Lung Cancer CT-Scan Classification

Sanjeev Narasimhan

Introduction

This report aims to provide an introductory multiclass classification of lung cancer types. Analysis is done using a convolutional neural network to classify CT-Scans into 1 of 3 lung cancer types, or as having no cancer.

Train/ Validation/ Test Split: The dataset came with a 70/10/20 TVT split

- 70% Train –> 613 files
- 10% Validation –> 72 files
- 20% Test –> 315 files

Label: each image has an associated label that classifies the image of either having 1 of 3 lung cancer types or being normal (having no cancer)

Adenocarcinoma

- Adenocarcinoma of the lung: Lung adenocarcinoma is the most common form of lung cancer accounting for 30 percent of all cases overall and about 40 percent of all non-small cell lung cancer occurrences
- Adenocarcinomas are found in several common cancers, including breast, prostate and colorectal
- Adenocarcinomas of the lung are found in the outer region of the lung in glands that secrete mucus and help us breathe
- Symptoms include coughing, hoarseness, weight loss, and weakness.

Large cell carcinoma

- Large-cell undifferentiated carcinoma: Large-cell undifferentiated carcinoma lung cancer grows and spreads quickly and can be found anywhere in the lung
- This type of lung cancer usually accounts for 10 to 15 percent of all cases of NSCLC
- Large-cell undifferentiated carcinoma tends to grow and spread quickly

Squamous cell carcinoma

- Squamous cell: This type of lung cancer is found centrally in the lung, where the larger bronchi join the trachea to the lung,or in one of the main airway branches
- Squamous cell lung cancer is responsible for about 30 percent of all non-small cell lung cancers, and is generally linked to smoking.

And the last folder is the normal CT-Scan images

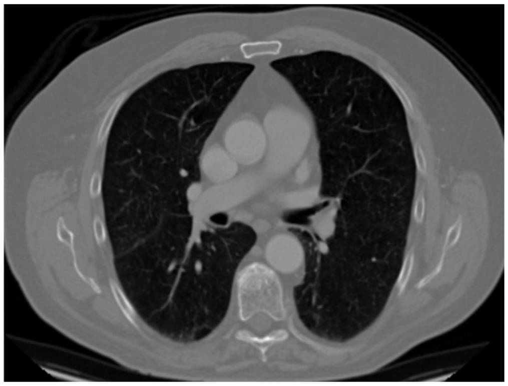
Because of the small size of the data, some inconsistencies within the actual images, and bias in the frequencies of certain classifications, this analysis is not a particularly strog classification tool. However, with further efforts using GradCAM, this model could provide a basis to understand the characteristics of images that determine the classifications.

Analysis

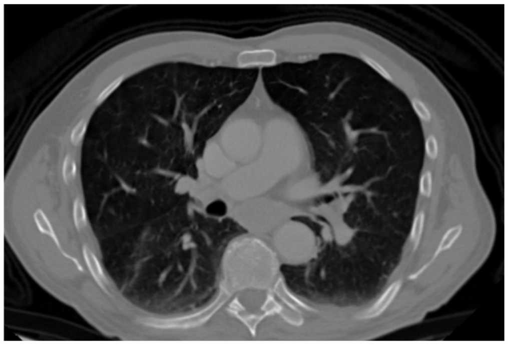
Dataset Breakdown

- Train Dataset:
  - 195 files labeled as adenocarcinoma
  - 115 files labeled as large cell carcinoma
  - 148 files labeled as normal
  - 155 files labeled as squamous cell carcinoma
- Vaidation Dataset:
  - 23 files labeled as adenocarcinoma
  - 21 files labeled as large cell carcinoma
  - 13 files labeled as normal
  - 15 files labeled as squamous cell carcinoma
- Test Dataset:
  - 120 files labeled as adenocarcinoma
  - 51 files labeled as large cell carcinoma
  - 54 files labeled as normal
  - 90 files labeled as squamous cell carcinoma

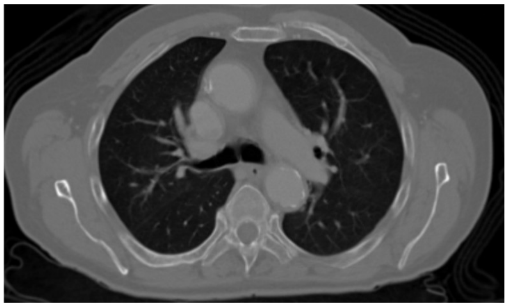
CT-Scan Examples



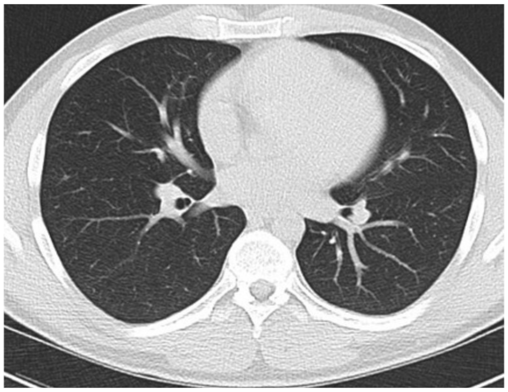
Adenocarcinoma



Large Cell Carcinoma



Squamous Cell Carcinoma



Normal – No Cancer

Methods

Model Structure

- Convolutional Neural Network with 20 layers
- Consists of:

- 5 convolutional layers –> each using relu activation and L2 regularization to prevent overfitting
- 5 max pooling layers (after each convolutional layer)
- Batch Normalization layers added every other convolutional layer
- Flattening Layer
- 4 dense layers –> each using relu activation and L2 regularization
- Batch Normalization and Dropout layer added to second dense layer to prevent overfitting
- 1 dense layer with 4 outputs (for the 4 possible classifications) and softmax activation

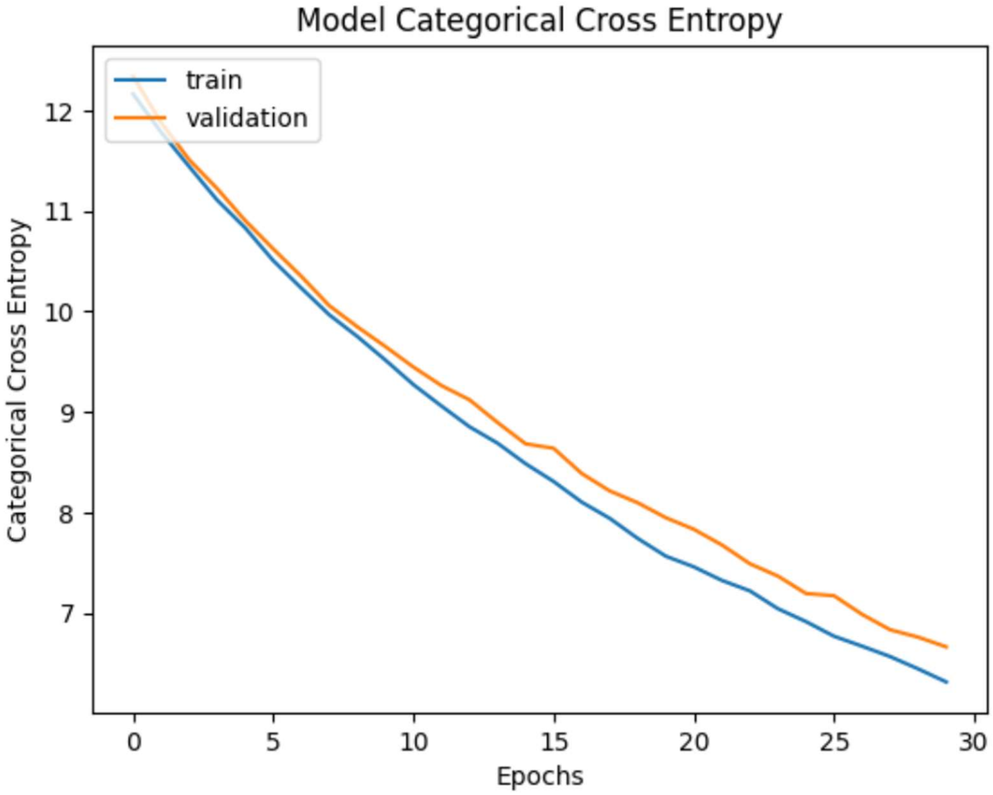
Training the Model

- The model was trained over 30 epochs
- The model was trained to minimize the categorical cross entropy
- The optimizer Adam was used with a learning rate of 0.0001

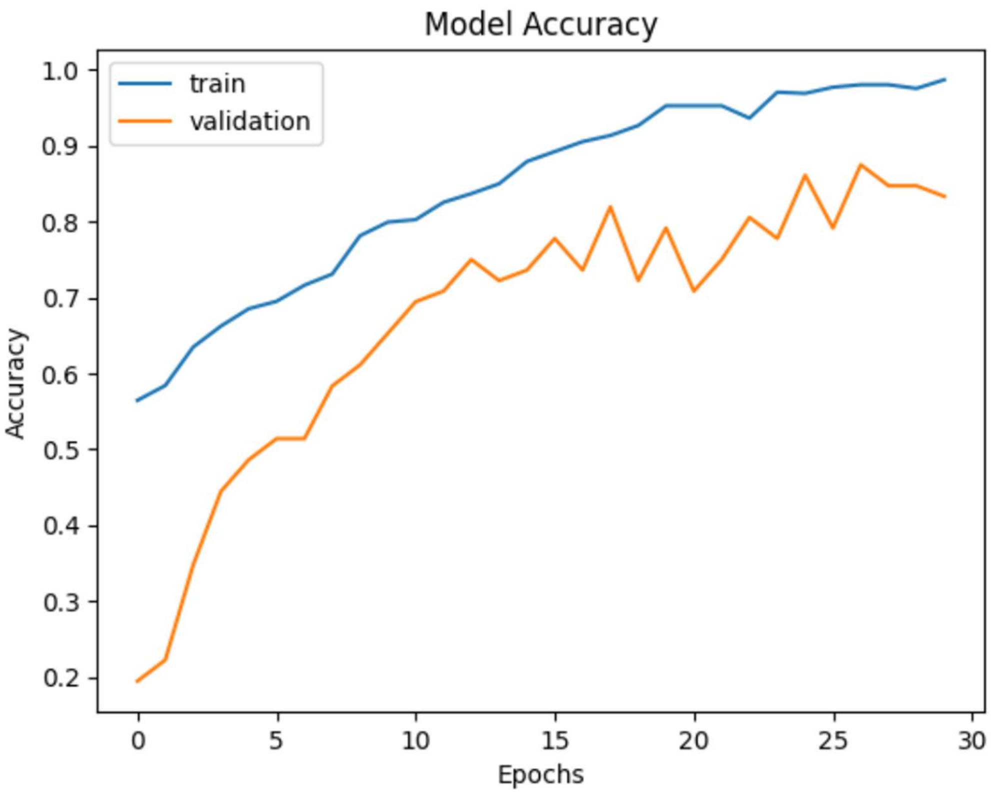
Evaluating Model Performance

- Categorical Cross Entropy calculated for Train, Validation, and Test data
- Accuracy calculated for Train, Validation, and Test Data

Results



Graph Plotting the Training and Validation Loss through each Epoch



Graph Plotting the Training and Validation Accuracy through each Epoch

The metrics as shown during the training process with the train and validation data would lead to the belief that the model would perform somewhat well. However, when we evaluate the model on the test data we can see that this is not necessarily the case. In fact the model performed quite poorly with unseen data.

Gauging Metrics

Convolutional NN - Train Categorical Cross Entropy: 6.3152 - Validation Categorical Cross Entropy: 6.6640 - Test Categorical Cross Entropy: 6.9112

- Train Accuracy: 0.9869
- Validation Accuracy: 0.8333
- Test Accuracy: 0.6698

Evaluating Process and Model

During the training process, I used the validation data to fine tune the structure of the model to create an architecture that would work best with unseen data and went through many iterations of adding/ removing batch normalization, dropout, and mooling layers. I also tried changing the dropout chance in certain layers to reduce overfitting, but but at the cost of having a much lower accuracy. The CNN architecture that I ended with was the one that produced the best metrics with the validation data averaging between ~0.8 and 0.86 accuracy, but looking at the metrics from evaluating the model with the testing data shows that these improvements didnt transfer over well at all.

With a test accuracy score of ~0.6698, our model is able to correctly classify across the 4 possible options (adenocarcinoma, large cell carcinoma, squamous cell carcinoma, and normal) ~67% of the time which is very bad. I tried to get the Classification Report to work, that would return the following metrics for each possible classification: precision, recall, and f1-score. In the context of our data asside from accuracy, we would want to focus most on the recall and f1-score. The recall metric would tell us how often our model is correct for positive cases. Since this model is determining whether an individual has cancer based on the CT-Scan image, we want to ensure that our model does not missclassify somewho who has cancer as not having cancer. the precision would tell us for all of our positive predictions, how many times are we correct, which is useful because we also wnt to know how many times our positive predictions are correct. Finally, the f1-score would combine those two metrics which is a very string measurement metric for the context of our data.

Reflection

Unfortunately I wasnt able to do all that I wanted to do with this assignment. I spent a lot of time messing around with the architecture of the CNN to make sure that the test data would also have a strong performance. I was hoping that if the metrics had a well enough performance, I could implement Grad-CAM to visualize the characteristics of the images that the model looked at. However, the the model did not perform anywhere near what I was expecting given the performance with the testing data. In the future I would want to consider data augmentation to ensure that the different splits had a good ratio of adenocarcinoma, large cell carcinoma, squamous cell carcinoma, and normal. With this the model would be trained using a reasonable ratio of classifications for the train and validation datasets, which would hopefully translate better with the testing set. I think it might be useful to also add some padding to maintain the size of the image while it is being processe, in case some features were ignored. While it would take some time, I think playing around with the architecture more would help me in determining the optimal strucutre and hyperparameters to use.