\*Note: the dataset contains many thousands of images, but each image is part of a slice of a single CT scan, and there can be as many as 20 slices from one scan. As such, many solutions that I surveyed may train on combined slices from the same scan, on single individual slices, or use a combination of both. All the papers/code listed here were trained on the same dataset, the RSNA Intercranial Hemorrhage (ICH) CT dataset.

## Existing Research

1. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7582288/> This paper comes from a group of researchers from Budapest and claims to be within the top 2% of submissions in the RSNA ICH Kaggle competition. Their model consists of a Convolutional Neural Network (CNN) which feeds its outputs into a Long Short-Term Memory (LSTM) network. The CNN takes in the raw input, as a DICOM image. They also use a “feature selection” method alongside the CNN which feeds the selected features into the LSTM network.
   1. **Preprocessing**: Each 512x512p image is condensed into 256x256p. Additionally, following radiologist advice, the researchers also applied 3 intensity windows to each CT slice to augment different types of tissue: the brain, subdural, and soft tissue windows. Then, these 3 (greyscale) images are combined into a 3-channel image and fed into the CNN. They also standardize all images prior to training. Additionally, they perform random transformations on the data, such as brightness scaling, shifting, rotation, and scaling.
   2. **Model:** The researchers divide training into 3 stages: (1) train the CNN to predict on single slices (starting from the pretrained ResNeXt model); (2) use some feature selection method (the authors consider 2 methods) to select features from the CNN’s penultimate layer; (3) extract features from slices of an entire CT scan using the CNN + feature extraction, then feed these into a LSTM network and train it.
   3. **Final loss**: 0.04989 on the test dataset
   4. Their public code is available at <https://github.com/warchildmd/ihd>
2. <https://arxiv.org/ftp/arxiv/papers/2102/2102.04869.pdf> This paper comes from researchers at the University of Toronto and St. Michael’s Hospital in Toronto, Canada.
   1. **Preprocessing:** First, each CT slice has its center and width adjusted (not specified exactly how) and then the brain, subdural, and soft tissue windows are applied to split each image into 3 channels.
   2. **Model:** The model trains on single individual slices while incorporating information from other slices in the CT scan. First, the 3-channel images are sent to a ResNeXt-50 and ResNeXt-101 (pretrained with ImageNet weights), which are first fine-tuned to predict ICH (an ensemble of 2 networks reduces variance). Next, the model finds the predictions of nearby slices of the same scan, and then this data is passed to an ensemble of LightGBM, CatBoost, and XGBoost whose results are averaged, and produce a final prediction for ICH in the slice. If a probability assigned to a particular subtype of ICH reaches a certain threshold, then the slice is determined to contain that subtype of ICH. The thresholds were determined using a Bayesian optimizer.
   3. **Final loss:** AUC of 0.984 on the test dataset.
3. <https://www.sciencedirect.com/science/article/pii/S2213158221002291> This paper comes from a group of researchers who won 1st place in the RSNA Intracranial Hemorrhage Kaggle competition.
   1. **Preprocessing:** The authors first apply the brain, subdural, and soft tissue windows to each image slice, and then scale each image to an 8-bit greyscale image instead of 16-bit.
   2. **Model**: First, a CNN classifier is trained on each slice (transformed into 3-channels) to give a prediction of ICH. Additionally, the CNN serves as a feature extractor, and the output of its last convolutional layer is taken as extracted features. The CNN classifier is run on each slice in a CT scan, and all these features are grouped into a 3D tensor. In the next stage of training, this tensor is sent to a bidirectional RNN with a GRU unit which produces a refined ICH prediction using all information in the scan. Finally, the refined prediction and the initial CNN prediction are sent to another RNN with a GRU unit to combine the predictions and produce a final output (which predicts ICH on every slice in the CT scan). The researchers used ResNeXt-101, Densenet169, an Densenet121 as pretrained weights for the CNN.
   3. **Final loss**: An AUC of about 0.99 for each subtype of ICH on the test dataset.
   4. Their public code is available here: <https://github.com/Scu-sen/1st-RSNA-Intracranial-Hemorrhage-Detection>. This code contains all data preparation and training code in Python 3 used for the competition.

## Discussion

Surveying some existing research was able to help me get a sense of the type of model I should build in order to be successful, however many techniques used are cutting-edge and I don’t understand how to use some of them, therefore I would like to start preliminary exploration + model-building using techniques I know and understand first.

Every single paper I read had a model that first split each image into a 3-channel that emphasizes different types of tissue. This is also how radiologists tend to diagnose ICH, so this is a critical preprocessing step that I will need to incorporate. All of them use a CNN with pretrained weights, which I also should use as is standard with image data. All models also used some sort of multi-stage system, where one portion of the model is trained and then fed into the next portion. So, my model will tentatively possess these three attributes: preprocessing into 3-channels, using a pretrained CNN, and using a multi-stage model.