COVID-19 X-ray Classification

Video Presentation: https://youtu.be/z-Qm2BJKWyA

Github: https://github.gatech.edu/arao338/Team8Covid19

Anita K. Rao, BS[1], Ilker Yaramis, BS[2], Sukru C. Sezer, BS[3]
[123]Georgia Institute of Technology, Atlanta, GA, USA;

Abstract

Chest X-ray image data and accompanying clinical attributes were obtained from several open source databases, and processed using machine learning and deep learning techniques, in order to diagnose patients as positive for COVID-19 or pneumonia, or no findings. The basis for model decisions are visualized on source X-ray images using COVID-19 Detection App to further assist healthcare workers.

1. Introduction

The global pandemic due to the Coronavirus Disease 2019 (COVID-19) has overwhelmed healthcare systems, with over 1 million deaths as a result of the virus thus far. The focus of this work is on early detection of the virus, using chest X-ray images and mark up source X-ray images to assist healthcare to identify COVID or pneumonia indicators in patients' lungs. Given that chest X-rays are used as a first line of defense for suspected COVID-19 and pneumonia patients[2], we have used various open-source databases of chest X-ray images to develop a deep learning classification model that can identify patients as positive for COVID-19, positive for pneumonia, or no findings. The goal is for our model to assist frontline healthcare workers who could greatly benefit from earlier detection of patients with COVID-19.

2. Literature Survey

Although COVID-19 and its impact on global society is recent in many ways, there have been trailblazing new roads into what is behind this mysterious disease and how to most effectively diagnose it. With relatively little known about this disease, there is a need for auxiliary diagnostic tools; one of these tools is machine learning applied with radiological imaging to construct a model that automatically detects COVID-19 in patients through a simple x-ray of the lungs.

Three new techniques are aiming to tackle this issue:

The DarkCovidNet model in "Automated detection of COVID-19 cases using deep neural networks with X-ray images"[6] was developed to provide accurate diagnostics for binary classification (COVID vs. No-Findings) and multi-class classification (COVID vs. No-Findings vs. Pneumonia). The DarkNet model leveraged 17 convolutional layers each with unique filtering. Each convolution layer of this model extracts features from the chest X-ray input, applies a pooling layer to reduce the size for computational performance, and leverages a neural network to recognize underlying relationships in this data. This model produced a classification accuracy of 98.08% for binary classes and 87.02% for multi-class cases.

The COVIDNet-S model in "COVIDNet-S: Towards computer-aided severity assessment via training and validation of deep neural networks for geographic extent and opacity extent scoring of chest X-rays for SARS-CoV-2 lung disease severity"[8] used a larger dataset containing 13,975 chest x-ray images and 13,870 patient cases consisting of healthy patients and patients with different forms of pneumonia. This model uses transfer learning to initialize the deep neural network parameters in this study and data augmentation to improve the neural network's performance. The model is based on training samples on which randomly generated translations, flips, zooms, and shifts (among other actions) were applied to increase data diversity for better neural network results. This model determines the geographic and opacity extent in a patient's lungs and can not only determine the presence of COVID-19 but also assess the disease severity of COVID-19 at a 92-94% accuracy.

Google's COVID-19 forecasting model in "Interpretable Sequence Learning for COVID-19 Forecasting"[1] uses two new methods that more accurately predict the exposure and progression at the individual, community and country level, known as compartmental modeling and encoding covariates. These compartments have also been made granular for more specificity and each compartment has its own equation which has been maximized for effectiveness. When put together, the compartmental modeling creates a truer picture of COVID-19 spread than the previous two models, since it accounts for policy changes such as travel bans or public restrictions on disease

progression. Most studies have been limited to the impact of one or two covariates, but this one models numerous static and time-varying covariates simultaneously, allowing for a truer reflection of time and policy impact.

In all, there is great interest in the AI/ML community to be able to aid epidemiologists and physicians in the fight against COVID-19. In the efforts mentioned above, the focus is on model training without emphasis on educating doctors and hospitals on usage of the model. For this project, in addition to classifying X-ray images accurately, we have leveraged data visualization techniques to convey this information to doctors in a palatable way.

We have provided a basis of our predicted diagnoses by overlaying heatmaps on source images. This can indicate potential COVID-19 findings such as bilateral multifocal consolidations, which may progress to involve the entire lungs or Small pleural effusions[12].

3. Data

For the analysis, we utilized the COVID-19 Dataset[4]. In this dataset, X-ray images were collected from two different sources: COVID-19 X-ray image database developed by Cohen JP for COVID-19 cases, and ChestX-ray8 database for normal and pneumonia cases. In addition to 930 COVID-19 positive case images there are 1845 images for pneumonia and 1054 images for normal cases.

From this dataset, we created 5 train/test pairs using the Monte Carlo Cross Validation method to mitigate potential overfitting of our models. The data was split into 80% training data and 20% test data.

4. Approach/Metrics

The focus of our analysis was on binomial classification (COVID-19 vs No Findings) and multinomial classification of the COVID-19 chest X-ray image dataset into the following categorical values: COVID-19, Pneumonia, and No Findings. Our approach considered both machine learning and deep learning classification algorithms.

4a. Pre-processing

The input data had images of varying brightness, shift and scale. We performed image augmentation to ensure scale, brightness and shift invariance in the extracted features and in predictions. Image augmentation included flipping images horizontally and vertically, augmenting brightness level, and applying random scaling and shifting. This step ensured increased accuracy and improved spatial symptom detection in prospective inferences. Images were also re-scaled to a standard size of 128 by 128 pixels.

4b. Feature Extraction

Feature extraction is a highly useful phase of the model development process for machine learning pipelines. Starting from a very large set of features, we narrowed down our feature set such that it was optimized for spatial separation between features and was ideally more effective for image classification. Since the COVID-19 dataset contains chest X-ray images, we needed to use feature extraction techniques that could process the input image data and isolate useful features for ”COVID-19”, ”Pneumonia”, and ”No findings” predictions.

The two techniques that were used for extraction were (1) a multilayered neural network and (2) Principal Component Analysis (PCA) dimensionality reduction.

Using deep learning for the feature extraction process is very useful for image classification, particularly by using hidden layers of a neural network to perform the mapping of a set of pixels to a target group.

Hand designed and carefully picked filters have always been used in image processing. One example for those filters are Haar-like features[11], which is useful in face detection on images[10]. These filters excel at catching facial features such as hair (dark) - forehead (bright) - eyes (dark) or horizontal filters to detect eye - nose - eye. Analogous to face detection, there can exist a set of filters to extract useful features from X-ray images to classify patients as healthy, COVID-19, or pneumonia. However, designing filters to detect a lung disease is a relatively non-trivial task compared to face detection. Additionally, the disease in question can be a novel disease and design of appropriate filters or feature extractors may not be convenient within a short time. At this point, the success of neural networks in non-linear function approximation can be integrated into medical image classification. A neural network can be used to learn those filters required to extract meaningful and discriminative features, which then allow to detect a certain disease.

Using convolutional networks over simply flattening the image and using multilayer perceptrons also decreases the number of features used therefore increasing Bayesian Information Criteria and risk of overfitting to the training data. This spatial approach also allows visualizing model decisions or extracted meaningful features at a deeper level.

A relatively shallow convolutional neural network was trained to learn useful filters that can extract features which is later fed to several different types of classifiers. A 3x3 convolutional block (CB) is defined as follows:

$$\text{Conv2D -> (3,3) -> Batch Normalization (BN) -> Leaky ReLU (LR) -> MaxPool2D (2, 2)}$$

And each convolutional layer consisted of multiple filters. The resulting Convolutional Neural Network (CNN) architecture is given below:

$$\text{CB -> CB -> CB -> CB -> FC(200) -> FC(8) -> Softmax(3)}$$

The output of the last convolution layer represented extracted features from the image, which were then flattened and dumped into a table. This table of over 8000 extracted features was then fed into a PCA step, which was performed to reduce the dimensionality of the feature space using Python's SciKitLearn library. 40 principal components were deduced from the existing feature list.

PCA is a common and effective technique used to reduce the dimensionality of a feature space, and can be particularly useful for input images, as "humans can not visualize data on more than 3 dimensions" and thus "it is usually helpful to reduce multidimensional datasets into 2 or 3 dimensions and graph them in order to get a better understanding of the data."[5]. Performing PCA showed a benefit of a 10x reduction in CPU processing time for models that were trained using big data tools.

4c. Training/Inference

Three models were considered for multinomial and binomial classification:

1. Logistic Regression(linear kernel space based)
2. Random Forest (tree based)
3. Convolutional Neural Network (deep learning model)

The first two models were implemented using Spark's MLlib package. For more background on the model parameters, the Logistic Regression model had a maximum of 10 iterations and used an L2 penalty, or Ridge regression, with regularization parameter λ set to 0.3. The optimal L2 penalty was determined by toggling this parameter and identifying where overfitting became apparent. The Random Forest Classifier was tuned to 10 trees with max depth of each tree equal to 6. This model underwent a similar hyperparameter tuning proess by toggling the max depth size to identify the optimal value before overfitting occurred.

The third model considered was a convolutional neural network, which was implemented using Python's SciKitlearn package. The model architecture is shown in the figure below.

In this model, features extracted by convolutional neural network are directly fed into fully connected layers which are on the same network as the convolutional parts. Fully connected multilayer perceptron part of the model then performs a non-linear function approximation using weights and activations, and outputs logits for each of three classes. These logits are transformed into a probability distribution using softmax function.

For the CNN, following parameters were used for the training: Adam algorithm for gradient descent with 0.999 as RMS coefficient β_2 and 0.9 as momentum coefficient β_1 with a learning rate of 0.001. Model was trained on categorical cross entropy loss for multi-class classification task, and binary cross-entropy loss for binary classification task. No regularization is performed during training.

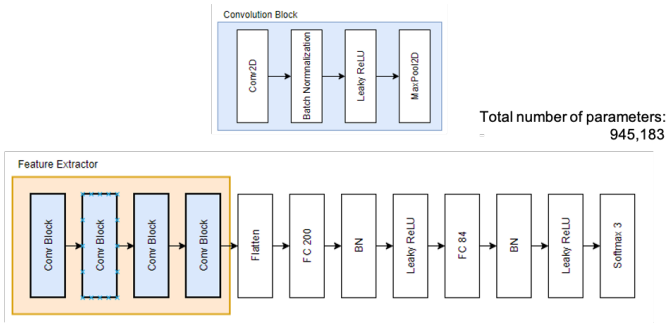

Figure 1: CNN Model Architecture

4d. Model Evaluation

The performance metrics that we used to measure effectiveness of the training model were accuracy, precision, recall, and f1 scores. In the first flow explained in the training/inference section above, these metrics were calculated using Spark MLlib's MulticlassClassificationEvaluator[7].

For the evaluation of the deep learning model containing Multi-Layer Perceptron (MLP) predictions, these metrics were calculated using weighted average for precision, recall and f1 scores.

For the binary classification task; precision, recall and f1 scores were calculated for COVID-19+ class only.

5. Experimental Results

Figure 2 compares the accuracy, precision, recall, and f1 score metrics for multi-class classification task across the Logistic Regression, Random Forest, and CNN models.

Figure 2: Model Comparison by Performance Metrics for Multinomial Classification

Figure 3 compares the accuracy, precision, recall, and f1 score metrics for binomial classification task across the Logistic Regression, Random Forest, and CNN models.

Figure 3: Model Comparison by Performance Metrics for Binomial Classification

We can see that across all metrics, the CNN model outperforms the Logistic Regression and Random Forest models for both in-sample and out-sample data. The similar accuracy scores for CNN on both training and test data indicate that earlier techniques in the pre-processing and feature extraction phases to avoid overfitting were ultimately advantageous.

Given that CNN was our best performing model, we compared its performance on the binomial versus multinomial classification problem. Figure 4 compares the accuracy, precision, recall, and f1 score metrics for binomial vs multinomial classification for the CNN model.

Figure 4: Model Comparison by Performance Metrics

We can see that across all metrics, the CNN model performs slightly better on the binomial problem of classifying chest X-ray images into COVID-19 Positive vs No Findings, over the multinomial problem of classifying images into COVID-19 Positive vs Pneumonia vs No Findings. This is a reasonable result, as binomial classification is a simpler problem overall than multi-class classification. On the whole, the model performs well for both types of problems as seen by the $>90\%$ accuracy for both binomial and multinomial results.

6. Model Interpretation

Model interpretation is crucial for understanding how well our models predict on unseen data. For classical machine learning models we can use various feature importance techniques to identify which features have predictive power and use these features to interpret our models. However for deep learning models we can not use

this technique. In this project we built a CNN-type neural network model and to interpret this model we are using Gradient-weighted Class Activation Mapping (Grad-CAM)[9] methodology. Grad-CAM proposes a technique for producing 'visual explanations' for decisions from a large class of CNN-based models, making them more transparent and digestible[9]. One characteristic of a CNN is that the earlier layers capture low-level concepts whereas the deeper layers capture high-level features. Therefore, GradCAM focuses on the last convolutional layer, as they provide a better picture of what the network is paying attention to when classifying a particular class.

$$a_k = \frac{1}{Z} \sum_i \sum_j \frac{\partial_y}{\partial A_{ij}^k} \tag{1}$$

$$L = ReLU(\sum_k (a_k A^k)) \tag{2}$$

Equations 1 and 2 show the process of the Grad-CAM algorithm. Here A^1, A^2, ..., A^k denotes the unpacked version of k spatial maps at the last convolutional layer in CNN model, while a_1, a_2, ..., a_k represents weights for the corresponding spatial map. We are computing a localization map by multiplying weights with respective spatial maps and summing them up.

In order to calculate weights, we are first computing influence of each feature within each spatial map on y, by calculating the partial derivative of y with respect to each feature in each spatial map (A_{ij}^1, A_{ij}^2 ...). Computation of feature influence gives us a matrix, telling us the influence of each feature on the output. After this, we are calculating the average influence of the spatial map on the output by taking the average of the feature influences. This average influence is our weight to compute a localization map. Next we are applying ReLU function to the localization map in order to set the negative numbers to 0 and keep positive numbers as they are.

Grad-CAM methodology gives us the power to understand how our CNN model is making decisions and it helps us to easily interpret our models.

7. Discussion

The goal for our analysis was to explore various models for training on Chest X-ray images. This required image pre-processing, feature extraction/selection and cross validation techniques, usage of scripting and big data tools such as Python and Spark, implementation of machine learning and deep learning models, and analysis of model performance and model interpretation.

The findings in the Experimental Results section show that the Convolutional Neural Network model performed better than Logistic Regression and Random Forest. We see that across all models, the training/test scores across each metric are quite similar.

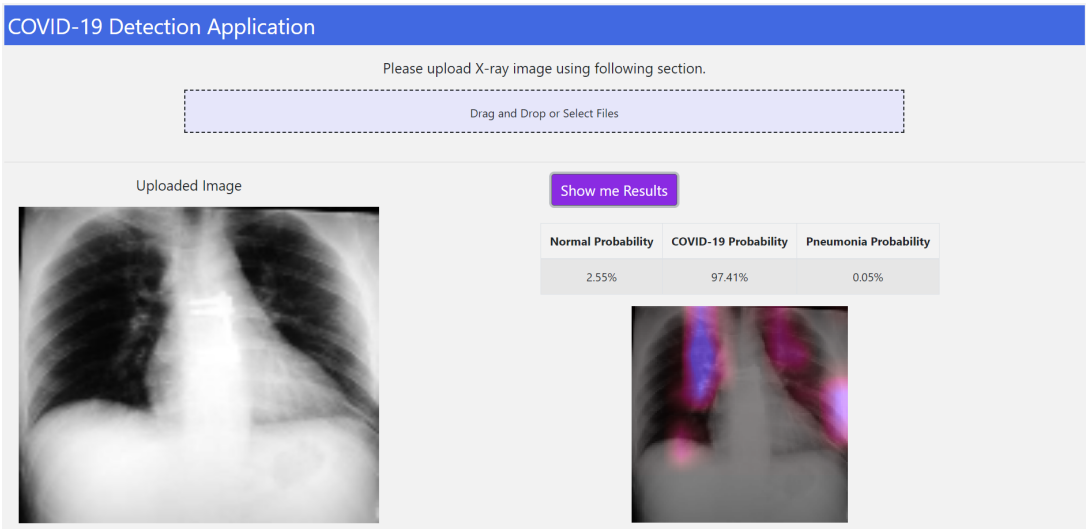

Figure 5: COVID-19 Detection Application

In tandem to the analysis, we have also developed a visualization tool called COVID-19 Detection Application to assist physicians interested in a more streamlined solution for chest X-ray classification. Our aim with this tool is not only to provide the predictions of the model but also to help physicians understand how the model is making determinations by adding markups on the source image.

Users can upload images to the tool by simply dragging and dropping the image or selecting the image from their device. Upon uploading image, the user will see this image on the left-hand side of the application. When users click on the "Show me Results" button, the application runs the CNN model in the backend and displays the probabilities of each class in tabular format. Figure 6 shows the source X-ray image and Grad-CAM applied image.

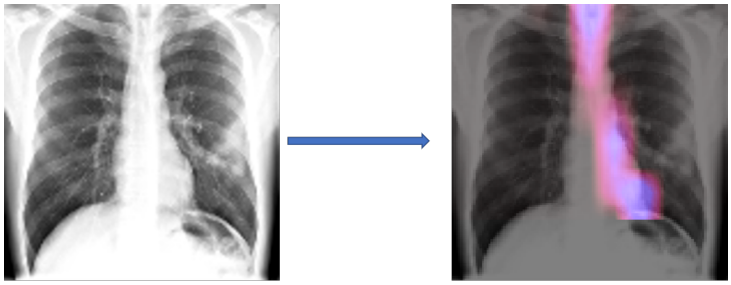

Figure 6: Original and GradCAM applied X-ray Image

By using Grad-CAM methodology we are able to visualize which aspects of the X-ray image the model is interested in when making decisions. In the image above, the model makes decisions based on brighter areas. Physicians can look closely to the brighter areas and make their evaluations quickly about patients.

8. Conclusion/Optimization

There were three key takeaways from our COVID-19 X-ray classification study.

1. Our models were initially overfitting when operating solely on the original dataset. By using a larger neural network with augmented data to generate new features, we saw the models improve in out-sample accuracy.
2. CNN outperformed Logistic Regression and Random Forest on all evaluation metrics for in-sample and out-sample data.
3. The COVID-19 Detection App that we developed sets our work apart by helping physicians to identify which aspects of X-ray images are strong indicators for COVID-19 and other conditions such as Pneumonia. We hope that our results can be utilized in the real world using this application.

One future optimization includes using our model to go further than simply highlighting a critical region by identify bounding box regions, such as sections of the right or left lung, that are most useful for diagnosis.

Moreover, we are going to deploy enhanced COVID-19 Detection application into Azure Web Services, in this way users can upload their x-ray images and see model predictions as well as how the model is making decisions.

9. Team Contributions

The tasks of this project were evenly split amongst the three team members. Sukru was responsible for the deep learning tasks, including feature extraction using a multi-layered neural network, and the implementation of a Convolutional Neural Net for model training. Anita oversaw the machine learning tasks, including the implementation and tuning of the Logistic Regression and Random Forest models. Ilker implemented the COVID-19 Detection Application using Grad-CAM methodology to bring the model training process to life by visualizing the key segments of the X-ray images used for COVID-19 classification. We would also like to acknowledge Shivanjali Singh, a former team member, who contributed to the Literature Survey research in the early project phases. The current team members contributed equally to the remaining scoping, research, and authoring of the paper/presentation.

References

1. Arik, S.O.; Li, C.; Yoon, J.; Sinha, R.; Epshteyn, A.; Le, L.T.; Menon, V.; Singh, S.; Zhang, L.; Yoder, N.; Nikoltchev, M.; Sonthalia, Y.; Nakhost, H.; Kanal, E.; Pfister, T. (2020). Interpretable Sequence Learning for COVID-19 Forecasting. https://arxiv.org/abs/2008.00646
2. CSE 6250 Projects: Big Data Analytics for Healthcare (2020). CSE6250_project_2020Fall.pdf
3. Machine Learning - Logistic Regression, https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_classification_algorithms_logistic_regression.htm
4. Muhammedtalo. (n.d.). Muhammedtalo/COVID-19. https://github.com/muhammedtalo/COVID-19
5. Norena, S. (2018, June 15). PCA (Principal Components Analysis) applied to images of faces. https://medium.com/@sebastiannorena/pca-principal-components-analysis-applied-to-images-of-faces-d2fc2c083371
6. Ozturk, T., Talo, M., Yildirim, E. A., Baloglu, U. B., Yildirim, O.; Acharya, U. R. (2020). Automated detection of COVID-19 cases using deep neural networks with X-ray images. Computers in Biology and Medicine, 121, 103792. doi:10.1016/j.compbiomed.2020.103792
7. What is the best validation metric for multi-class classification? (2020, October 03). https://sebastianraschka.com/faq/docs/multiclass-metric.html
8. Wong, A.; Lin, Z.Q.; Wang, L.; Chung, A.G.; Shen, B.; Abbasi, A.; Hoshmand-Kochi, M.; Duong, T.Q. (2020). COVIDNet-S: Towards computer-aided severity assessment via training and validation of deep neural networks for geographic extent and opacity extent scoring of chest X-rays for SARS-CoV-2 lung disease severity. https://arxiv.org/abs/2005.12855
9. Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.;, T.Q. (2019). Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. https://arxiv.org/abs/1610.02391
10. Viola, P.; Jones, M.(2001). Rapid object detection using a boosted cascade of simple features. https://www.merl.com/publications/docs/TR2004-043.pdf
11. Haar, A. (1910). On the Theory of Orthogonal Function Systems. http://www.laurent-duval.eu/Documents-WITS-starlet/Haarlets/Haar_A_1910_ma_zur_tofs-haarlet.pdf
12. Cleverly J.; Piper J.; Jones, M. M.(2020). The role of chest radiography in confirming covid-19 pneumonia. https://doi.org/10.1136/bmj.m2426
13. BIMCV Medical Imaging Databank of the Valencia Region, Pertusa, A.; Vaya, M. (2020, May 20). BIMCV-COVID19+. https://osf.io/nh7g8/
14. Ieee8023. (n.d.). Ieee8023/covid-chestxray-dataset. https://github.com/ieee8023/covid-chestxray-dataset https://github.com/ieee8023/covid-chestxray-dataset