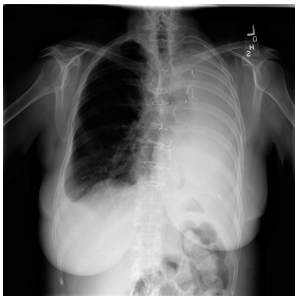
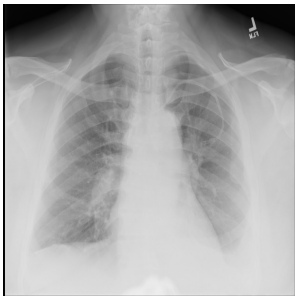
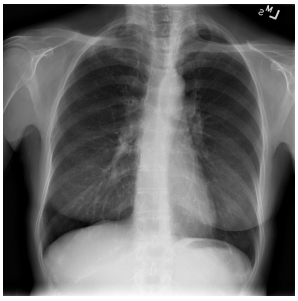
Anthony Le, Daniel Valner
Sunet ID: antle1, dvalner

Final Report: RSNA Pneumonia Detection and Localization

Overall Task:

In 2018 the Radiological Society of North America had a competition for creating an algorithm that not only detected the pneumonia through computer vision, but also localized the area in which the pneumonia affected. Pneumonia is one the leading causes of mortality in children under the age of 5 with a 16% mortality rate. One of our goals in researching computer vision in the medical field is to further our knowledge in a technology that could potentially be world-changing amidst a global pandemic. Early detection algorithms could point out things that eluded doctors' first judgements, ultimately saving more lives. In a time where COVID-19 is the world's main concern, any sort of improvement in healthcare can have tremendous impact.

CNN's have been used for medical image classification recently as seen with studies using neural networks to help with mammography in breast cancer patients[5]. These researchers found that with a three-layer, feed forward neural network, they could distinguish between benign and malignant lesions better than the average performance of an attending resident radiologist alone. This, along with many other studies, have proven that neural nets and computer vision will play a major role in healthcare in coming years.

As the Kaggle competition has concluded and is open source we analyzed the winner's solution in order to use transfer learning or inspire our model based on their architecture. They first built a classification-detection pipeline using a 10 fold cross-validation ensemble. Then for detection they used Keras's RetinaNet, a deformable CovNet and Relational Network for object detection. They noted, however that pretraining the backbone of RetinaNet on the pneumonia dataset showed no improvement. They reached a final IoU (intersection over union) of .256. We will be using accuracy percentage as opposed to IoU but could use this to compare our model with theirs.

Based on preliminary research, we understood that convolutional neural networks were all the jive nowadays[2], so we decided upon using architectures similarly shown in class when

working with high resolution images. Knowing that some of these architectures require a lot of computational power, we ran our model using an EC2 AWS instance. Using AWS's GPU's seriously improved our runtime and cut down the cost of training one epoch by staggering amounts. We also looked through several papers concerning the use of deep learning to identify or classify medical diagnoses and found several implementations that guided the architecture we put together. After seeing that most of the incredibly successful models were all using transfer learning, we aimed to import pre-trained models and weights such as VGG16 and ResNet-InceptionV3. Most of these models were able to correctly label diagnoses up to 87%[3]. So this is the aim of our final project.

Our goal is to build a classifier that exceeds the level of perception that a doctor would have. We judge this to be about 95% accuracy. Specific goals for our first milestone included achieving a baseline model that could perform better than that of a layman's level of perception, judged to be about 60%. For the training set we wanted to achieve a 75% accuracy rate, along with a 70% validation set accuracy. We did not achieve this accuracy with our first milestone, but rather got a 78.6% training accuracy and a 64.3% validation accuracy. This milestone we aimed at achieving our original goal of 75%. We managed to accomplish this task which will be described in more detail later.

Dataset:

Our dataset[1] was sourced from a kaggle competition for predicting bounding boxes around areas of lung x-rays that were signs of RSNA Pneumonia, and then classifying patients as either positive for pneumonia or negative for pneumonia. Our training set includes 26,500 data points stored as DCM, or DICOM files. These files contain a medical image, in this case x-rays of the lungs, as well as several other attributes concerning the patient such as ID, name, age, and even information about the image pixel data. The image associated with each datum is an x-ray of the patient's lungs. The labels accompanying the images are bounding box dimensions in the format (x_min, y_min, width, height), as well as a binary label indicating whether or not the patient had Pneumonia. The label also includes a more detailed version including the type of positive or negative class for each image/patient: no lung opacity/not normal, lung opacity, normal. The training and test sets have a roughly even ratio of positive and negative examples. The test set includes 500 data points. With a decently large amount of training data, we can consider having a larger training set compared to the validation and test sets. For milestone 1, we tested different Keras architectures with a smaller portion of the data (500 training examples, 100 validation examples) as a preliminary step towards creating a great computer vision application. For this milestone we greatly improved our data usage with 10,000 training examples and 2,000 validation examples.

Baseline:

Our baseline model was a default Keras convolutional neural network consisting of 4 convolutional layers using the ReLu activation function and a final binary output layer using the sigmoid activation function. We used dropout in the last layer of the model with a 0.5 probability of dropping out a certain node. In order to use this model we had to convert our dcm files into jpg and transfer the labels (excluding the bounding box dimensions) into its own csv file. This model was trained on 500 images and validated on 100 images. The data had a 1:1 ratio of positive to negative examples. There was no transfer learning at this stage of the project, but in the future we may consider using transfer learning to use a pre-trained classifier in order to jumpstart our model. This will only be the case if 26,500 images/data points are not sufficient to achieve our intended accuracy ratings.

For milestone 1, our baseline model had the following:

Train Error: 78.6% accuracy **Validation Error:** 64.3% accuracy

With such a large gap between the validation and training error, we can see that overfitting may be a problem. Only using 500 data points, we can see why the accuracy is so low on our baseline model. If we increase the number of training examples, this should produce more accurate results.

Development:

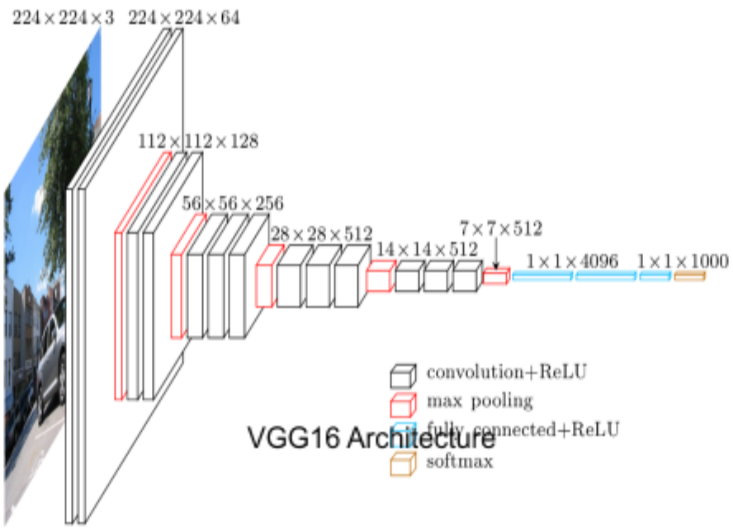
Our next approach was to move the codebase into AWS and train on their GPU's. This would allow us to use more of the data since every epoch of training would be reduced by a large factor. We started to train on 10,000 training examples and 2,00 validation examples. With the same Keras model we received:

Train Error: 74.11% accuracy **Validation Error:** 71.88% accuracy

We found it odd that the training accuracy went down when training on a larger number of examples. Our next attempt was to increase the number of epochs from 10 to 50 because we noticed that every epoch was tending towards lower training error and loss and figured this pattern could only continue with more training. We also changed the optimizer from rmsprop to Adam to see whether or not it improved accuracy. From these changes we received:

Train Error: 83.63% accuracy **Validation Error:** 77.08% accuracy

Final Approach:



Our final approach was to train on all of our 26,000 images and validate on the final 500 test images. With this increase in data, we had our model output an accuracy close to the ~85% accuracy of models found in contemporary literature on medical diagnosis classifiers[3].

Our next approach also included transfer learning with pre-trained CNN layers that we can stack our classifier on top of[4]. We used pre-trained models such as ResNet Inceptionv3 and VGG16 trained on ImageNet.

Train Error: 87.64% accuracy
Train Error: 85.64% accuracy

Validation Error: 82.3% accuracy (VGG16)
Validation Error: 83.4% accuracy (Inceptionv3)

Analysis of Results:

Our final approach yielded the above results with transfer learning. We knew that the addition of pre-trained layers into our model would increase test performance with generalizable functions such as object classification. The Inceptionv3 turned out to perform the best, as was seen in the winner of the Kaggle competitions method/procedure. Throughout our project, we could see how important the amount of training data was. In the beginning when we weren't training on AWS, we found that we were overfitting to the training set and did not perform as well as we wanted at test time. But as we increased the amount of data we had, we saw the test accuracy increase to the levels we wanted. We also introduced dropout with a 0.5 chance for each node in the later layers of our network to decrease this overfitting. From our results, we understand the importance of using pre-trained architectures to get to industry standard performance in image classification.

Contributions:

Over a plethora of Zoom calls, we did quite a bit of preliminary research to determine what topic we wanted to cover. While our interests ranged from Neural Style Transfer models to

analyzing COVID data. We both ended up agreeing to do something in the medical realm. Daniel found the Kaggle competition and we both were excited about it and created a joint document to compile our research.

Through this joint research we planned the architecture of the model together. Anthony wrote the first draft of the model and ran it on his computer and Daniel revised it. We then decided to move the project onto AWS where Anthony made the account and uploaded the revised draft. Finally, we both worked on the final draft together and jointly wrote the reports. Although being physically separated was a slight barrier we were able to collaborate fairly easily and we each got to contribute our fair share of effort into the project.

References:

[1] "RSNA Pneumonia Detection Challenge." *Kaggle*, www.kaggle.com/c/rsna-pneumonia-detection-challenge/data.

[2] Brownlee, Jason. "Object Classification with CNNs Using the Keras Deep Learning Library." *Machine Learning Mastery*, 12 Sept. 2019, machinelearningmastery.com/object-recognition-convolutional-neural-networks-keras-deep-learning-library/.

[3] Singh, Pradeep, et al. "Breast Cancer Classification with Keras and Deep Learning." *PyImageSearch*, 18 Apr. 2020, www.pyimagesearch.com/2019/02/18/breast-cancer-classification-with-keras-and-deep-learning/.

[4] Chollet, Francois. "The Keras Blog." *The Keras Blog ATOM*, blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html.

[5] Wu, Y. "Artificial Neural Networks in Mammography: Application to Decision Making in the Diagnosis of Breast Cancer." *Radiology*, 1 Apr. 1993, pubs.rsna.org/doi/abs/10.1148/radiology.187.1.8451441.