Machine Learning Approaches for Detecting Pneumonia by Analyzing Chest X-Ray Images

Weidong Gao (wg222), Yufei Wang (yw569), Yijie Yang (yy586)

Abstract—**Pneumonia is one of the most wide-spread diseases that may result in death and diagnosing pneumonia in the clinical practices is critically expensive and time-consuming. We hypothesized that machine learning can help with the task, as machine learning and deep learning models these years have demonstrated promising performance in facilitating medical image processing. We selected an open accessed pneumonia dataset on Kaggle, implemented and evaluated a variety of machine learning models upon it. We also adopted a range of techniques to tune the models. We evaluated algorithms using 4 metrics: accuracy, precision, recall, and f1-score. We will particularly focus on the recall and the f1-score due to pneumonia's severity. Due to the imbalance of the data, SVM as a traditional machine learning algorithm has the worst performance: an accuracy of 0.734, a precision of 0.708, a recall of 0.977 and an f1-score of 0.8211. By addressing the problem of imbalanced data, CNN and transfer learning from pre-trained models outperformed SVM. The performance of CNN is 0.921, 0.893, 0.992, 0.940. Using Inception V3 model in transfer learning has the best performance: 0.967, 0.970, 0965, 0.967. Based on our experimental results, we found that machine learning, especially deep neural networks have great potential in medical imaging.**

Index Terms—**Convolutional Neural Network, Deep Learning, Transfer Learning, Pneumonia, Medical Image Processing**

I. Introduction

Pneumonia remains one of the most lethal and wide-spread diseases. To detect Pneumonia, chest X-ray images are routinely checked by skilled experts in the clinical practices, which can be both time and financially costly. With the advances in machine learning and image processing techniques, emerging applications of autonomous medical image classification have demonstrated the great potential of machine learning in the field. Our goal is thus to apply machine learning methods to classify pneumonia patients according to chest X-ray images.

Our work is based on the Chest X-Ray Images (Pneumonia) dataset on Kaggle [7]. We perform tuning, data augmentation and regularization methods over our own convolutional neural network and pre-trained deep learning models to improve their performance. We then compare our results to previous work on the Chest X-Ray dataset in the Discussion and Prior Work section. The CNN model we designed illustrated an accuracy of 92.1% on the test data set while the deep CNN models we trained in a transfer learning fashion resulted in an accuracy of 96.7%, which both over performed the existing works focusing on this dataset. Our main accomplishments in the current project can be summarized as follows:

- We explored, preprocessed and rearranged the dataset, tackled the major problems lying beneath the data set, so that we can obtain better model performance and interoperability from the dataset.
- We crafted a Convolutional Neural Network (CNN) based on the insights from the dataset and the techniques we learned from the lectures, which shows better performance than most of CNN models available on the Kaggle community.
- We explored the potential of transfer learning in the context of medical image processing. Our experiment reveals that fine-tuning on pretrained Deep Convolutional Neural network (D-CNN) can significantly improve the model performance.

II. Background

A. Pneumonia

Pneumonia is an infection of lungs that can be life-threatening. According to WHO, in 2017, 15% of all deaths of children under 5 years old are due to pneumonia, and the situation in South Asia and sub-Saharan Africa is the severest due to the huge gap between the numbers of patients and doctors.

COVID-19 has infected over 219 million people and over 4.55 million people are dead due to this pandemic. According to Million, et al, an early treatment can significantly reduce the fatality rate in patients [1]. Therefore, pneumonia needs to be diagnosed as early as possible. Therefore, it is worth developing a reliable tool for pneumonia detection, which can provide service for more people and reduce the cost of human labor.

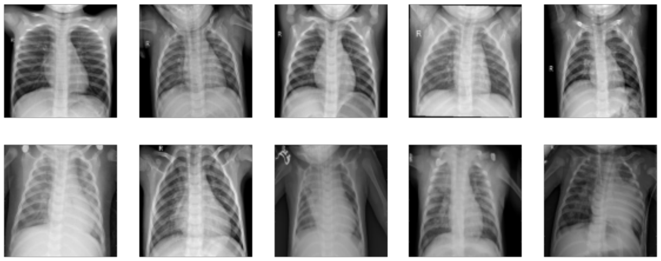


Fig. 1: The first row are X-ray images of normal chests; the second row are X-ray images of chests infected by Pneumonia

B. Deep Learning in Medical Imaging

Deep learning, as one of the most popular technologies to process images and to predict categorical results, haved played essential roles in medical areas. For example, Yuexiang Li et al. [2] proposed a deep learning framework consisting of two fully convolutional residual networks to detect melanoma in dermoscopy images. The deep learning method has an accuracy of 0.848 for melanoma classification, providing a fast and reliable way to detect melanoma in an early stage. Ezz El-Din Hemdan et al. [3] built a new deep learning framework (COVIDX-Net) consisting of popular deep learning models such as VGG19 and ResNetV2 to assist the early diagnosis of COVID-19 in X-ray images. In their experiment, VGG19 and DenseNet201 models have an accuracy of 0.9. It proves that deep learning models can be a useful tool to help detect COVID19. Dayong Wang et al. [4] proposed a deep convolutional neural network to identify metastatic breast cancer. The result shows that with the help of a deep learning model, the human pathologists' diagnoses achieve an area under the receiver operating curve of 0.995. It shows that deep learning has the potential to help increase the accuracy of pathological diagnoses. Deep learning can ensure a high accuracy of detecting disease and minimizes the costs of capital and time. Therefore, we adopt deep learning in our project to detect pneumonia in chest X-ray images.

III. Method

A. SVM

Firstly we adopt the support vector machine(SVM) as our baseline to compare with our neural network classifiers. It is a popular machine learning method prior to neural networks and can be applied on image classification. SVM will generate hyperplanes that will separate images into two classes(Pneumonia or normal) with maximum segregation between the hyperplane and image feature vectors. The hyperplanes are built according to a kernel function. We apply the RBF kernel and soft margin since they can handle non-linear relationships between class labels and extracted features.

B. CNN

Convolutional Neural Network (CNN) is a deep neural network used in image recognition and processing. CNNs are especially useful in processing pixel data. A typical CNN contains several convolutional layers, activation functions, pooling layers, dropout layers, normalization layers, and fully connected layers. The details of different types of layers are defined as below.

1) Convolutional layer: Convolutional layer is used to extract regional features from input matrices. Each convolutional layer will produce a set of 2D matrices representing the local features extracted from the previous matrices using kernel matrices. Each input matrix f is convoluted with a kernel k to produce a new matrix based on the following equation:

$$y[m,n] = f[m,n] \times k[m,n] = \sum_j \sum_i f[i,j] \times k[m-i, n-j], \tag{1}$$

where y is the new matrix, m, n are the coordinates of the new matrix, f is the input matrix and k is the kernel matrix. The number of channels in the input and output can be different by setting the number of kernel matrices. The aim of setting a different number of kernel matrices is to increase the number of local features to improve the accuracy of the model.

2) ReLU Activation: Activation function is used to map the resulting values in a specific range. Here, we use the ReLU activation function in our neural network. The ReLU activation function will map the resulting values in between 0 and infinity as the following equation,

$$ReLU(z) = max(0, z) \tag{2}$$

There are two main advantages of using ReLu as the activation function. First, the resulting matrix from ReLU is sparse, which can accelerate the training process. Second, it effectively reduces the likelihood of the gradient to vanish. When z>0, the gradient is a constant value. While constant value can prevent the gradient from vanishing, it can also accelerate the learning process.

3) Max-Pooling layer: Max pooling is used to reduce the image size by selecting the max value in neighboring elements of the input matrix. Except for reducing the image size, using max pooling can also solve the problem of overfitting.

$$MaxPool(A) = max(A), \tag{3}$$

where A is the input matrix.

4) Dropout Layer: Dropout algorithm is used to improve the performance by probabilistically dropping the weights in each layer. During training, some layer outputs are randomly dropped out, which means it loses the connection to the next layer. This is an effective regularization method to reduce overfitting.

5) Batch Normalization: Batch normalization is a common method used in neural networks, especially in CNNs, to accelerate the training process and get a more stable and accurate result through re-centering and re-scaling each mini batch. Batch normalization has the ability to address many problems in training deep neural networks, including reducing sensitivity to the choice of weight initialization, increasing the stability of neural networks, and mitigating the problem of internal covariate shift.

6) Fully connected layer: Fully connected layer is one of the basic types of layers in deep learning. Fully connected layer is a layer in which every neuron in one layer is connected to every neuron in another layer. The structure is the same as the Multi-layer perceptron. In CNNs, it is usually used at the end of the neural networks to output the final result and classify the images.

$$y_j^{(k)}(x) = f(\sum_{i=0}^{n} w_j^{(k)} x^{(i)} + w_j^{(0)}), \tag{4}$$

where j is the j^{th} layer, k is the k^{th} weight, and i represents the i^{th} input.

C. Transfer Learning

One of the major challenges of utilizing deep learning models in the field of healthcare, particularly medical imaging, is the lack of training data. Medical images are only accessible when patients voluntarily go to medical institutions for screening, and the collections have to be performed by experts. To tackle this drawback, transfer learning (TL) has been widely deployed in medical imaging tasks by using pre-trained state-of-the-art models from the ImageNet dataset. It is also proved via experiment by previous works that Deep Convolutional Neural Network (D-CNN) trained on ImageNet dataset [12], which comprises mainly natural images, can be utilized to enhance the performance of the medical image classification task [10]. In this project, we will tune pre-trained deep learning models such as DenseNet, VGG Net, ResNet, and Inception Net to test their performance on classification of pneumonia images.

1) VGG Net: VGG net is a CNN model proposed by Simonyan and Zisserman in 2015 [13], which is symbolized by the stacked 3×3 convolutional layers stacked on top of each other in increasing depth.

2) ResNet: The ResNet model was proposed in 2015 by He et al [14]. It is motivated by the inability of multiple non-linear layers to learn identity mappings and degradation problem.

3) DenseNet: Huang et al. in 2016 [15] introduced a densely connected convolutional network architecture. All layers in DenseNet are connected directly with each other in a feed-forward manner to ensure maximum information flow between layers.

4) Inception V3: Inception v3 is an updated version of the inception architecture proposed by Szegedy, et al [16]. It is made up of symmetric and asymmetric building blocks, and Batch norm is used extensively throughout the model and applied to activation inputs.

5) Fine-tuning Layers: Fine-tuning is an important step in transfer learning. During transfer learning, we froze the weights of each D-CNN pretrained on ImageNet, and add some fresh layers for our target task. In order to make the performance comparable, in the experiment we adopted the same structure of fine-tuning layers for each model.

IV. Experimental analysis

A. Dataset

Our project is mainly based on the Kaggle dataset Chest X-Ray Images (Pneumonia) [7], which consists of in total 5863 chest X-Ray images. The X-ray images were selected from retrospective cohorts of pediatric patients of one to five years old from Guangzhou Women and Children's Medical Center, China. These images are labeled by two classes, i.e., either Normal or Pneumonia. We selected this dataset due to its considerably large size, well controlled image quality, and trustworthy labeling.

After investigating the dataset, we featured some of the major characteristics of the dataset, which will likely affect our choices of preprocessing techniques and machine learning models.

	Training	Validation	Testing
Original dataset	5216	16	624

TABLE I: Original dataset splits

1) Imbalance: We found the dataset highly imbalanced in terms of the pathological group and normal group, and this issue is particularly serious for the training dataset. To tackle this problem, we plan to use a weighted loss function in our deep learning models, which will mitigate the bias toward the dominating class (Pneumonia).

	Training	Validation	Testing
Normal	0.26	0.50	0.38
Pneumonia	0.74	0.50	0.62

TABLE II: Class Distribution in Each Dataset

2) Oversize: We randomly selected 9 images from the target dataset and found that although there are variability in the image size, the typical image sizes are above (500 * 800). Considering the number of images we have in our training dataset and the computing resource we have, it would be unrealistic to retain the original image sizes in training the model. Therefore, we decided to downsize the image to (120, 120) for SVM and (150 * 150) for CNN and transfer learning.

B. Metrics

We monitored 4 metrics to measure the performance of our model: accuracy, precision, recall and F1 score. We define True Positive (TP) as correctly classifying PNEUMONIA as PNEUMONIA, True Negative (TN) as correctly classifying NORMAL as NORMAL, False Positive (FP) as misclassifying NORMAL as PNEUMONIA, and False Negative (FN) as misclassifying PNEUMONIA as NORMAL. The 4 metrics are calculated as below:

$$\begin{aligned} Accuracy &= \frac{TP + TN}{TP + TN + FP + FN} \\ Precision &= \frac{TP}{TP + FP} \\ Recall &= \frac{TP}{TP + FN} \\ F1\ score &= 2 \cdot \frac{precision \cdot recall}{precision + recall} \end{aligned} \tag{5}$$

Although we monitor 4 metrics in our case, we focused more on F1 score since we had an imbalanced dataset. It is not meaningful to analyze the accuracy of a model trained on an imbalanced data, since the model can still achieve a high accuracy although the model predict all samples in the minority class as the majority class in binary classification. So F1 score is a more meaningful metric to monitor and analyze. Recall is another metric that we want to emphasize in the analysis. Due to the high contagion and the fatality of pneumonia, we want to reduce the number of misclassifications in the class PNEUMONIA. Accuracy and precision are used to help us to determine how well the model trained.

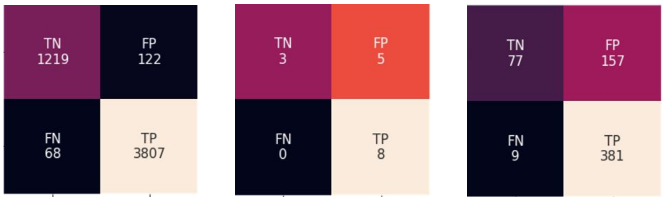
C. Hardware and Software

The experiments were performed mainly on Google Colab, where we can easily co-worked on the code, and GPUs are provided to accelerate our training process. As for software, we implemented our experiments from data preprocessing, training to evaluation purely in Python. We mainly used *Tensorflow Keras* to implement the deep neural networks.

D. SVM

We first normalize each image since regression models prefer floating point values within a smaller range. We then flatten all images to see the vertical structure of the images and train the SVM. As a baseline, we implement an soft margin SVM with default c = 1.0 and apply the RBF kernel to handle non-linear relationships between class labels and features. Due to the RAM limit of colab, we only considered the first 14400 pixels for each training image. Same downsampling was also applied to the validation and testing data.

After preprocessing all images, we first trained the SVM model with the 5216 images from the training dataset and evaluated the model on the validation set with 16 images(8 for abnormal(with Pneumonia), 8 for Normal). We then built their confusion matrix to calculate their precision, recall and F1-score.



(a) Training (b) Validation (c) Testing

Precision, recall and F1-score on the validation dataset are as follows: 0.6154, 1.0 and 0.7619. Regarding the training dataset, they are 0.9689, 0.9825, 0.9757. Precision and F1-score drops significantly in the validation set. Since our training dataset is imbalanced with more abnormal data(74 percent of the total), the sudden performance drop shows it fails to generalize on the balanced validation dataset. We then calculated the metrics over the testing dataset so that we have a baseline to compare with CNN and Transfer Learning NNs models. The SVM achieved a 0.7340 testing accuracy. The Precision, recall and F1-score are 0.7082, 0.9769 and 0.8211.

E. CNN

1) Data Load and Preprocess: In order to compare with different previous work, we used two different dataset splits: keep the original splits and rearrange the entire dataset.

- **Modified dataset splits**
  Since we only have 16 images in the validation set (8 for PNEUMONIA, 8 FOR NORMAL), we randomly selected 192 PNEUMONIA images and 192 NORMAL images from the train set to form a balanced validation set which has 400 images in total (200 for PNEUMONIA, 200 FOR NORMAL). But we did not modify the original test set. Therefore, the train set contains 4832 images, the validation set contains 400 images, and the test set contains 624 images.

	Training	Validation	Testing (unchanged)
Normal	1149	200	234
Pneumonia	3683	200	390

TABLE III: The distribution of the **modified** dataset

- **Rearranged dataset splits**
  In order to compare with the previous work conducted by Stephen et al. [8]. We rearranged the entire dataset into three sets: train set, validation set and test set. In order to compare the accuracy with Stephen el al. 's result, we decided to make a test set with 2134 images. The train set contains 2977 images and the validation set contains 745 images.

	Training	Validation	Testing
Normal	814	195	574
Pneumonia	2163	550	1560

TABLE IV: The distribution of the **rearranged** dataset

We employed several data augmentation methods on the train set to preprocess the X-ray images to deal with overfitting and dataset imbalance. During the loading process, the images are resized to (150, 150) to accelerate the training process and reduce the problem of overfitting. First, we rescaled the pixel values from (0, 255) to (0.0, 1.0) to accelerate the training process. The rest of the augmentation procedures are randomly rotating the images in the range of 5 degrees, randomly zooming image 10%, randomly shifting images horizontally 10%, and randomly shifting images vertically 10%.

2) CNN structures: The following figure shows the architecture of the proposed CNN model. The CNN model consists of two parts: feature extractors and a classifier. The feature extractors consist of 5 convolutional layers: (conv, 3x3, 32), (conv, 3x3, 64), (conv, 3x3, 128), (conv, 3x3, 128), and (conv, 3x3, 256); each convolutional layer is followed by a ReLU activation function, a batch normalization layer and a max-pooling layer with a kernel size of 2 and a stride of 2; except for the first one, the other 4 convolutional layers have dropout layers with dropout rates of 0.1, 0.15, 0.2, 0.2, after the max-pooling layer. The classifier is placed at the end of the neural network. Before the classifier, a flatten layer is used to reshape the features from the feature extractor to a shape of (batch size x 256 x 5 x 5, 1) as the input of the fully connected layer. The classifier contains two fully connected layers with output shapes of 128 and 1, using ReLU and sigmoid activation functions, respectively. The output of the model will be a float value, if the float value is larger than 0.5, then the image is classified as PNEUMONIA, otherwise the image is classified as NORMAL.

3) Hyper-parameters selection:

- **Batch size**

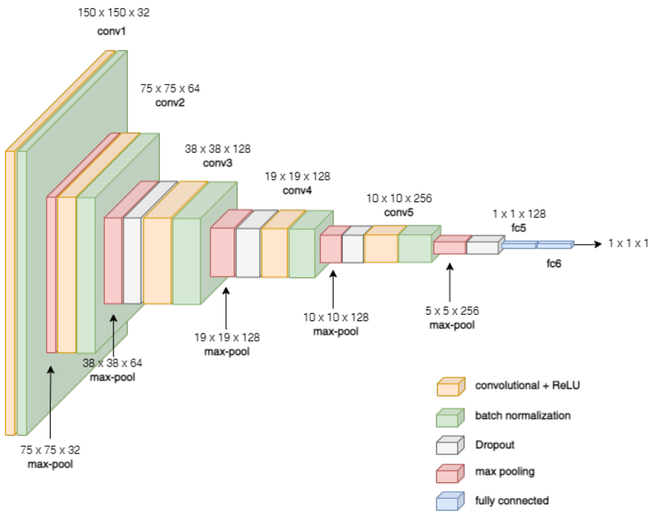


Fig. 3: The architecture of the proposed CNN

Since our dataset is an imbalanced dataset, it is necessary to choose a large batch size so that for each batch, it will at least contain several images that belong to the NORMAL category. So we selected a batch size of 32.

- **Optimizer, Learning rate, Epochs, Regularization**
We used Adam optimizer since it is one of the most popular optimizers used massively in CNNs. Instead of a constant value of learning rate, we used a reduced learning rate. The learning rate starts with 0.0001. The model monitors the accuracy of the train set. If train accuracy has not improved for 2 epochs, the learning rate will be reduced by a factor of 2. In order to reduce the probability of overfitting, we associated the early stopping technique with it. The early stopping technique monitors the accuracy of the validation set. If the validation accuracy has no improvement for 10 epochs, then the training process will stop.
However, we only applied the early stopping technique on the rearranged dataset splits. We can't use the early stopping technique on the original dataset splits since the validation set does not reflect the data imbalance in the train set. We use an epoch of 50 to train the model on the original dataset splits.
- **Loss function**
Instead of binary cross entropy, we used weighted binary cross entropy. Weighted binary cross entropy makes the model focus more on a specific class by penalizing more on the misclassification of that class. There are two reasons we used this loss function. First, since we have an imbalanced dataset, we need to use weighted binary cross entropy to help reduce the problem resulting from the imbalanced dataset. Second, as pneumonia is a highly contagious and deadly disease, we want to detect pneumonia as much as possible.

$$loss = \frac{1}{n}\sum_{i=1}^{n} -(w_0(1-y^{(i)})(1-y^{(i)}log\hat{y}^{(i)}) + w_1 y^{(i)}log\hat{y}^{(i)}), \quad (6)$$

where $\hat{y}^{(i)}$ is the predicted label for the i^{th} image, $y^{(i)}$ is the true label for the i^{th} image, w_0 is the weight for class 0 (class NORMAL in our case), and w_1 is the weight for class 1 (class PNEUMONIA in our case).
Since the majority of images are of the class PNEUMONIA, the model will predict more PNEUMONIA without data augmentation and the weighted loss function. In this case, we focused more on increasing the F1-score of the model since both precision and recall are significant in our case. We used a class weight of w_0 = 3.53 and w_1 = 1.4 for modified dataset and w_0 = 3.06 and w_1 = 1.2 for rearranged dataset. The details of weight calculation is discussed in the "How we tuned CNN" part in the Appendix B.

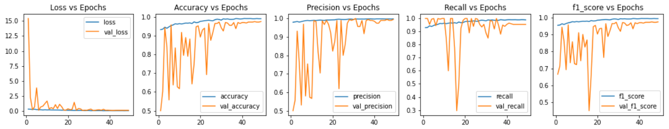


Fig. 4: The Change of Performance Indicators in Training (Proposed CNN, Modified dataset)

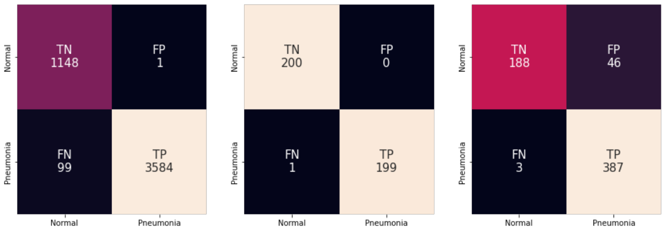


Fig. 5: The Confusion Matrices (Proposed CNN, Modified dataset)

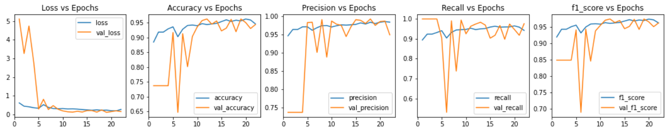


Fig. 6: The Confusion Matrices (Proposed CNN, Rearranged dataset)

4) Results and Analysis: From the loss figure, we can know that our model was trained well, free of overfitting. As the training process continues, four metrics (accuracy, precision, recall and f1 score) increase and loss decreases. While plots of the modified dataset from the validation set fluctuate a lot, plots from the rearranged dataset have less fluctuation than

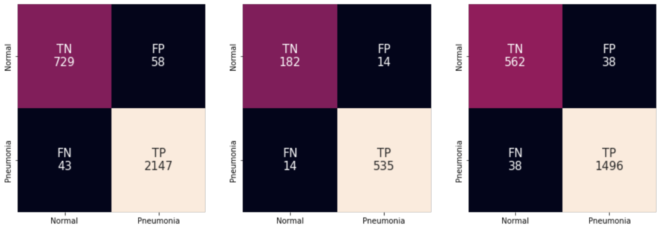


Fig. 7: The Confusion Matrices (Proposed CNN, Rearranged dataset)

TABLE V: Performance Comparison of Different Deep-CNNs

Dataset		Accuracy	Precision	Recall	F1-Score
Modified	Training	0.980	0.999	0.973	0.986
	Testing	0.921	0.893	0.992	0.940
Rearranged	Training	0.966	0.973	0.980	0.977
	Testing	0.964	0.975	0.955	0.975

TABLE VI: Performance Comparison of Different Deep-CNNs

Model		Accuracy	Precision	Recall	F1-Score
ResNet	Training	0.970	0.980	0.981	0.980
	Validation	0.955	0.960	0.950	0.955
DenseNet	Training	0.965	0.978	0.976	0.977
	Validation	0.957	0.960	0.955	0.957
VGG Net	Training	0.959	0.974	0.973	0.973
	Validation	0.955	0.946	0.965	0.955
Inception Net	Training	0.958	0.974	0.971	0.973
	Validation	0.967	0.970	0.965	0.967

those from the modified dataset. The validation set of the rearranged dataset reflects the imbalanced data in the training set. Therefore, the model had a better performance on the rearranged validation set. In terms of the rearranged dataset fluctuations, we think that one of the reasons is due to the imbalanced data. In imbalanced data, it is inevitable that some batches may not contain the minority class (class NORMAL). The model will then only learn from the less informative batch, which will lead to a bias to the majority class (class PNEUMONIA).

F. *Transfer Learning*

We applied four state-of-art deep neural networks designed for image processing, including ResNet, DenseNet, VGG Net, and Inception Net to test transfer learning performance on the dataset. We selected these four models mainly because they have been proved by previous work to have promising performance when they are trained in a transfer learning fashion. We incorporated these neural networks with frozen weights pre-trained on Image-Net [12], and added the same additional layers to all of them. After that, we trained the model on the original kaggle dataset with the partitions kept the same as the one we used for CNN training (4832 images for training, 400 images for validation, and 624 images for testing). By comparing their performance to each other, and to the performance of our customized CNN model, we are able to grasp a taste of how much our problem can benefit from transfer learning. During the training process, we kept track of a variety of performance indicators that we thought would be important to the task of Pneumonia detection. After we completed the training, we obtained the confusion matrices for each model on the testing dataset. We applied early stopping as we did in training the proposed CNN model, while we set the monitoring indicator to be the validation accuracy. We found all of the fine-tuned models were stopped within 15 epochs, showing fast convergence. While the learning curve fluctuates in some cases, the result models consistently have high performance in detecting Pneumonia cases (with accuracy, precision, recall, and f1-score all above 95%).

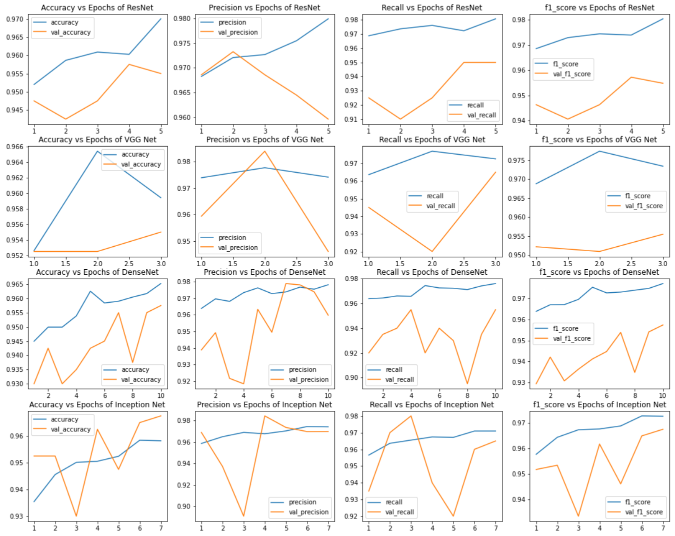


Fig. 8: The Change of Performance Indicators in Training

V. Discussion and Prior Work

A. *Prior Work Revisit*

This dataset was collected and labeled by Kermany et al. [5] in 2018. They applied a transfer learning model to classify whether the X-ray images belonged to patients with pneumonia or not. They used an Inception V3 architecture [6] pretrained on the ImageNet dataset as the model for transfer learning. Through retraining the final, softmax layer to make the model specialize in classifying pneumonia. They achieved an accuracy of 92.8%, with a sensitivity of 93.2% and a specificity of 90.1%. In 2019, Stephen et al. [8] used a convolutional neural network for this work. Instead of using the provided dataset splits, they rearranged the entire dataset into the train set and the validation set. A total of 3722 images were allocated to the training set and 2134 images were allocated to the validation set. They deployed a

CNN containing 4 convolutional layers with ReLU activation function, max-pooling layers and 2 fully connected layers with sigmoid activation function. Before training the model, they applied data augmentation on the images, such as random image rotation and vertical translation. They achieved a test accuracy of 93.73%.

Since the prior work used accuracy, sensitivity and specificity as their metrics, which is different from our choices, we will only compare accuracy. Compared with Kermany et al.'s transfer learning method, we used data augmentation and added more layers after the pre-trained models to extract features. In terms of the same pre-trained model: Inception V3, the performance of our transfer learning, 96.7%, outperformed their transfer learning, 92.8%. This proves that data augmentation and adding fine-tuned layers can improve the classification performance. In order to compare our CNN model with Stephen et al. 's CNN model, we split the dataset into train, validation and test sets and the size of the test set is the same as theirs. The accuracy of our model is 96.4%, which is higher than their accuracy, 93.73%. This proves that we successfully reduce the overfitting of our model and mitigate the pain resulting from imbalanced data.

B. Takeaways

We implemented and experimented a variety of machine learning models ranging from traditional machine learning model (SVM), shadow neural networks (self-designed CNN), to deep neural networks (VGG, DenseNet etc.) in this project. One important takeaway is that although these models differ fundamentally in terms of algorithm, complexity and how we train them, their performance on detecting Pneumonia cases are unexpectedly good - even SVM as the baseline can achieve an testing accuracy of 73.4%. By applying simple Convolutional Neural Networks, we are able to improve the accuracy to over 90%, with promising precision, recall and f1 scores as well. Those deeper CNN models, after fine-tuned based on ImageNet pre-training, are performing overwhelmingly good results. These results illustrates the promising potential of applying machine learning, particularly deep learning models in assisting medical image processing.

Another important takeaway is that imbalanced data has a significant effect on the performance. Without data augmentation and weighted loss function, the model tends to predict all samples as the majority class. Data augmentation is necessary to reduce the bias to the majority class. In addition, we found out that weighted loss function is not only useful in solving the problem of imbalanced data, but also useful to make the model focus on a particular class. In our case, we want make sure our model will not misclassify any samples from the class PNEUMONIA, then we used weighted loss function to penalize more when the model make wrong predictions on the class PNEUMONIA.

Our experiments also demonstrated the notable benefits of applying Tranfer Learning in the context of medical image processing. By fine tuning D-CNN models pretrained on large scale dataset of general purpose, we can significantly improve the neural network performance comparing to those CNN models with fewer layers without suffering from issues such as the lack of training data and demanding computational power.

VI. Conclusion

In this paper, we applied 3 machine learning methods to the same dataset to classify pneumonia from chest x-ray images. SVM as our baseline does not have an ideal result. We discovered a radical drop in the precision and F1-score of the testing dataset compared with the training results. For the proposed CNN model, with the help of data augmentation and weighted loss function, we can mitigate the problem of imbalanced data. This is proved by the precision tested on the test data, which is close to the recall on the test data. CNN is especially useful in classifying image data. Therefore, the performance of CNN is much better than the performance of SVM. We think the CNN model can be improved by adding more layers while using other regularization such as L1 or L2 regularizations and addressing the imbalanced data further. If possible, collecting more data of chest X-ray images from healthy people is always the best solution to an imbalanced data. We also applied transfer learning on the dataset. It turns out that as the complexity of the fine-tuned layers increases, the performance increases. And the accuracy of all fine-tuned models is better than our proposed model. We think one of the main reasons is due to the complexity of the neural network. All of the pretrained models are state-of-the-art models, which means that the architectures of those models are sophisticated in design. But none of the pre-trained models was trained on medical images. We think it is still possible to increase the performance of fine-tuned neural networks by using a pre-trained model which was trained on medical images. As we know, transfer learning is especially useful when the dataset used to train the model is related to our dataset. In conclusion, deep learning models including CNN and transfer learning from pre-trained models are helpful in medical image processing.

References

[1] Million, M., Lagier, J. C., Gautret, P., Colson, P., Fournier, P. E., Amrane, S., ... Raoult, D. (2020). Early treatment of COVID-19 patients with hydroxychloroquine and azithromycin: A retrospective analysis of 1061 cases in Marseille, France. Travel medicine and infectious disease, 35, 101738.

[2] Li, Y., Shen, L. (2018). Skin lesion analysis towards melanoma detection using deep learning network. Sensors, 18(2), 556.

[3] Hemdan, E. E. D., Shouman, M. A., Karar, M. E. (2020). Covidx-net: A framework of deep learning classifiers to diagnose covid-19 in x-ray images. arXiv preprint arXiv:2003.11055.

[4] Wang, D., Khosla, A., Gargeya, R., Irshad, H., Beck, A. H. (2016). Deep learning for identifying metastatic breast cancer. arXiv preprint arXiv:1606.05718.

[5] Kermany, D. S., Goldbaum, M., Cai, W., Valentim, C. C., Liang, H., Baxter, S. L., ... Zhang, K. (2018). Identifying medical diagnoses and treatable diseases by image-based deep learning. Cell, 172(5), 1122-1131.

[6] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).

[7] Mooney, P. (2018, March 24). Chest X-ray images (pneumonia). Kaggle. Retrieved December 9, 2021, from https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia.

[8] Stephen, O., Sain, M., Maduh, U. J., Jeong, D. U. (2019). An efficient deep learning approach to pneumonia classification in healthcare. Journal of healthcare engineering, 2019.

[9] Prasad, S. (2021, March 17). What is image segmentation or segmentation in image processing? Blogs amp; Updates on Data Science, Business Analytics, AI Machine Learning. Retrieved September 27, 2021, from https://www.analytixlabs.co.in/blog/what-is-image-segmentationmethod5.

[10] Raghu, M., Zhang, C., Kleinberg, J., Bengio, S. (2019). Transfusion: Understanding transfer learning for medical imaging. arXiv preprint arXiv:1902.07208. https://proceedings.neurips.cc/paper/2019/file/eb1e78328c46506b46a4ac4a1 Paper.pdf

[11] Kermany, D., Zhang, K., Goldbaum, M. (2018). Labeled optical coherence tomography (oct) and chest X-ray images for classification. Mendeley data, 2(2).

[12] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). Ieee.

[13] Simonyan, K., Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[14] He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[15] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K. Q. (2017). Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4700-4708).

[16] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).

Appendix A
Confusion Matrices of Fine-tuned D-CNN

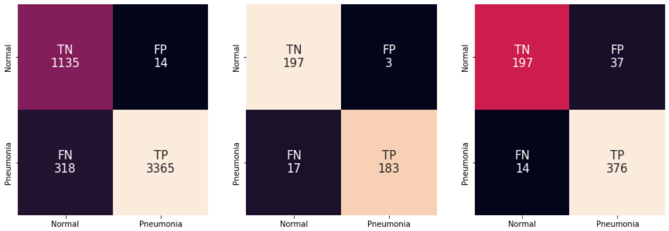


Fig. 9: The Confusion Matrices of Fine-tuned Inception Net

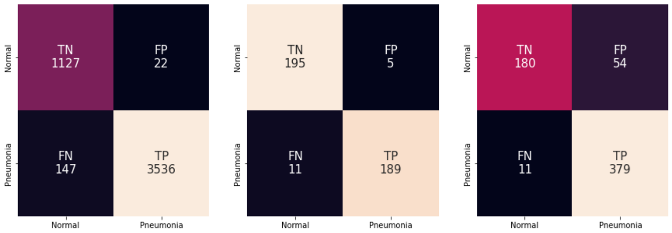


Fig. 10: The Confusion Matrices of Fine-tuned ResNet

Appendix B
How we tuned CNN

First, we checked the dataset and realized that the dataset is imbalanced. In order to solve the problem of an imbalanced dataset, we decided to use data augmentation. On one hand, data augmentation will relieve the problem of imbalanced data, on the other hand, its ability to reduce overfitting provides us an opportunity to use a complex model. We used regular data augmentation for the train set including random rotation, random zoom and random shift horizontally and vertically.

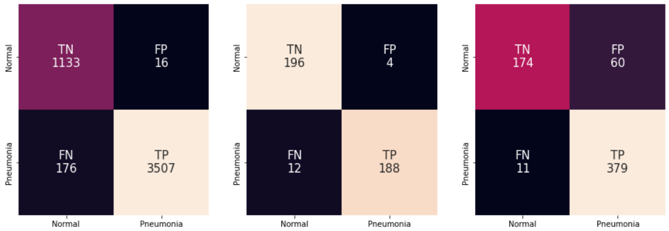


Fig. 11: The Confusion Matrices of Fine-tuned DenseNet

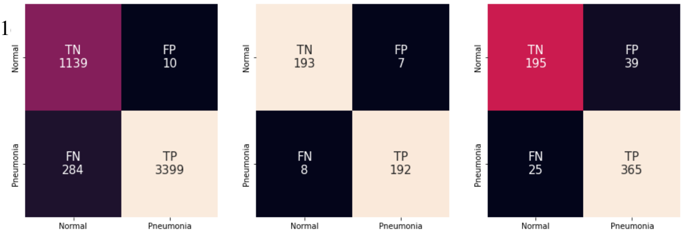


Fig. 12: The Confusion Matrices of Fine-tuned VGG Net

Second, by checking others' open sourced model, we learn that it is possible to have a good model with only three convolutional layers. In order to achieve a better result, we decided to use a CNN model with 5 convolutional layers. At the beginning, the number of kernels for each layer is 32, 64, 128, 256, 512. However, we found out that the model converged too fast, and the accuracy of the validation set fluctuated dramatically, which was apparently overfitting. So, we decided to reduce the number of parameters, then the number of kernels for each layer is 32, 64, 128, 128, 256. We decided to use Adam optimizer since it is one of the best optimizers in CNN models. Then we added batch normalization to stabilize the training process by centering and scaling mini-batches and dropout layers to prevent overfitting. In terms of the ordering of different layers, we consider the purpose of each kind of layer. Batch normalization will recenter and rescale mini batches and dropout will disable some neurons by setting zeros to weights. If we put dropout before batch normalization, then 'zero' weights will be no longer zeros, which will add noise and destroy the purpose of dropout layers. Max-pooling tends to take the maximum value within a kernel. If we put max-pooling layers after dropout, then randomly setting zeros to weights may affect the maximum values in the kernel. In addition, max-pooling layers after dropout will reduce the effectiveness of the dropout layer since by selecting the maximum value it will ignore the effect of the dropout layer. The ordering between batch normalization and ReLU activation functions does not affect the performance. After considering all these situations, we chose the ordering of layers as: Conv -> ReLU -> Batch Normalization -> Max-pooling -> Dropout.

Third, In order to achieve the best result, we decided to use a reduced learning rate to help the model converge and achieve the best model. By using the default learning rate of the Adam optimizer (0.001), the model converged too fast and only trained for a few epochs. We decreased the learning rate to 0.0001 to slow down the training process to have a better visualization of the process. In order to reduce the probability of overfitting, we applied an early stopping technique. However, we found out that when the early stopping monitored the accuracy of the validation set, it did not guarantee to have a model with the best result. It turned out that the reason was because the validation set is balanced but the train set was not. Therefore, the validation set could not reflect the imbalanced data, and it was impossible to use the accuracy of the validation set as an indicator to stop the training. So we only used early stopping on the rearranged dataset.

Then, for the purposes of solving the problem of imbalanced data and emphasizing the recall metric, we employed weighted binary cross-entropy. By manipulating the class weights in the loss function, we can make the model focus on the class we want. The class weights for solving imbalanced data are calculated by the following formula: $n_samples/(n_classes * n_samplesinclassi)$, where $n_samples$ is the total number of samples in the dataset, $n_classes$ is the number of classes, and $n_samplesinclassi$ is the number of samples belonging to $classi$. After calculating the class weights, we added more weights to class PNEUMONIA to penalize the misclassification of class PNEUMONIA. Therefore, we can achieve a ‘balanced’ train set while emphasizing the importance of recall.