Intracranial Hemorrhage Detection and Classification with CNNs

By Mukul (21052082) and Ankit (2105179)

Problem Statement

- Intracranial hemorrhage (ICH) refers to bleeding that occurs within the skull. This bleeding can happen for various reasons, such as trauma, ruptured blood vessels, or medical conditions that affect blood clotting or blood vessel integrity.
- Use the dataset provided to build a system that automatically detects any hemorrhage and the type of hemorrhage if present
- Type of hemorrhage are: epidural, intraparenchymal, intraventricular, subarachnoid and subdural

About The Dataset

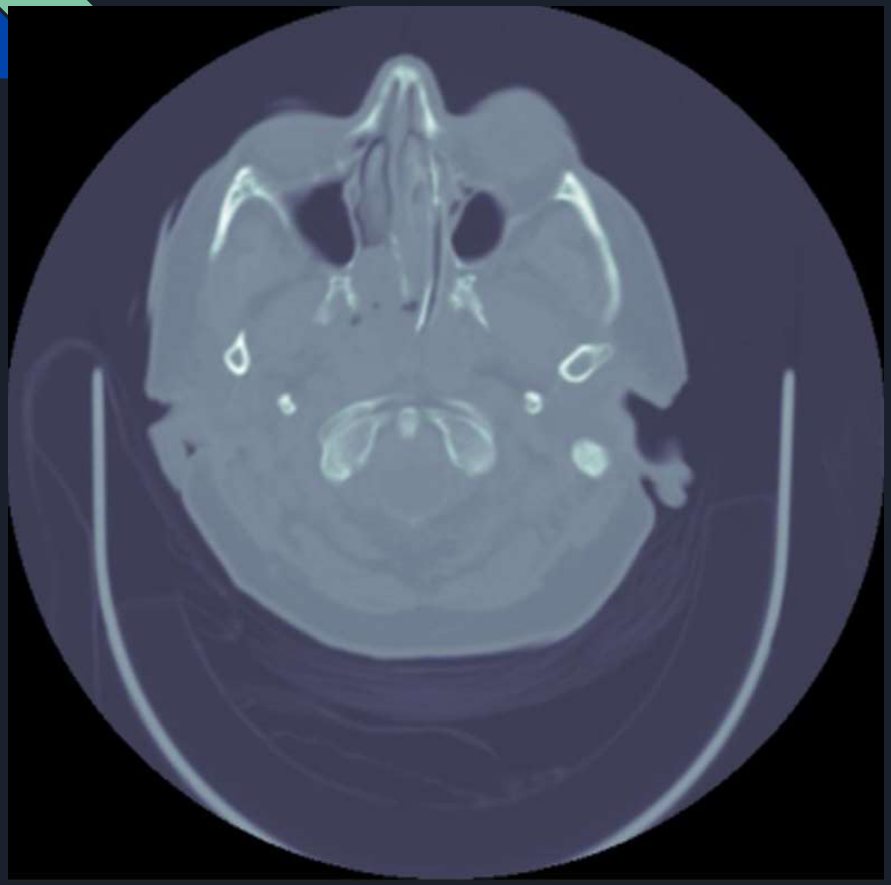
- The dataset is provided by Radiological Society of North America (RSNA) hosted on kaggle.
- It contains a total of 847k images, out of which 753k are labelled, rest are unlabelled. The total coming out to be 485.94 Gb.
- The images are of cranial CT scans, present in DICOM (digital imaging and communications in medicine) image format, which is commonly used in radiology.
- Due to hardware limitations, only 16Gb data was used

Challenges

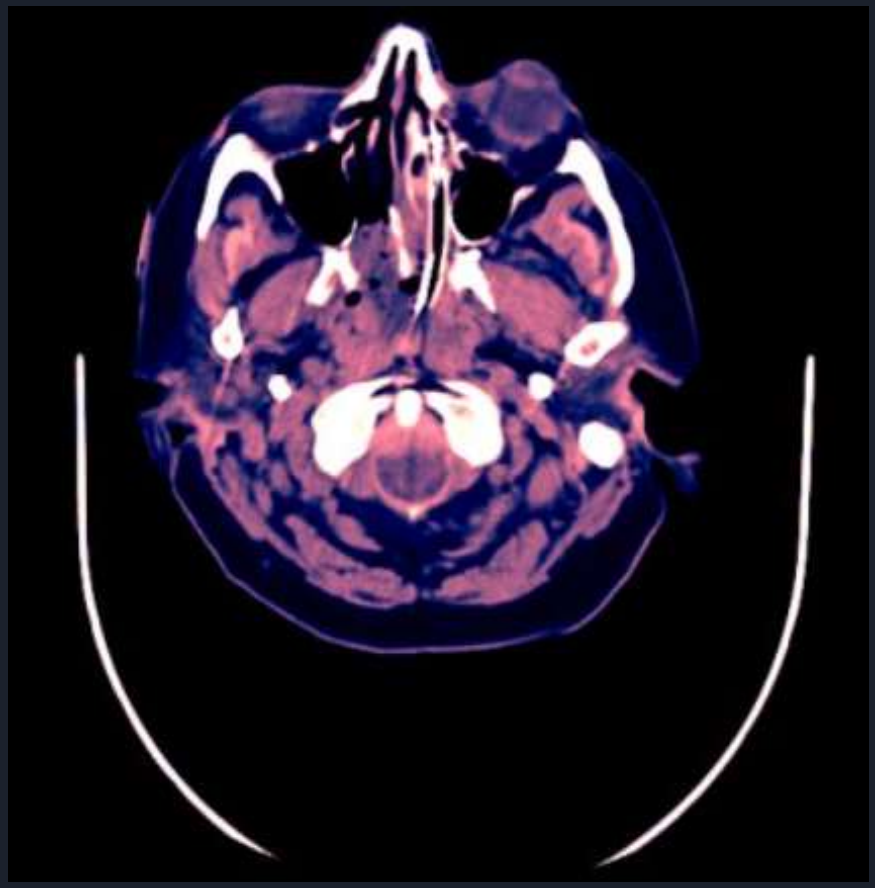
- Format of CT scans are provided in DCM format
- In order to look at CT images, we need to look at windows of different Hounsfield Unit (HU) ranges as different ranges highlight different tissues of the body.
- The 3 different windows we used are subdural (space between the skull and the brain), brain window, and soft tissue window and we stacked on each other, resembling the RGB channels.

CT Number Ranges in Hounsfield Units (HU)	
Bone	+1000
White matter	+20 to 30
Muscle	+20 to 40
Gray matter	+30 to 40
Hemorrhage	+65 to +95
CSF (water)	0
Fat	-30 to -70
Air	-1000

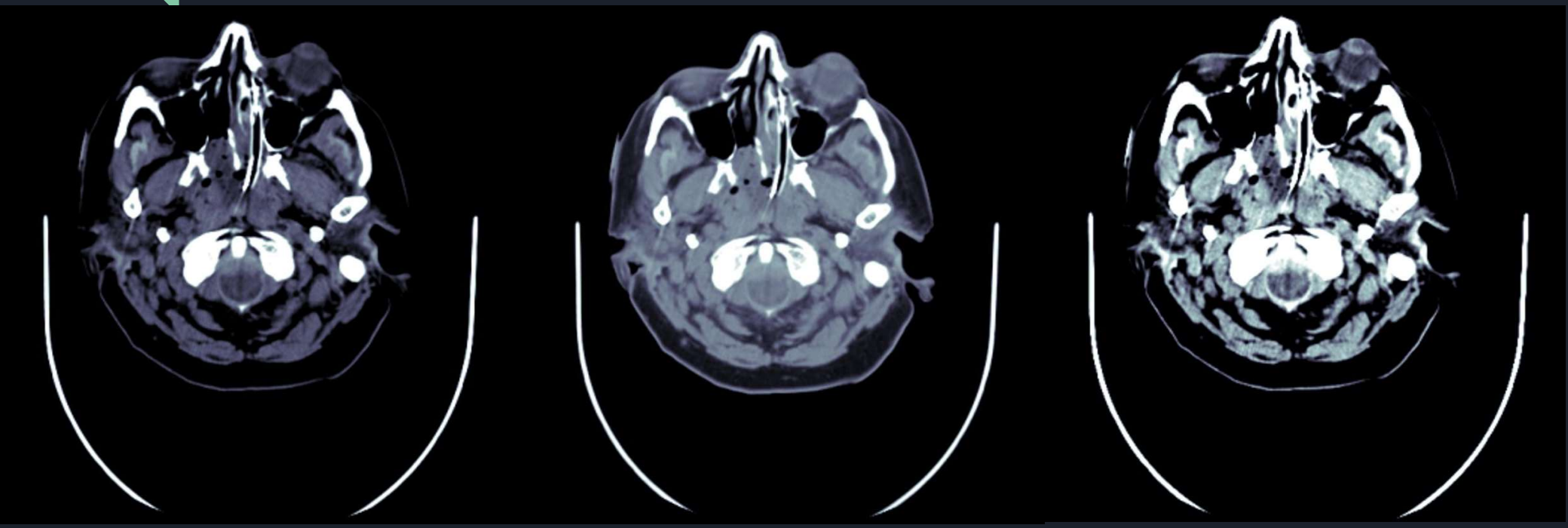
DCM image without windowing



DCM image after stacking windows



Images of Subdual, Soft and Brain windows



Technology Used

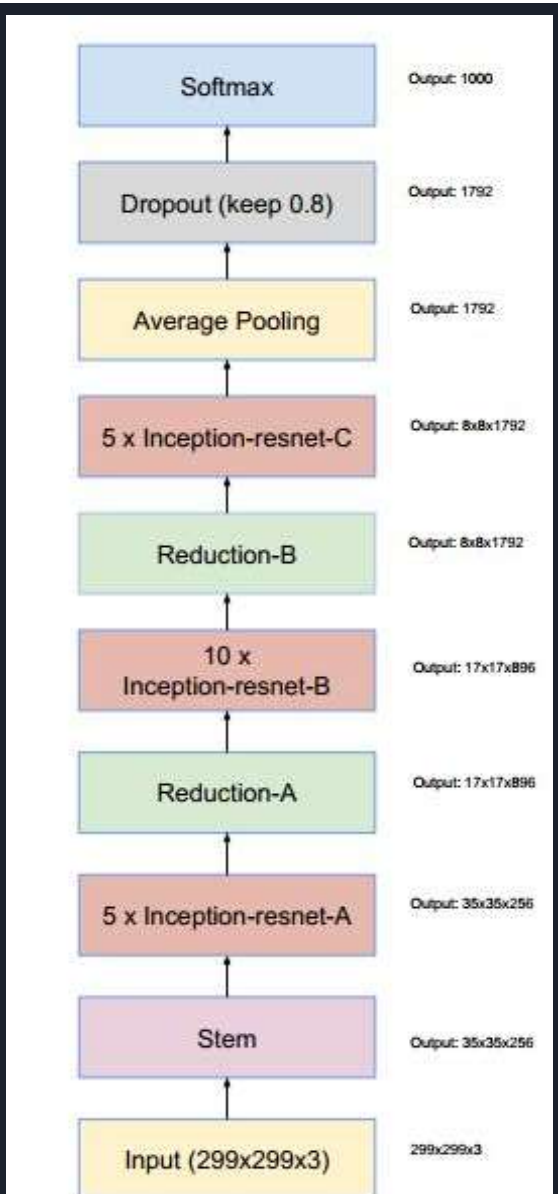
- Pytorch
- Pandas
- Numpy
- Matplotlib
- SKLearn
- Loss Function used in Binary Cross Entropy Loss
- Optimizer used is Adam with learning rate 10^-5

Code

Provided in Kaggle

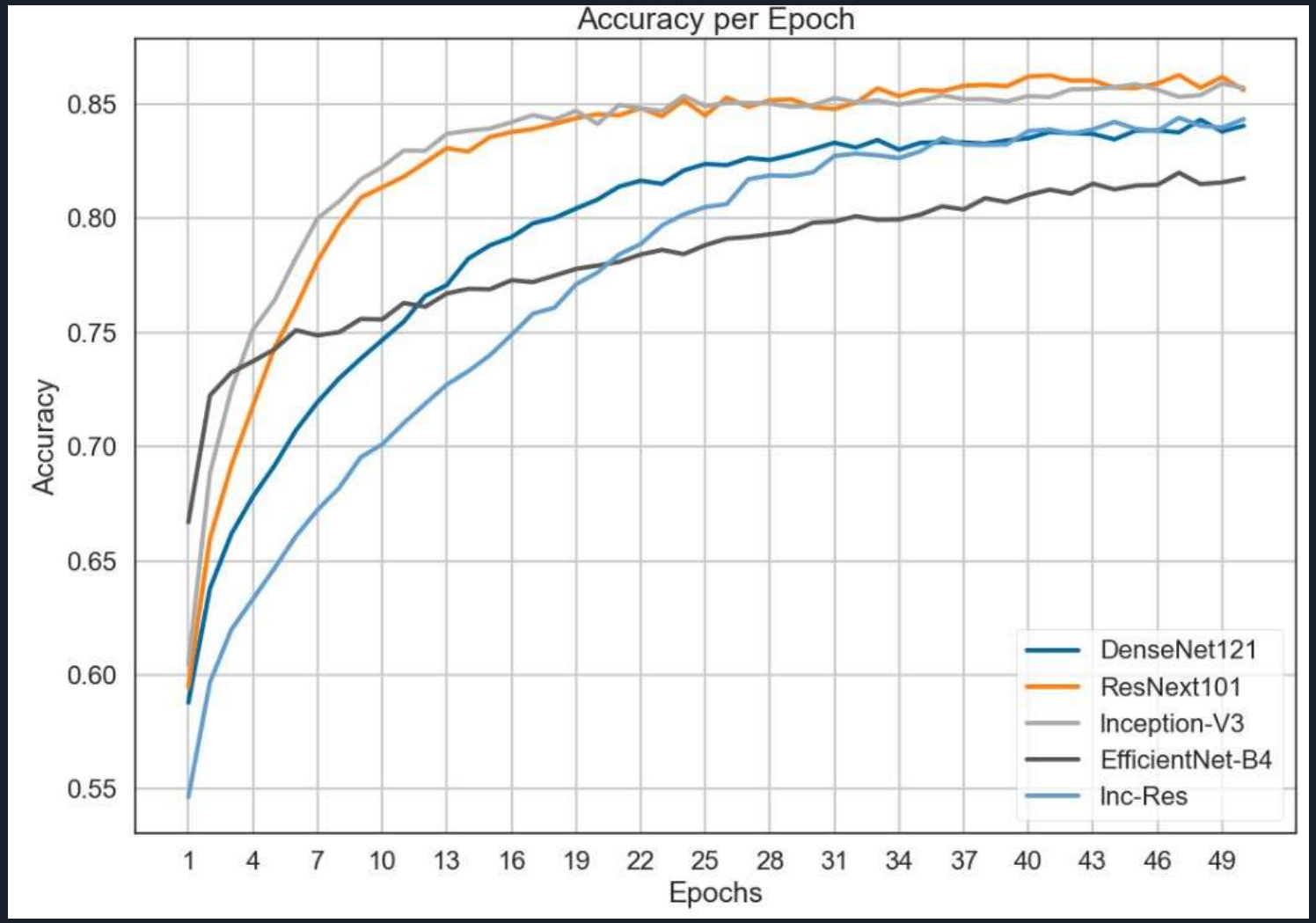
Inception-ResnetV2

Schema for
Inception-Resnet-V2



Outcome

- Several different models (Inception,ResNet,EfficientNet,Densenet) were used, out of which inceptionV3 performed the best with an accuracy of 85%
- Then a custom model was built which modified Inception V4 to include skip connections to get Inception-Resnet-V2
- There was no increase in performance from added skip connection to Inception V4


Accuracy per Epoch
Accuracy
0.85
0.80
0.75
0.70
0.65
0.60
0.55
1
4
7
10
13
16
19
22
25
28
31
34
37
40
43
46
49
Epochs
DenseNet121
ResNext101
Inception-V3
EfficientNet-B4
Inc-Res

Thank you