COVID-19 Detection Of X-Ray Image

https://www.youtube.com/watch?v=l7xWC0mCf9M

Niliang Lu
University of Illinois - Urbana Champaign
Master of Computer Science
niliang2@illinois.edu

Jiayu Peng
University of Illinois - Urbana Champaign
Master of Computer Science
jiayup4@illinois.edu

Jinggong Zheng
University of Illinois - Urbana Champaign
Master of Computer Science
jz34@illinois.edu

Pui Sze Ng
University of Illinois - Urbana Champaign
Master of Computer Science
ppn2@illinois.edu

ABSTRACT

Objective: In this project, we have combined Professor Sun's FLANNEL paper and other X-Ray abnormal detection papers. We would like to detect if a patient has been affected COVID-19 based on their X-Ray images. This study contains two types of X-Ray images in the dataset: COVID-19and non COVID-19 images from normal and pneumonia cases. Our goal is to differentiate COVID-19 against non COVID-19 X-Ray images.

Materials and Methods: We collect the X-Ray images dataset from Curated COVID-19 dataset and COVID Chest X-Ray dataset. There are $1,934$ chest X-Ray images consisting three types of images: normal, pneumonia, and COVID-19. We put the X-Ray image dataset into 4 baseline models - AlexNet, VGG-16, Inception V3, and ResNet18. Then, an Ensemble model is composed by these four baseline models.

Results: Ensemble model has higher accuracy, better AUC, and F1 among 4 baseline models. Comparing to the best baseline model VGG-16, we find that the standard ensemble model achieved 4.7% F1 score improvement. In addition, the ensemble model V1 with focal loss functions and image augmentation achieves 5.8% F1 score improvement on COVID-19 classification.

Keywords
Deep learning, Healthcare, Machine learning, COVID-19 detection, Neural network ensemble.

1. INTRODUCTION

The coronavirus disease 2019 (COVID-19) [1] is a contagious disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) that has tremendously spread to the whole world within a few months. The COVID-19 not only damages the global economy but also causes a global public health crisis. Lots of people lost their jobs, relatives and even their lives. Up to Mar 2021, which the pandemic [2] lasted almost 1 year globally, the COVID-19 has killed 2.77 million people and infected more than 126 million people globally [3]. Seriously, the number of infected people has explosively increased within a few months. Every country in the world gets into shortage of medical supplies and skyrocketing hospitalization rate. Therefore, early detection and diagnosis of COVID-19 can reduce tremendous amount of workload for doctors.

When a doctor diagnoses an undetermined COVID-19 infected patient, chest radiography (X-Ray) is taken as the first-line imaging modality. There are several reasons why X-Ray is more popular than computed tomography(CT). First of all, X-Ray diagnosis is more sensitive than CT. Secondly, as a tool for detection, quantification, analysis and follow-up COVID-19 cases, many researchers believe that X-Ray based system is more effective. Last but not least, X-Ray is highly accessible in every hospital.

Therefore, our research project would like to apply what we have learned in our lectures - Deep Learning for Healthcare and focus on the image classification model on the X-Ray image dataset. We will use convolution neural networks (CNN) especially in binary classification (COVID-19 vs No-Findings and pneumonia). The dataset contains 1,934 X-Ray images with 500 images of no-finding, 500 images of pneumonia, and 566 images of COVID-19. We put the dataset into 4 different baseline models of CNN: AlexNet, VGG-16, Inception V3, and ResNet18.

Our goal is to create a Ensemble model using 4 baseline models that can improve accuracy on differentiate COVID-19 X-Ray images against non COVID-19 X-Ray images.

2. RELATED WORK

In 2019, Lu, Ivanov, and Mayrhofer published an article on

JAMA Network Open that proved the convolutional neural network (CNN) was able to identify the pre-stage of chest diseases based on the radiographs. The CNN model can detect the high risk of long-term mortality from chest X-Ray images [4].

According to Ghaderzadeh and Asadi's article in the Journal of Healthcare Engineering, they concluded deep learning was an accurate and effective technique in the diagnosis of COVID-19. Through their systematic review of 168 articles, they found that the ResNet model had better performance than other models. Especially using ResNet-50 architecture had the best efficacy in diagnosing the COVID-19 by analyzing X-Rays images among the research models that used deep learning approaches to diagnose COVID-19 by analyzing X-Rays, ranging from November 1, 2019, to July 20, 2020. They provided strong evidence that applying deep learning of the CNN model can reduce false-positive and false-negative errors in COVID-19 diagnosis [5].

From Jadon's research paper, he thought the less amount of COVID-19 X-Rays dataset and biased data scenarios caused bad inference on the predicting results. He trained the model in a low-data regime, few-shot metrics, applied with CNN, transfer learning, and unsupervised learning(t-SNE and PCA) as the model architecture which improved the model performance when the data is scarce [6].

Since the CNN is an effective deep learning model that not only applied for classifying general pneumonia symptoms on radiographs but also had exceptional accuracy on COVID-19 diagnosis. Our research is based on CNN and multi-class classification to achieve better performance than other baseline models.

3. DATASET

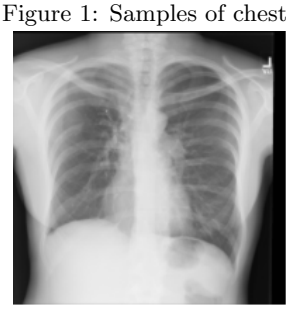
In this project, we have combined two datasets - Curated COVID-19 dataset[7] and COVID Chest dataset [8].

The first dataset is the Curated COVID-19 dataset[7] and it is a publicly available dataset compiled by Ozturk et al. It contains 1125 images with three types - 500 images of normal, 500 images of pneumonia, and 125 images of COVID-19 (Figure 1).

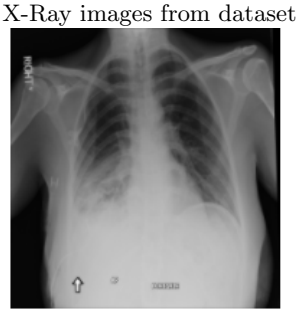
The second dataset is COVID Chest dataset[8] and it is a publicly available dataset compiled by Cohen et al. It contains 930 images with three types - 22 images of normal, 344 images of pneumonia, and 563 images of COVID-19. The new included COVID-19 images also have some X-Ray scanned on the side or other position instead of having the regular X-Ray image scanned in front. The different positions of images will improve the difficulty of training.

The Curated COVID-19 dataset and the COVID Chest X-Ray dataset have been combined together in order to have a more compressive super dataset. However, by combining COVID-19 images of both datasets, some images have the same image name. We are not 100% sure if the images with the same name are really the same. But we think it is highly likely the images with the same name from the different datasets are the same. Therefore, the images with the same file name from both datasets are only kept one instead of two.

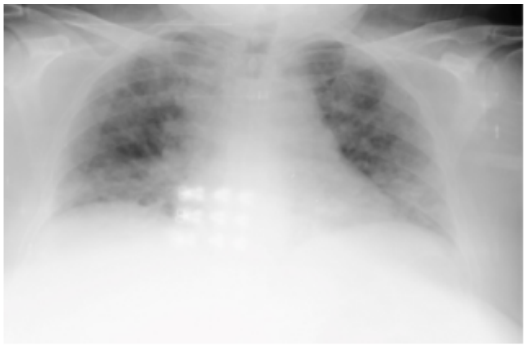
Figure 1: Samples of chest X-Ray images from dataset



(a) No findings



(b) Pneumonia



(c) COVID-19

Table 1: Final dataset summary

Type	Number of Images
COVID-19	688
Non COVID-19	1367
Total	1934

4. APPROACH/METHOD

Four baseline models which include AlexNet, VGG-16, Inception, and ResNet-18 will be used to evaluate the prediction accuracy of COVID-19. In order to make improvements on top of the baseline models, a deep analysis and team discussion will be performed based on the prediction accuracy results of four baseline models.

4.1 Image Preprocessing

4.1.1 Image scale

The original images have a different aspect ratio, we need to pre-process the image to make it compatible with baseline models. We first scale the original images to make a shorter edge equal to 256, then we crop a 224 $\times$ 224 image at the center of the scaled image. For Inception V3, because the model takes 299 $\times$ 299 images as input, we scale the original image to 350 first, then center crop to 299.

The cropped image is slightly smaller than the scaled im-

ages because there are usually some text or special markers around the X-Ray images, we remove them to prevent models from overfitting to such information. Since the image of the lung is located at the center, the cropping edge will not cause any loss of critical information for classification.

4.1.2 Image normalization

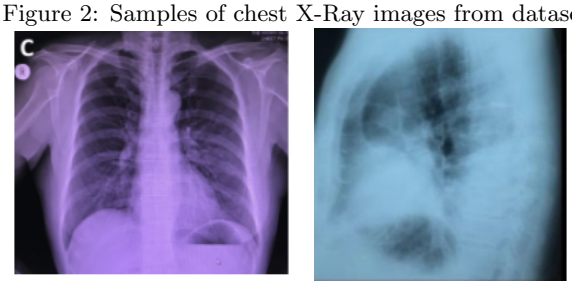
Since we are using pre-trained baseline models, we normalize the X-Ray images with the same mean and standard deviation that were used for training baseline models on the ImageNet dataset.

$$mean = [0.485, 0.456, 0.406], std = [0.229, 0.224, 0.225]$$

4.1.3 Image augmentation

In our COVID-19 dataset, there is a large variance in color and brightness among some X-Ray images, e.g., Figure 2. To reduce the impact on model performance, we first convert the original images to grayscale with three channels, then augment the image dataset by adding random brightness noise and randomly flipping images horizontally at probability 0.5.

Figure 2: Samples of chest X-Ray images from dataset



(a) No findings (b) Pneumonia

4.2 Four Baseline Models

The Convolutional neural network plays an important role in feature extraction for image classification tasks. Instead of constructing a neural network from scratch, we choose 4 standard classification models as the skeleton and perform optimizations on top of that. In this way, we can leverage transfer learning to reduce overfitting when using deep CNN models on small image dataset.

The 4 baseline models used in this work are AlexNet, VGG-16, Inception V3, and ResNet-18. We select these 4 baseline models because each of them has different features to avoid gradient vanishing problems and achieve better classification performance. The network depth is proper for our classification task, we didn't select a very deep network, e.g., ResNet-152, since our image data set is much smaller compared to ImageNet, using a very deep network may lead to over-fitting and greatly increases the training time.

4.2.1 AlexNet:

The AlexNet model [9] is the winner of the 2012 ImageNet ILSVRC challenge It uses five convolutional layers and three max-pooling layers for feature extraction. On the top, three fully connected layers predict classifications.

4.2.2 VGG-16:

Visual Geometry Group (VGG) is a CNN model introduced by Karen Simonyan and Andrew Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition" [10]. It also won the ILSVRC challenge in 2014. VGG16 has 16 layers in total with 13 convolutional layers and 3 fully connected layers. Compared with AlexNet, VGGNet goes more in-depth and has a much smaller filter.

4.2.3 Inception-V3:

Inception-V3 is the 3rd version of the Deep Learning Convolutional Architectures following the popular GoogleLeNet introduced by Szegedy et al., Rethinking the Inception Architecture for Computer Vision (2015) [11]. Inception-V3 is trained to use a dataset of 1,000 classes from the original ImageNet dataset with 48 layers deep. The expected color image input of this model is 299 × 299. Inception-V3 is one of the first algorithms that uses batch normalization. The factorization methods are also used to have more efficient computations.

4.2.4 ResNet-18:

ResNet-18 is a convolutional neural network that is 18 layers deep. Residual neural networks use skip connections to jump over some layers. This can help speed up the training and avoid the problem of vanishing gradients while allowing the network to go deeper.

4.2.5 Baseline models fine-tuning:

We use pre-trained baseline models and fine-tune parameters from all layers on our X-Ray image dataset, During the fine-tuning, all layers are trainable with no weights frozen. For all of the 4 baseline models, we replace the last fully connected layer with a new Linear layer, since the original final layer has 1000 outputs for the ImageNet classification task. In this work, we train models to classify COVID-19 and non-COVID-19 cases, so the output size is 2 in the new Linear layer.

Then, we apply the Softmax function on each model output to convert the linear output to probability y_i and have $y_1 + y_2 + ... + y_c = 1$

$$Softmax(y) = \frac{e^{y_i}}{\sum_{i=1}^{c} e^{y_i}}, i \in C$$

We use the standard Cross-Entropy loss function to calculate the loss. The weights are updated with the stochastic gradient descent method.

$$LossFunc(y, \hat{y}) = \sum_{m=1}^{M} -y_m \log(\hat{y}_m)$$

Baseline models selection: we train each based on a model for 30 epochs, we calculate the model loss on both training and testing set after training on each epoch. An example of the loss trend for training VGG-16 is shown in Figure 4. We select the best model version with the minimum loss on the testing set to avoid over-fitting.

4.3 Ensemble Model

We use Ensemble learning as an approach for further improvement on top of the baseline models, Figure 3 shows the

Figure 3: Overview of Ensemble model

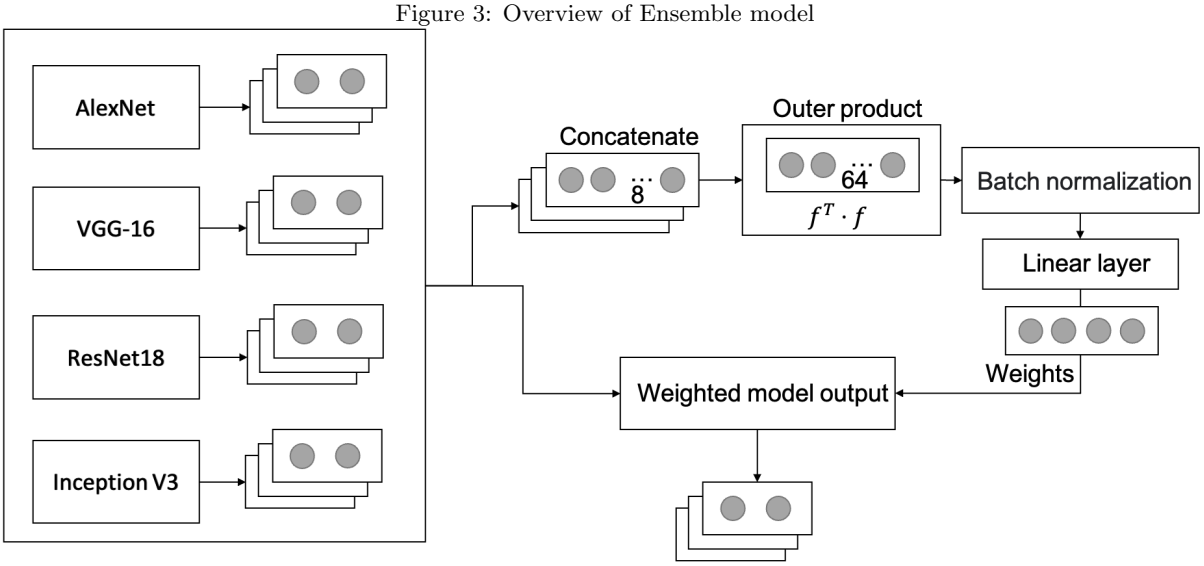
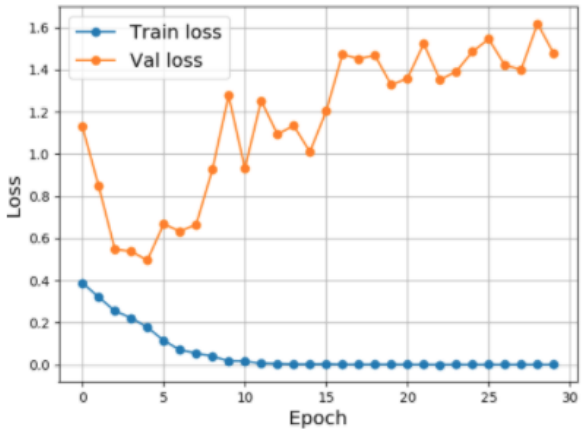


Figure 4: Loss on training and testing set



overview of our Ensemble model. Because each of the baseline models has its advantages in improving classification performance, Ensemble learning can combine these baseline models and increase the robustness and accuracy of the classification from an intuitive perspective. Different baseline models can be diverse and capture different patterns in the same datasets. The ensemble of these baseline models can bring two advantages:

(1) less training time and computational efficiency since the architecture is built on pre-trained baseline models [12] [13]

(2) more generalized model that combines characteristics from different models.

4.3.1 *Ensemble layer*

For Ensemble methods, we didn't use the traditional methods such as averaging, voting. We added a neural ensemble layer that was introduced by Prof. Sun [14] which proves to be effective in improving the model performance compared with individual classifiers.

We build an Ensemble model on top of the 4 baseline models. Instead of directly making predictions on the linear output from each baseline model, we combine them with trainable weights and predict the final classification based on combined outputs [14].

We select the best baseline model versions as mentioned in Section 4.2, the output from each baseline model is a 32×2 tensor, which are concatenated to form a 32×8 tensor f.

4.3.2 *Batch normalization*

The outer product is computed for $f^T f$ to capture latent information across different model outputs. Then the flattened outer product 32×64 is fed into a batch normalization layer [15].

The batch normalization can make neural networks training faster and more stable through normalization of the layers' inputs by re-centering and re-scaling. During training, with weight updates and data variance in different batches, the distribution of input to each layer varies a lot, which can introduce noise and fluctuation to model training. Batch normalization helps mitigate the problem of internal covariate shift, where parameter initialization and changes in the distribution of the inputs of each layer affect the learning rate of the network. In addition, batch normalization also has the effect of regularization that helps mitigate over-fitting.

$$x_i = \mu \cdot \frac{x_i - E[x_i]}{\sqrt{Var(x_i) + \epsilon}} + \beta$$

Then a Linear layer with $Tanh$ activation function calculates the weight for each baseline model.

$$z = w \cdot x + b$$

$$Tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

Finally, we aggregate the output y_m from 4 baseline models

by computing the weighted sum. Where w_m represents the weight from the linear layer for baseline model $m \in M$. M denotes the number of baseline models for the ensemble.

$$\hat{y} = \sum_{m}^{M} \hat{y}_m \cdot w_m$$

The final output is convert to probability with the Softmax function $\hat{y} = Softmax(\hat{y})$.

4.3.3 Loss function: Standard vs Focal

We explored two loss functions:

(1) Standard Cross Entropy function

$$LossFunc(y, \hat{y}) = \sum_{i=1} -y_i \log(\hat{y}_i)$$

(2) Focal loss function [14]

$$LossFunc(y, \hat{y}) = \sum_{i=1} -\alpha_i \cdot y_i \cdot (1 - \hat{y}_i)^{\beta} \log(\hat{y}_i)$$

Where y_i represents the probability prediction of class i. There are two additional coefficient α_i and $(1-\hat{y}_i)^b eta$ based on the cross entropy loss. α_i is a class-specific coefficient that increases the weight on minority class to mitigate the issue of imbalanced training dataset. α_i is set to reciprocal of class frequency, i.e.,

$$\alpha_{covid-19} = \frac{total\_examples}{covid\_examples}$$

in this work, $\alpha_{covid-19}$ is set to 3.4, and α_{other} is set to 1.4. $(1 - \hat{y}_i)^{\beta}$ increases the weight for examples with inaccurate predictions, β is set to 2 based on grid tuning.

5. EXPERIMENTAL EVALUATION

To compare the result among the four baseline models, the dataset is split into the train (80%) and testing (20%) groups, and compare the model performance on testing data.

Metrics for performance evaluation:
For the performance evaluation of four baseline models and the ensemble model, the metrics used in this paper were accuracy, ROC-AUC, precision, recall, and F1-score. The formulas of these indicators are shown below.

$$Accuracy = \frac{TN + TP}{TP + FP + TN + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

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

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

where TP stands for True Positive, which measures the actual COVID-19 cases that are successfully predicted. TN stands for True Negative, which measures the actual non-COVID-19 cases that are successfully predicted. FP stands for False Positive, which shows the cases that should be Non-COVID-19 cases but incorrectly classified as COVID-19 cases by model. FN stands for False Negative, which means the model classifies COVID-19 cases as non-COVID-19 cases incorrectly.

6. EXPERIMENT

6.1 Experiment setting

We split the entire X-Ray dataset into 2 parts, 80% of total X-Ray images from the combined datasets will be randomly chosen as training data. The rest of the 20% of images will be reserved for testing data. During model training, each mini-batch contains 32 images, and we use the stochastic gradient descent method, with learning rate set to 0.001 and momentum set to 0.9.

Google Colaboratory (Google Colab) with Python is used for implementing all of the models and algorithms. The project colab is set up and is accessed by all of the team members. Everyone can contribute their ideas and improve models in the shared codes.

We also use PyTorch framework in the local environment and train our models on Workstation with:

CPU: Intel® Core™ i9-9900K CPU @ 3.60GHz × 16

GPU: NVIDIA GeForce RTX 2080 Ti/PCIe/SSE2

6.2 Experiment results

The table below summarizes the performance from 4 baseline models and 2 Ensemble model variants on testing data.

In the regular Ensemble model, we only applied the ensemble layer with standard cross-entropy loss.

While in the Ensemble model v1, we also applied Focal loss and performed image augmentation as mentioned in Section 4.1, including grayscale conversion and randomly horizontally flipping.

Table 2: COVID-19 Metrics

	Accuracy	AUC	Precision	Recall	F1
AlexNet	0.811	0.790	0.724	0.724	0.724
VGG16	0.787	0.811	0.636	0.888	0.741
Inception V3	0.749	0.767	0.596	0.828	0.693
ResNet18	0.772	0.755	0.659	0.698	0.678
Ensemble model	0.825	0.838	0.694	0.879	0.776
Ensemble model V1	0.837	0.843	0.719	0.862	0.784

Table 3: Non-COVID-19 Metrics

	Accuracy	AUC	Precision	Recall	F1
AlexNet	0.811	0.790	0.856	0.856	0.856
VGG16	0.787	0.811	0.926	0.734	0.819
Inception V3	0.749	0.767	0.887	0.707	0.787
ResNet18	0.772	0.755	0.837	0.811	0.824
Ensemble model	0.825	0.838	0.927	0.797	0.857
Ensemble model V1	0.837	0.843	0.920	0.824	0.869

From the results, we can see the ensemble model achieved 4.7% F1 score improvement based on the best individual

classifiers of VGG-16 on COVID-19 classification. Furthermore, the Ensemble model v1 achieved 5.8% performance improvement.

7. DISCUSSION

While doing this project, we have encountered few challenges. We will talk about those challenges in the following subsections.

7.1 Steep learning curve on deep learning

One of the biggest challenges of this project is all of the team members do not have enough industrial experience in deep learning. All of the experiences are from the school assignments and lectures. When we do the real project, we don't have a clear direction on how to achieve the final goal. Also, we lack some pointers about how to make this project like a real-world application.

7.2 Steep learning curve on academic paper

The second challenge we have faced is we have no experience in writing a publishable paper. We don't have a clear idea about how to get started to do the research and combine all of the ideas to make the final paper publishable. For example, how to find the math/hyperparameters so that the final accuracy will be improved. What is the upper limit of the accuracy? Should we keep trying different hyperparameter combinations to compare the accuracy among them? What will be the short path to know what we should try? The last challenge we have faced is training the models with different hyperparameters will take a long time. For example, we have four baseline models to compare. Each model will have many hyperparameters to try. To get a better accuracy result, each run will take between two to three hours to get the final result. It is almost impossible to try all of the hyperparameters combinations we originally want to try because we only have few weeks.

Using the paper template of Microsoft word to write this paper is extremely easy. All of us can import the template recommend by the professor into Google Doc and we can write and edit the paper at the same time. However, the template of MS word has some formatting issues that TA points out. To eliminate the barrier, our team decide to use LaTeX instead of using the Google Doc. All of us do not have a strong background in using LaTeX. We have spent some time to figuring out how to use Latex and how to adjust the format in LaTeX, etc.

7.3 Similarity of X-Ray images

The last challenge we have faced is we use one dataset with all X-Rays images having the same position in the very beginning. This causes the accuracy to become extremely high and leaves us no room for making any improvements. To solve this problem, we have increased the dataset with chest X-Ray taken in different positions to lower the accuracy of the models. Then, we have tried weighing the dataset. However, there is no significant improvement. Finally, we have tried Ensemble models.

8. CONCLUSION

The coronavirus disease 2019 (COVID-19) has tremendously spread to the whole world within a few months, which causes the severe global public health crisis in history. Almost every country in the world has been facing a shortage of medical supplies. Therefore, applying AI and big data techniques on X-Ray images can be very helpful to do the low-cost early detection and diagnosis of COVID-19.

The motivation of this work is to solve the problems of differentiating the similar X-Ray images between COVID-19 and non-COVID-19 images. To solve the problems, we choose four standard baseline models and try to make some improvements on these models with our dataset.

By combining different X-Ray datasets from different sources, we get the X-Ray dataset with COVID-19 and non-COVID-19 cases. Because of the variances of the dataset, we preprocess the dataset before training. We choose the best version of the each of four baseline models and combine them by using ensemble learning. Further improvements include applying the Focal Loss function and performing data augmentation to address the imbalance categories in the dataset.

We can see the standard ensemble model achieved 4.7% F1 score improvement based on the best individual classifiers of VGG-16 on COVID-19 classification. Furthermore, the ensemble model V1 with focal loss functions and image augmentation achieves 5.8% F1 score improvement on COVID-19 classification. It indicates that the V1 model performs better after addressing the problems of the dataset.

9. OPTIMIZATION

Due to limited time and computing resources, we could not perform comprehensive hyper-parameter tuning. For further improvements, we can try cross validations for hyperparameter optimization, including tuning for learning rate, batch size, etc. to see if we can further improve models performance. Further work includes comparing different ensemble methods, e.g., averaging, voting, etc.

10. CONTRIBUTION

Niliang Lu:: She trains the ResNet-18 model with different parameters and records the results. She revises four baseline models for COVID-19 image dataset, implements the ensemble model with the PyTorch framework, including the Focal loss function, neutral network skeleton, image preprocessing. She sets up a training environment with GPU and conducts training, collects and calculates metrics for evaluating model performance. She also contributes to the parts of the related work in the paper.

Jiayu Peng: She does the related background search that the CNN model has great performance on the COVID-19 diagnosis and finds the related deep learning approaches on COVID-19 X-Ray images that can be the reference to the project model and optimized. She implements the AlexNet model with different parameters and records the results. She is also responsible for writing the background and performance of AlexNet, graphing the results, and related work sections in the paper.

Jinggong Zheng: He sets up the cloud development en-

vironment such as Google Colab and Google Drive for the dataset storage to get it ready to train the four baseline models. Apart from the development environment setup, he splits the dataset into the train and the evaluate dataset. He creates different versions of the train dataset for measuring the performance of four baseline models. He takes care of measuring the ResNet18 performance test and comparing the accuracy between different hyper-params. Besides he also responsible for writing the ResNet18 section of this paper and he also contributes to the part of introduction section and few other sections in this paper.

Pui Sze Ng: She analyzes the research papers about COVID-19 X-Ray images and deep learning. She also analyzes related X-Ray images datasets. She utilizes the dataset, that Jinggong has been cleaned, to implement the VGG model with Python. She runs the VGG model and records the performance of this model with different parameters such as a number of epochs, different loss functions, and different optimizers. Besides, she acts as a note-taker during each meeting. She is responsible for writing the background and performance of the VGG model. She also writes part of the introduction, abstract, and discussion.

11. REFERENCES

[1] "Covid-19". O. e. (online ed.). oxford university press. Available at `http://https://www.oed.com/view/Entry/88575495` (2021/04/12).

[2] CDC COVID-19 Response Team, CDC COVID-19 Response Team, CDC COVID-19 Response Team, Stephanie Bialek, Ellen Boundy, Virginia Bowen, Nancy Chow, Amanda Cohn, Nicole Dowling, Sascha Ellington, et al. Severe outcomes among patients with coronavirus disease 2019 (covid-19)—united states, february 12–march 16, 2020. *Morbidity and mortality weekly report*, 69(12):343–346, 2020.

[3] Coronavirus Resource Center. Hopkins, j. university and medicine. Available at `https://coronavirus.jhu.edu/map.html` (2020).

[4] Michael T Lu, Alexander Ivanov, Thomas Mayrhofer, Ahmed Hosny, Hugo JWL Aerts, and Udo Hoffmann. Deep learning to assess long-term mortality from chest radiographs. *JAMA network open*, 2(7):e197416–e197416, 2019.

[5] Mustafa Ghaderzadeh and Farkhondeh Asadi. Deep learning in detection and diagnosis of covid-19 using radiology modalities: A systematic review. *arXiv preprint arXiv:2012.11577*, 2020.

[6] Shruti Jadon. Covid-19 detection from scarce chest x-ray image data using few-shot deep learning approach. In *Medical Imaging 2021: Imaging Informatics for Healthcare, Research, and Applications*, volume 11601, page 116010X. International Society for Optics and Photonics, 2021.

[7] Tulin Ozturk, Muhammed Talo, Eylul Azra Yildirim, Ulas Baran Baloglu, Ozal Yildirim, and U Rajendra Acharya. Automated detection of covid-19 cases using deep neural networks with x-ray images. *Computers in biology and medicine*, 121:103792, 2020.

[8] Joseph Paul Cohen, Paul Morrison, Lan Dao, Karsten Roth, Tim Q Duong, and Marzyeh Ghassemi. Covid-19 image data collection: Prospective predictions are the future. *arXiv 2006.11988*, 2020.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[10] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[11] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[12] Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, 12(10):993–1001, 1990.

[13] AMANDA J C SHARKEY. On combining artificial neural nets. *Connection science*, 8(3-4):299–314, 1996.

[14] Zhi Qiao, Austin Bae, Lucas M Glass, Cao Xiao, and Jimeng Sun. Flannel (focal loss based neural network ensemble) for covid-19 detection. *Journal of the American Medical Informatics Association*, 28(3):444–452, 2021.

[15] Christian Szegedy Sergey Ioffe. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *https://arxiv.org/abs/1502.03167*, 2015.

12. ABOUT THE AUTHORS:

NILIANG LU is a graduate student of Master of Computer Science program University of Illinois - Urbana Champaign. She received her B.S in Accounting at Shanghai University of Finance and Economics.

JIAYU PENG is a graduate student of Computer Science program at University of Illinois - Urbana Champaign. She had a B.A. in Computer Science at University of Washington, Tacoma.

JINGGONG ZHENG is a software engineer focused on developing big data applications in Amobee. He currently takes care of the data ingestion pipeline and manages workflows in Apache Airflow for the reporting team. He earned a B.S from the Computer Science Department of The University of Utah.

PUI SZE NG received her B.A. in Mathematics and B.S. in Economics at University of Washington. She also worked as a Computer Scientist intern at Air Force Civilian Service last summer. She is currently working on her Master degree in Computer Science at University of Illinois - Urbana Champaign.