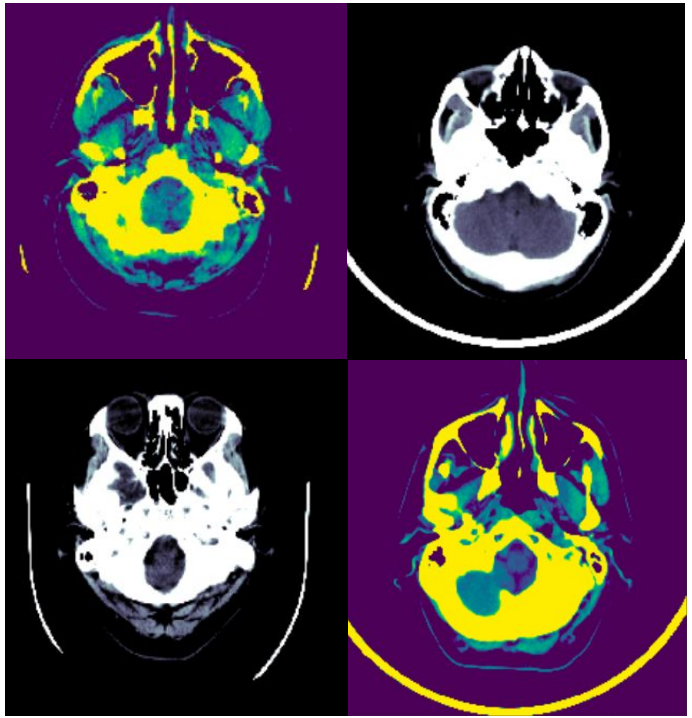
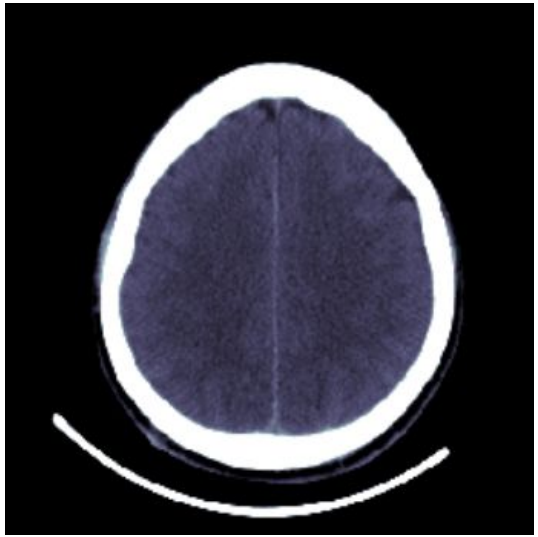
Detecting Brain Hemorrhage

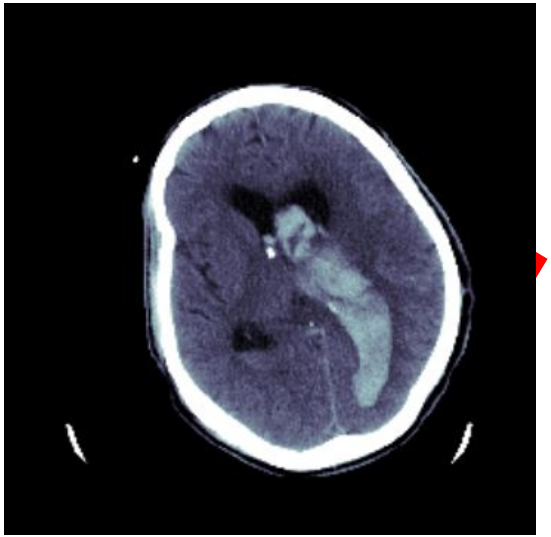
Ramin Ostad
Flatiron Data Science Fellowship



Problem

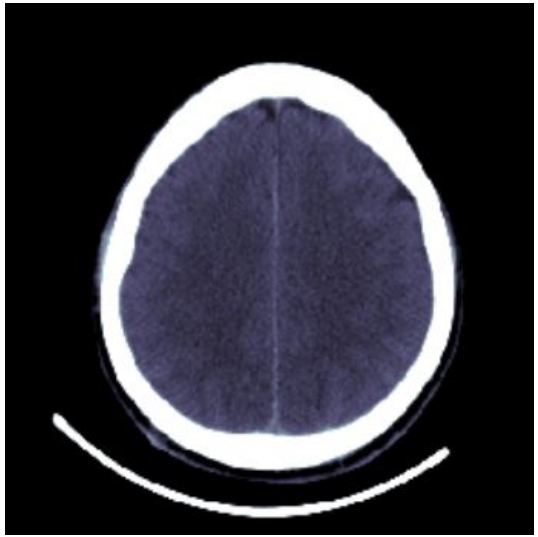
“Hemorrhage or no hemorrhage?” That is the question...

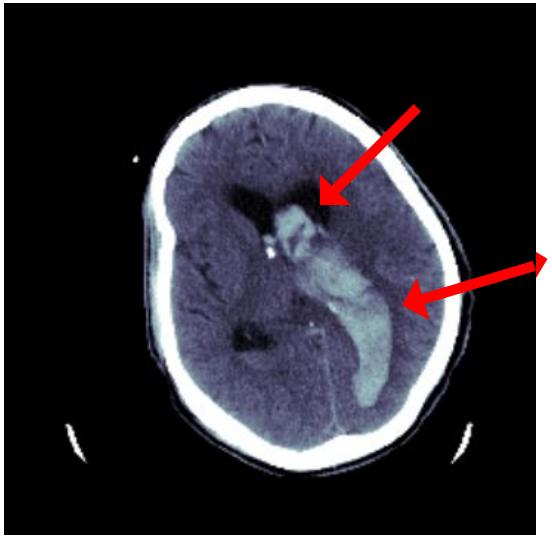




Problem

“Hemorrhage or no hemorrhage?” That is the question...





Approach

We will train a convolutional neural network to identify signs of brain hemorrhage using 600K+ previously labeled brain scanned images

Image Preprocessing → Convolutional Neural Network → Prediction

Data

- CT Scan images enter the model as DICOM files
- We extract a density level that allows our model to ‘see’ brain, blood and bones

Model

- Model will be built in tensorflow using the Keras wrapper
- We will use various layers to detect similarities across same-class images

Hemorrhage Detection

- The model can be used to predict whether a new patient has brain hemorrhage

CT Scans of the Brain

CT scans combine many individual x-rays to produce a more detailed image of a specific area of the body

What is a CT scan?

A Computed Tomography scan, otherwise known as a CT or CAT scan, is a process whereby many x-rays are taken of an body part from different angles and combined to form a single more detailed image of that body part

How are CT scans taken?

CT scans are taken using a CT scanner. Typically a CT scanner consists of a tube which houses the x-ray detector. The x-ray detector moves around the body in the tube as the body as the body is moved through the tube.

Siemens CT Scanner at the University of Virginia

Prevalence of Hemorrhage in Training Data

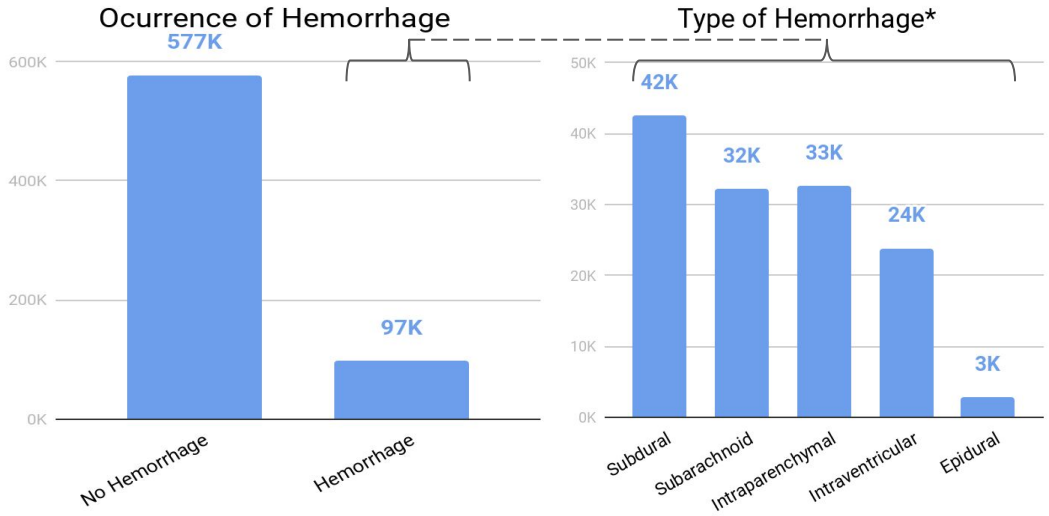
7% of training images show hemorrhage of which the most common type is subdural hemorrhage which occurs in 43% of cases

Data Overview

Training Images
674K

Training Image Size
332 GB

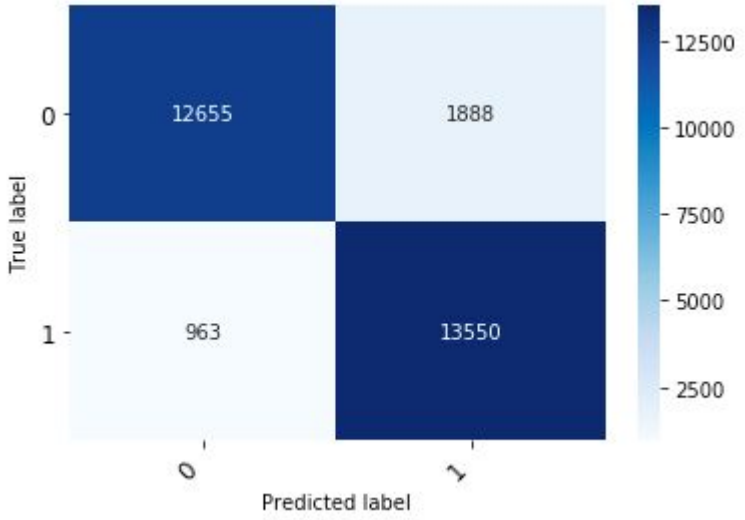
Data Source
Radiological Society of North America (RSNA) and Kaggle



*Images with hemorrhage can have multiple types of hemorrhage

Predicting Hemorrhage

The model can predict hemorrhage with ~90% accuracy



As of now, our network can correctly classify ~90% of CT scans

Can radiologists take long vacations? Probably not yet, but the tool can be a good double-check or first response when radiologists are not on call

Uses of the Classifier

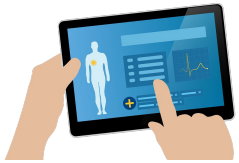
The classifier can be used in various situations where cost and speed of hemorrhage detection are a factor

Disaster Response

Models can often be deployed in a fraction of the time of human resources

Rural and High Cost Areas

In areas areas that are currently underserved due to the cost constraints, a model can provide a low cost alternative



Radiologist's Assistant

Models can serve as assistants to radiologists by providing a low-cost double check of results

Catching Fringe Cases

With improved training, the model would have the ability to classify fringe cases that may be hard to predict otherwise

Future improvements

Avenues for future improvements include addition of dropout layers as well as use of portions of pre-trained models

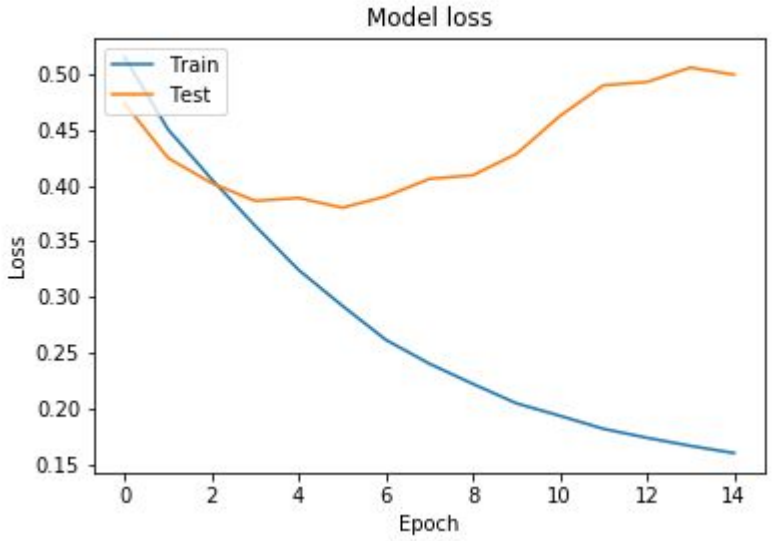
Dropout layers to reduce overfitting
at present we experience overfitting to the training data. Addition of dropout layers

Bagging of various models
Models of varying performances could be combined to reduce overfitting while increasing accuracy

Addition of pre-trained models
Pretrained models such as VGG16 and InceptionV3 have been shown to be effective in CT scan classification

Cost function changes
In application, different scenarious will have different cost distributions. The model should incorporate this cost function during training

Sample Training Run Showing Overfitting



Thank You

Special thanks to the amazing professors and coaches!!

Selecting the Best Density Window

During image preprocessing we will choose the relevant window density from our images that captures blood, bones and brain matter

What is a DICOM file?

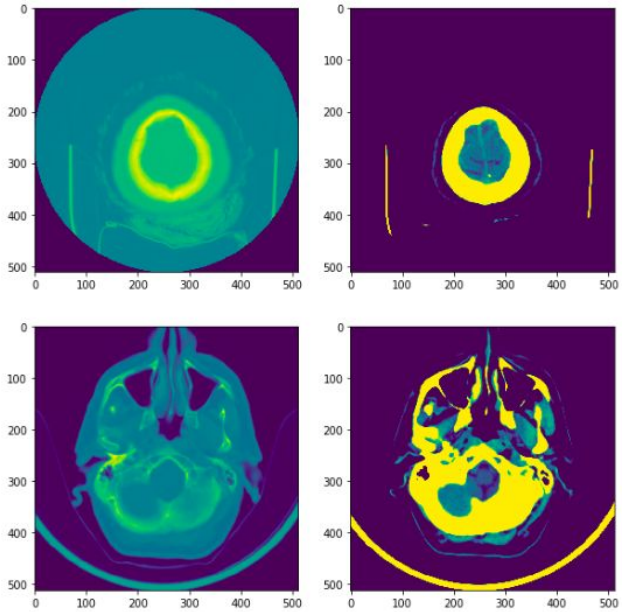
CT scans used in this model are stored as Digital Imaging and Communications in Medicine (DICOM) files. These files contain an image along with various metadata

What is windowing?

CT images stored DICOMs allow us to see a wide range of density levels. In our image processing, we will extract only the density window that capture blood, bones and brain

Before Windowing

After Windowing



Model Setup

The Convolutional Neural Network (CNN) uses various layers to identify key features in the image

Convolution layers apply conversions or filters across the image that will help detect features such as horizontal or vertical lines

Activation layers allow the model to decide which image features should be considered and which ignored

Pooling layers condense the image by extracting the most relevant features of each group of pixels

Dense layers condense information into fewer categories that we can use for classification

Model Layers

```
Model: "sequential_9"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_18 (Conv2D)           (None, 510, 510, 32)      320
_________________________________________________________________
activation_27 (Activation)   (None, 510, 510, 32)      0
_________________________________________________________________
max_pooling2d_18 (MaxPooling (None, 255, 255, 32)      0
_________________________________________________________________
conv2d_19 (Conv2D)           (None, 253, 253, 32)      9248
_________________________________________________________________
activation_28 (Activation)   (None, 253, 253, 32)      0
_________________________________________________________________
max_pooling2d_19 (MaxPooling (None, 50, 50, 32)        0
_________________________________________________________________
flatten_9 (Flatten)          (None, 80000)             0
_________________________________________________________________
dense_27 (Dense)             (None, 50)                4000050
_________________________________________________________________
activation_29 (Activation)   (None, 50)                0
_________________________________________________________________
dense_28 (Dense)             (None, 6)                 306
_________________________________________________________________
dense_29 (Dense)             (None, 1)                 7
=================================================================
Total params: 4,009,931
Trainable params: 4,009,931
Non-trainable params: 0
_________________________________________________________________
```