneurysm. Model failed to predict the negative test set provided by VasoGnosis, with same source as other MRI images. Appendix [1] for training results.

### 5.4.2 Validation metrics

We initially cropped the positive and negative images and calculated the accuracy per sampled image, which was very high but this did not represent the total accuracy of the image. To fix this, we combined the predicted output of all samples per image and then recalculated the metrics. The resulting accuracy dropped significantly for positive MRI images because the accuracy evaluation algorithm had zero tolerance for errors. To improve the accuracy, we set threshold to 1 to tolerate false predictions. This accuracy improvement technique was later confirmed by VasoGnosis.

## 6 Conclusion/Future Work

### 6.1 Results Analysis and Conclusion

Based on the samples of MRI images provided by VasoGnosis, our model outperforms the targets that were set initially as 85% sensitivity and 85% specificity. Model-1 worked well on sampling the images, but resulted with less accuracy on entire MRI image with 0 threshold (no tolerance for errors). On increasing the threshold, the accuracy improved. Hyperparameter updates like, increasing learning rate, dropout or networks size did not have significant impact on the final outcome. Statistically we noticed that, using greater than 20 samples per MRI image caused degradation in accuracy. This is because of the class imbalance, which resulted the model to misclassify few negative samples as positive and because the accuracy algorithm tolerates a threshold of 1, which means, if there are more than 1 false positives out of 30 samples, it would be considered as positive, resulting in overall negative prediction accuracy to drop. We also found that transfer learning to re-train model on different source of MRI images, was very successful as seen in Model-5. Additionally, training using cross validation also showed very good results as seen in Model-6

### 6.2 Inference

VasoGnosis currently deploys a patch based scanning algorithm that detects aneurysm in each patch. For small sizes of aneurysm, only 1 patch out of [200-500+] patches could have aneurysm. This reduces aggregated classifier accuracy. We use Spatial random crop method which increases the probability of multiple samples to have aneurysm and hence have ability to tolerate some incorrect predictions. Another method to explore would be inspired by [5] to use anatomically informed CNN model so that more patches are selected in parts of MRI that have higher probability to contain aneurysm.

### 6.3 Segmentation

After classification, we need to find the region where aneurysm exist in the MRI image. Many papers [2][6][7][9] have been published on segmentation of brain aneurysm. We did attempt segmentation using UNet provided in MONAI framework. This is still work in progress and based on [7] Unet might not yield great results. Other models to be tried include mask R-CNN [17] for segmentation.

## 7 Contributions

All the team members contributed equally towards understanding of baseline code from MONAI, network architectures and evaluation metrics. Guidance on data and various models were provided by Ali Bakhshinejad and Alexa Miller from VasoGnosis. As a team, we met frequently (4-5 days per week). Synced weekly with VasoGnosis team and held regular touchbase with TA and mentor (Jonathan Li) on the progress and feedback. Jonathan Li provided valuable inputs and insights on various methods that we could try to improve the accuracy and helped understand the results.

Finally, we are very thankful to our instructors, Andrew Ng and Kian Katanforoosh who have shared key foundations of deep learning and taught us many advanced concepts that we could apply to our project which eventually contributed to the overall success.

Below are the contribution details per member:

### 7.1 Vivek Gupta

Contributed documentation of initial proposal, mid-term report, final report and video. Helped installing relevant softwares and setup of EC2 machines on AWS. Contributed towards code structure, development and valuation metrics algorithm using MONAI platform. Contributed towards documenting intermediate results and communicating same with Vasognosis/TAS. Contributing towards trying segmentation code development, patch based inference for Vasognosis.

## 7.2 Navya Veeturi

Initial project idea brainstorming and discussion. Setting up AWS servers, managing the capacity and installing relevant packages for running PyTorch based trainings. Data loading, data augmentation code development and Grid patch data set [26] implementation for cropping the 3D image. Contribution towards the code development for training the model and plotting metrics. Hyperparameter tuning. Running inference on new test data. Coordinating within team and setup regular sync-ups with TA and VasoGnosis team. Midterm, Final project report writing and video creation.

## 7.3 Abhishek Akkur

Contributed towards project proposal, milestone and final reports. Brainstorming various architectures and hyperparameters. Exploring Data Augmentation techniques and Classification metrics. Experimenting different architectures for classification and building models. Consolidating results and plots of best models.

## References

- [1] H. Ko, H. Chung, H. Lee and J. Lee, "Feasible Study on Intracranial Hemorrhage Detection and Classification using a CNN-LSTM Network," 2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC), Montreal, QC, Canada, (2020): pp. 1290-1293, doi: 10.1109/EMBC44109.2020.9176162.
- [2] Sichtermann, T et al. "Deep Learning-Based Detection of Intracranial Aneurysms in 3D TOF-MRA." AJNR. American journal of neuroradiology vol. 40,1 (2019): 25-32. doi:10.3174/ajnr.A5911
- [3] Dai X, Huang L, Qian Y, Xia S, Chong W, Liu J, Di Ieva A, Hou X, Ou C. "Deep learning for automated cerebral aneurysm detection on computed tomography images. Int J Comput Assist Radiol Surg." (2020): 715-723. doi: 10.1007/s11548-020-02121-2. Epub 2020 Feb 13. PMID: 32056126.
- [4] Ueda, Daiju et al. "Deep Learning for MR Angiography: Automated Detection of Cerebral Aneurysms." Radiology vol. 290,1 (2019): 187-194. doi:10.1148/radiol.2018180901
- [5] Di Noto T. et al. "An Anatomically-Informed 3D CNN for Brain Aneurysm Classification with Weak Labels." In: Kia S.M. et al. (eds) "Machine Learning in Clinical Neuroimaging and Radiogenomics in Neuro-oncology." MLCN 2020, RNO-AI 2020. Lecture Notes in Computer Science, vol 12449. Springer, Cham. (2020): [https://doi.org/10.1007/978-3-030-66843-3\\_6](https://doi.org/10.1007/978-3-030-66843-3_6)
- [6] Tatsat R. Patel, Nikhil Paliwal, Prakhar Jaiswal, Muhammad Waqas, Maxim Mokin, Adnan H. Siddiqui, Hui Meng, Rahul Rai, Vincent Tutino, "Multi-resolution CNN for brain vessel segmentation from cerebrovascular images of intracranial aneurysm: a comparison of U-Net and DeepMedic," Proc. SPIE 11314, Medical Imaging 2020: Computer-Aided Diagnosis, 113142W (2020); <https://doi.org/10.1117/12.2549761>
- [7] Arbabshirani, Mohammad Fornwalt, Brandon Mongelluzzo, Gino Suever, Jonathan Geise, Brandon Patel, Aalpen Moore, Gregory. "Advanced machine learning in action: identification of intracranial hemorrhage on computed tomography scans of the head with clinical workflow integration." npj Digital Medicine. 1. (2018): 10.1038/s41746-017-0015-z.
- [8] M. Kolařík, R. Burget, V. Uher and M. K. Dutta, "3D Dense-U-Net for MRI Brain Tissue Segmentation," (2018) 41st International Conference on Telecommunications and Signal Processing (TSP), Athens, Greece, (2018): pp. 1-4, doi: 10.1109/TSP.2018.8441508.
- [9] Qin, Jason, Emmeric, Harry "Automatic Detection of Brain Aneurysm: segmenting CT scans using CNN" (2018) [http://cs230.stanford.edu/projects\\_fall\\_2018/reports/12446000.pdf](http://cs230.stanford.edu/projects_fall_2018/reports/12446000.pdf)
- [10] G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, (2017): pp. 2261-2269, doi: 10.1109/CVPR.2017.243.
- [11] S. Jégou, M. Drozdal, D. Vazquez, A. Romero and Y. Bengio, "The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation," 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA (2017): pp. 1175-1183, doi: 10.1109/CVPRW.2017.156.
- [12] Project MONAI Medical Open Network for AI <https://docs.monai.io/en/latest/index.html>
- [13] Brain MRI 3D classifier on MONAI [https://github.com/Project-MONAI/tutorials/tree/master/3d\\_classification/torch](https://github.com/Project-MONAI/tutorials/tree/master/3d_classification/torch)
- [14] <http://biomedic.doc.ic.ac.uk/brain-development/downloads/IXI/IXI-MRA.tar>
- [15] <https://vasognosis.com/>
- [16] Model Ensemble by MONAI <https://github.com/Project-MONAI/tutorials/blob/master/modules/modelsensembles.ipynb>
- [17] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, (2017): pp. 2980-2988, doi: 10.1109/ICCV.2017.322.
- [18] PyTorch Cross Entropy Loss function <https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html>
- [19] About Brain Aneurysm and Diagnosis <https://hospital.uillinois.edu/primary-and-specialty-care/neurology-and-neurosurgery/neurological-conditions-we-treat/brain-aneurysm/diagnosis-imaging>
- [20] Review: DenseNet — Dense Convolutional Network (Image Classification) <https://towardsdatascience.com/review-densenet-image-classification-b6631a8ef803>
- [21] Chablani, Manish <https://towardsdatascience.com/densenet-2810936aeabb>
- [22] TensorBoard <https://www.tensorflow.org/tensorboard>
- [23] Sci-kit Learn <https://scikit-learn.org/stable/modules/classes.html>

- [24] RandCropByPosNegLabel <https://docs.monai.io/en/latest/transforms.html#randcropbyposneglabel>
- [25] RandSpatialCrop <https://docs.monai.io/en/latest/transforms.html#randspatialcrop>
- [26] GridPatchDataSet <https://docs.monai.io/en/latest/data.html?highlight=gridpatchgridpatchdataset>

## Appendix

[1] DenseNet Output size for 121, 169 and 264 layer network architecture

Layers	Output Size	DenseNet-121( $k = 32$ )	DenseNet-169( $k = 32$ )	DenseNet-201( $k = 32$ )	DenseNet-161( $k = 48$ )
Convolution	$112 \times 112$	$7 \times 7$ conv, stride 2			
Pooling	$56 \times 56$	$3 \times 3$ max pool, stride 2			
Dense Block (1)	$56 \times 56$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	$56 \times 56$	$1 \times 1$ conv			
	$28 \times 28$	$2 \times 2$ average pool, stride 2			
Dense Block (2)	$28 \times 28$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	$28 \times 28$	$1 \times 1$ conv			
	$14 \times 14$	$2 \times 2$ average pool, stride 2			
Dense Block (3)	$14 \times 14$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 36$
Transition Layer (3)	$14 \times 14$	$1 \times 1$ conv			
	$7 \times 7$	$2 \times 2$ average pool, stride 2			
Dense Block (4)	$7 \times 7$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 16$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 24$
Classification Layer	$1 \times 1$	$7 \times 7$ global average pool			
		1000D fully-connected, softmax			

Table 1. DenseNet architectures for ImageNet. The growth rate for the first 3 networks is  $k = 32$ , and  $k = 48$  for DenseNet-161. Note that each “conv” layer shown in the table corresponds the sequence BN-ReLU-Conv.

Figure 8. Output size for DenseNet121 and bigger networks

[2] Classification - Training results using DenseNet121 with Data Augmentation for entire dataset (Model-1)

**Validation Accuracy: 95.79% , Test Accuracy: 96.76%**

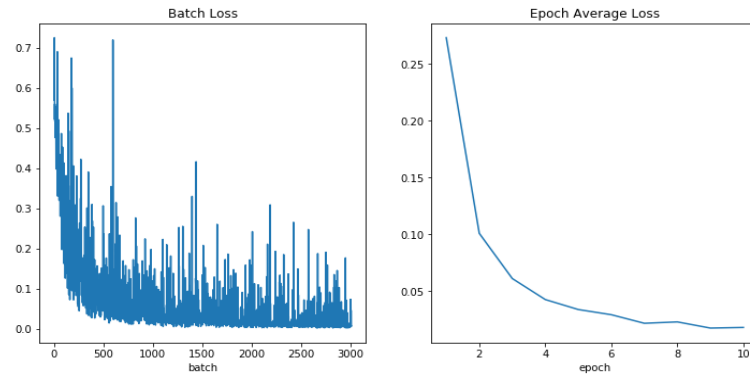


Figure 9. Loss Graph for Model-1

Validation Results			
Precision	Recall	F1-score	Support
0.98	0.97	0.97	851
0.90	0.93	0.92	289
		0.96	1140
0.94	0.95	0.94	1140
0.96	0.96	0.96	1140

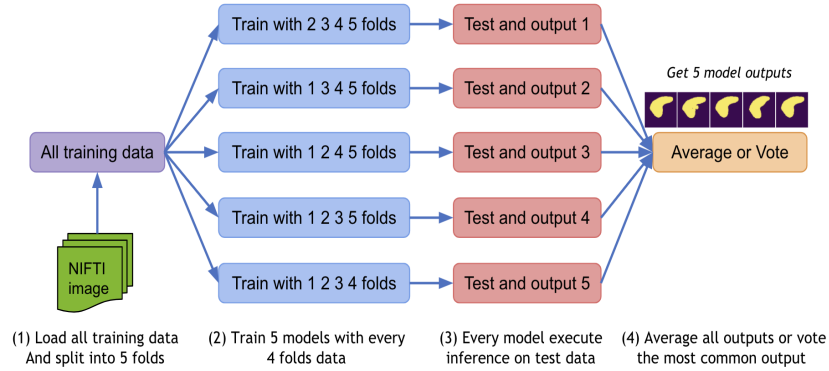
Table 3. Validation - F1 score and Recall

Test Results			
Precision	Recall	F1-score	Support
0.98	0.99	0.98	786
0.95	0.92	0.93	234
		0.97	1020
0.96	0.95	0.96	1020
0.97	0.97	0.97	1020



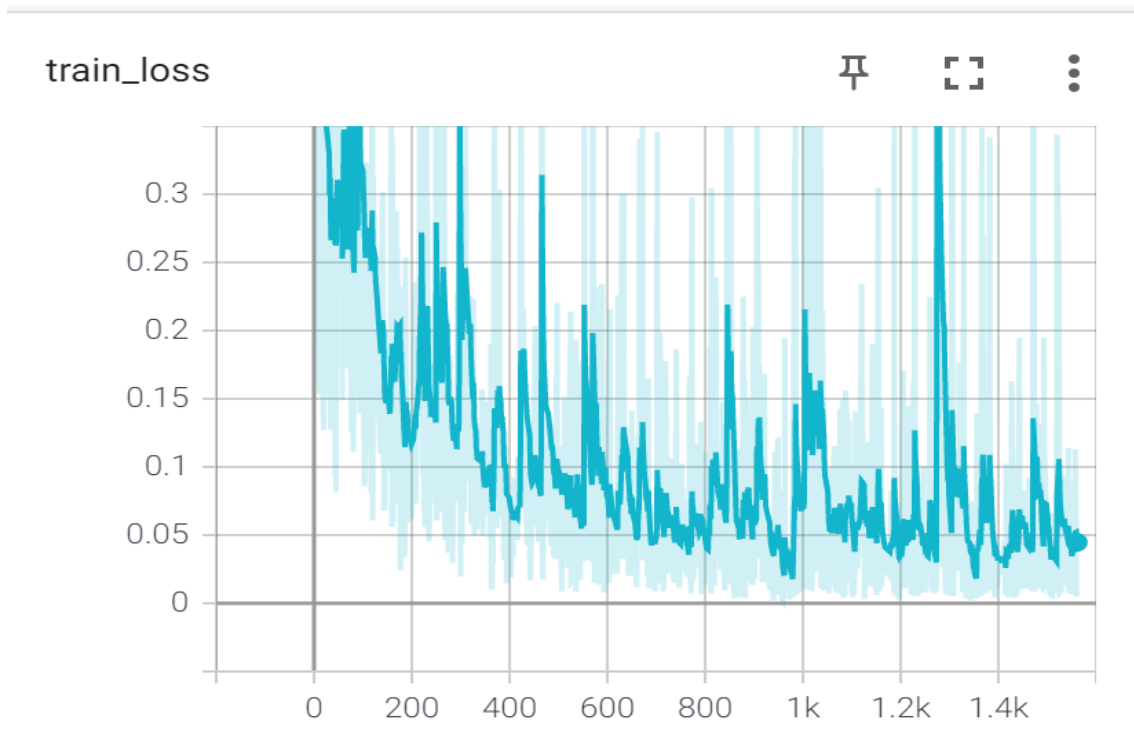
**Table 4.** Test - F1 score and Recall

[5] K-Fold Cross Validation process implemented



**Figure 10.** K-Fold Cross Validation by MONAI

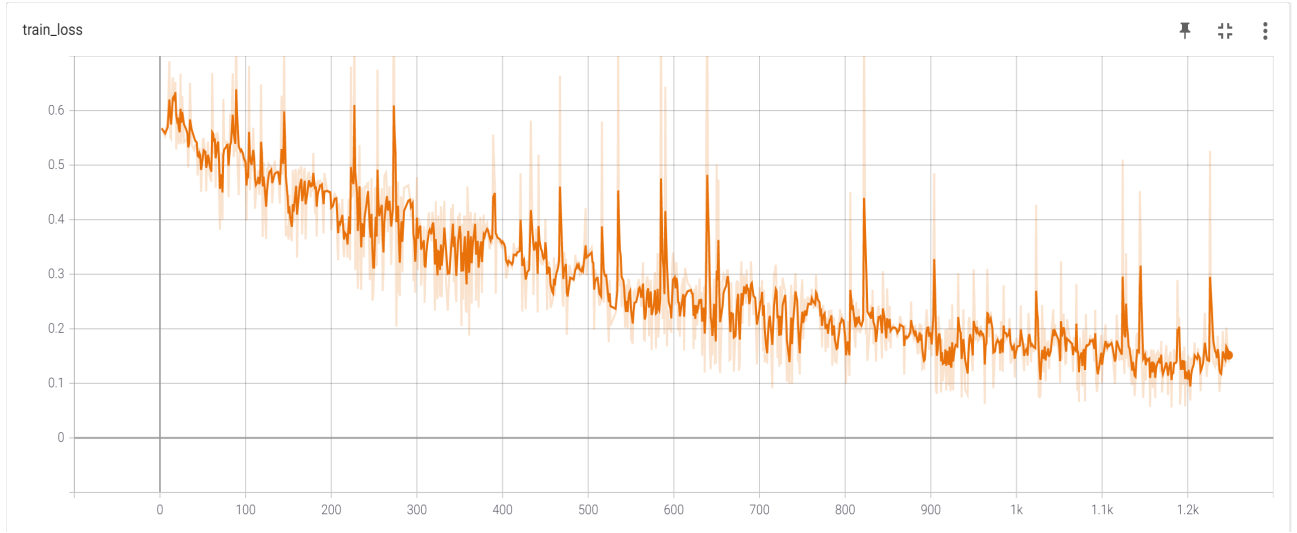
[6] Training Loss for Model-2 (DenseNet169)



**Figure 11.** Loss Plot for Model-2

[7] Training Loss for Model-3 (DenseNet121)





**Figure 12.** Loss Plot for Model-3

[8] K-fold Cross Validation results on the test data from the 5 models trained using 5-folds each with 2 epochs. We used RandomPos-NegLabel and RandomSpatial cropping mechanism for sampling.

RandCropByPosNegLabel	Predicted +ve	Total +ve	Predicted -ve	Total -ve
test_mri_accuracy_kf0_epoch0	66.00	82.00	75.00	75.00
test_mri_accuracy_kf0_epoch1	80.00	82.00	73.00	75.00
test_mri_accuracy_kf1_epoch0	82.00	82.00	68.00	75.00
test_mri_accuracy_kf1_epoch1	81.00	82.00	75.00	75.00
test_mri_accuracy_kf2_epoch0	82.00	82.00	63.00	75.00
test_mri_accuracy_kf2_epoch1	79.00	82.00	74.00	75.00
test_mri_accuracy_kf3_epoch0	81.00	82.00	75.00	75.00
test_mri_accuracy_kf3_epoch1	82.00	82.00	61.00	75.00
test_mri_accuracy_kf4_epoch0	80.00	82.00	49.00	75.00
test_mri_accuracy_kf4_epoch1	82.00	82.00	61.00	75.00

**Figure 13.** MRI Accuracy of all 5 models using RandomPosNegLabel cropping mechanism

RandomSpatialCrop	Predicted +ve	Total +ve	Predicted -ve	Total -ve
test_mri_accuracy_kf0_epoch0	64.00	77.00	75.00	80.00
test_mri_accuracy_kf0_epoch1	76.00	77.00	74.00	80.00
test_mri_accuracy_kf1_epoch0	78.00	78.00	68.00	79.00
test_mri_accuracy_kf1_epoch1	73.00	75.00	75.00	82.00
test_mri_accuracy_kf2_epoch0	78.00	78.00	68.00	79.00
test_mri_accuracy_kf2_epoch1	74.00	77.00	76.00	80.00
test_mri_accuracy_kf3_epoch0	78.00	79.00	74.00	78.00
test_mri_accuracy_kf3_epoch1	77.00	78.00	65.00	79.00
test_mri_accuracy_kf4_epoch0	76.00	76.00	59.00	81.00
test_mri_accuracy_kf4_epoch1	79.00	79.00	67.00	78.00

**Figure 14.** MRI Accuracy of all 5 models using RandomSpatial cropping mechanism