Adam Villarreal

Machine Learning Capstone

Domain Background

This is a project is in the research field of Deep Learning, more specifically Convolutional Neural Networks(CNNs). Convolutional Neural Networks are now being widely used for image classification, which has been used in production especially. One example of production use is facial recognition systems that our hardware products use, which adds a level of security that is above a password code system. The research paper that inspired my curiosity can be found here.

Problem Statement

From above, CNNs are actively being used and improved for image classification, which has now entered into the world of Medicine. From my own conclusions, it seems to me that the use of CNNs has the opportunity to quicken the process of diagnoses in patients so that treatment can be expedited. Some might argue that the need for doctors will decrease and the risk of misclassification is just too high to rely on machines. I do not see it in this way since instead of wanting to remove doctors, artificial intelligence should be used as a diagnostic tool for doctors. CNNs are known to outperform doctors in diagnoses based on images, an example of this is here. I would like to investigate how to create a Convolutional Neural Network to provide an image classifier that will have high accuracy and that will minimize the number of false negatives, which are images that contain the sickness that is classified as not having the sickness.

Datasets and Inputs

The dataset I acquired contains labeled images of chest X-rays that showed either the patient had pneumonia or did not. There are 5,863 total images and are separated into separate folders for training, validation, and testing. The data can be downloaded from Kaggle here. If you want to see who gathered the data and their research, click here.

Info about the dataset:

- Most of the images are greyscaled, but some are in RGB format
- The dimensions of the images are not consistent
- The dataset is already separated into a train, test, and validation datasets
- The training dataset is not balanced
    - 25% of the training dataset is composed of normal images
    - 75% of the training dataset is composed of images with pneumonia
- The testing dataset is not balanced
    - 38% of the testing dataset is composed of normal images
    - 62% of the testing dataset is composed of images with pneumonia

Solution Statement

I will use this data from above to create different CNNs for the purpose of diagnosing pneumonia. I will then compare these models to choose the best one based on accuracy and decreasing the number of false negatives. I will then use AWS to

deploy the best model and making it accessible through a Web-App that would allow anyone to input chest X-rays to classify the images and return whether the image contained pneumonia or otherwise.

Benchmark Model

I will create three Convolutional Neural Networks and compare the results. The first CNN will have my own architecture that I will construct using Pytorch. The second CNN will have an architecture of the ResNeXt-101 32x8d. The third CNN will be the AWS Built-in Image Classification algorithm that uses the ResNet-152 architecture. The benchmark model that I will use will be AWS Built-in Image Classification Model.

Evaluation Metrics

The metrics that will be used are by measuring these:

1. True Positives(TP) - The number of correctly identified images with pneumonia.
2. False Positives(FP) - The number of images without pneumonia falsely identified with having pneumonia.
3. True Negatives(TN) - The number of correctly identified images that do not have pneumonia.
4. False Negatives(FN) - The number of images with pneumonia falsely identified with not having pneumonia.

We want to decrease FP and FN as much as possible. Especially FN, because if a patient that has pneumonia is classified as if not having pneumonia, that is a big problem!

Using those measures we can compute these ratios:

1. Sensitivity (or Recall) - The percentage of correctly classified images with pneumonia of all images of pneumonia.

$$\frac{TP}{TP + FN}$$

2. Specificity - The percentage of correctly classified images without pneumonia of all images that do not contain pneumonia.

$$\frac{TN}{TN + FP}$$

3. False Positive Rate -

$$1 - Specificity = \frac{FP}{TN+FP}$$

Just as above, we would like to decrease FP and FN as much as possible. We should want to maximize Sensitivity as much as possible. This is because we want to maximize the number of images that have pneumonia to be correctly classified.

Using Sensitivity and FPR, we can use the AUC metric. The Reciever Operating Characteristics (ROC) measures the performance of the model on different thresholds and the AUC (Area Under the Curve) accesses that performance. You can watch this video here to learn more. Or you can read this here.

For a quick guideline of these metrics please refer to this Lecture Slideshow by James D. Wilson Ph.D. from the University of San Francisco here.

Project Design

These will be the steps that I will use to create this Image Classifier:

1. Clean the Data and prepare it for Pytorch and AWS
    a. Convert all images to the RGB format
    b. Resize all images to be 224x224
    c. Save transformed images
    d. Upload cleaned dataset to S3
2. Build the models
    a. Build ResNet-152 using AWS Built-in Image Classifier
    b. Build ResNeXt-101 32x8d Model with Pytorch Built-in model
    c. Build standard CNN with PyTorch with three convolutional layers
3. Both train and test the models
    a. Train each model
    b. Record their AUC score
4. Determine which model performs the best using the above evaluation metrics
    a. Compare the AUC scores of each model
    b. Choose the model that has the closest AUC to 1.00
5. Build the Web-App using the Lambda and API services of AWS
6. Deploy Model and Web-App