

CIS 545: Big Data Analytics

Project Name:

Rong, Fan; Chenrui, Hu; Yue, Xing

April 30, 2022

I. Introduction

The chest X-ray is one of the most commonly accessible and usable radiological examination for screening and diagnosis of many lungs' disease¹. The NIH Clinical Center had released over 100,000 anonymized chest x-ray images and their corresponding classification label to the scientific community.

Based on this dataset, our project is mainly focus on combining machine learning with data analysis tools to realize Chest X-ray image classification and meanwhile, to perform some statistical data analysis from the result of model training. We also want to obtain some deeper, non-intuitive data features and potential connections between labels. Since this is a multi-class and multi-label task, each image may belong to one or more classes of the 14 pathologies.

II. Pre-trained Exploratory Data Analysis (EDA)

Since we are about to achieve Chest X-ray image classification, we only care about image and their labels. The first step of exploratory data analysis is to drop certain columns and row, which includes useless information and empty diagnosis labels.

We found that labels in this dataset is represented with list of words divided by '|'. So, for the further step of model training, we also need to encode labels and transform them to numeric expression. Because each label is equal and there is no priority of labels, we choose to use One-Hot Encoding to make it available for machine learning. Fig 1 shows the encoding result.

	Atelectasis	Cardiomegaly	Consolidation	Edema	Effusion	Emphysema	Fibrosis	Hernia	Infiltration	Mass	Nodule	Pleural_Thickening	Pneumonia	Pneumothorax
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	1	0	0	0	0	0	0	0	0
2	0	1	0	0	1	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	1	0	0	0	0	0	0
4	0	0	0	0	0	0	0	1	0	0	0	0	0	0
...
51754	0	0	1	0	0	0	0	0	0	0	0	0	0	0
51755	0	0	0	0	0	0	0	0	1	0	0	0	0	0
51756	0	0	0	0	0	0	0	0	0	1	1	0	0	0
51757	0	0	0	0	0	0	0	0	0	0	0	1	0	0
51758	0	0	0	0	0	0	0	0	0	1	0	0	1	0

51759 rows x 14 columns

Fig 1. Labels and One-Hot Encoding result

Then we use image indexes to get their storage location and aggregate all information to get the final dataframe for machine learning. The final dataframe only contain image Index, Patient ID, all the

encoded labels and the file paths of images, which is shown in Fig 2. This figure is just an example and only use one sub dataset, so some paths in column “FilePath” are NaN.

Index	Patient ID	Atelectasis	Cardiomegaly	Consolidation	Edema	Effusion	Emphysema	Fibrosis	Hernia	Infiltration	Mass	Nodule	Pleural_Thickening	Pneumonia	Pneumothorax	FilePath
0	00000001_000.png	1	0	1	0	0	0	0	0	0	0	0	0	0	0	/content/images_001/images/00000001_000.png
1	00000001_001.png	1	0	1	0	0	0	1	0	0	0	0	0	0	0	/content/images_001/images/00000001_001.png
2	00000001_002.png	1	0	1	0	0	1	0	0	0	0	0	0	0	0	/content/images_001/images/00000001_002.png
3	00000003_001.png	3	0	0	0	0	0	0	0	1	0	0	0	0	0	/content/images_001/images/00000003_001.png
4	00000003_002.png	3	0	0	0	0	0	0	0	1	0	0	0	0	0	NaN
...
51754	00030786_006.png	30786	0	0	1	0	0	0	0	0	0	0	0	0	0	NaN
51755	00030789_000.png	30789	0	0	0	0	0	0	0	0	1	0	0	0	0	NaN
51756	00030793_000.png	30793	0	0	0	0	0	0	0	0	0	1	1	0	0	NaN
51757	00030795_000.png	30795	0	0	0	0	0	0	0	0	0	0	0	1	0	NaN
51758	00030801_001.png	30801	0	0	0	0	0	0	0	0	0	1	0	0	1	NaN

51759 rows x 17 columns

Fig 2. Final Dataframe

After we got the final dataframe, we want to see if there is some obvious correlation in the raw unprocessed data. We drew a heatmap to check correlation (Fig 3) and did not find any obvious conclusion.

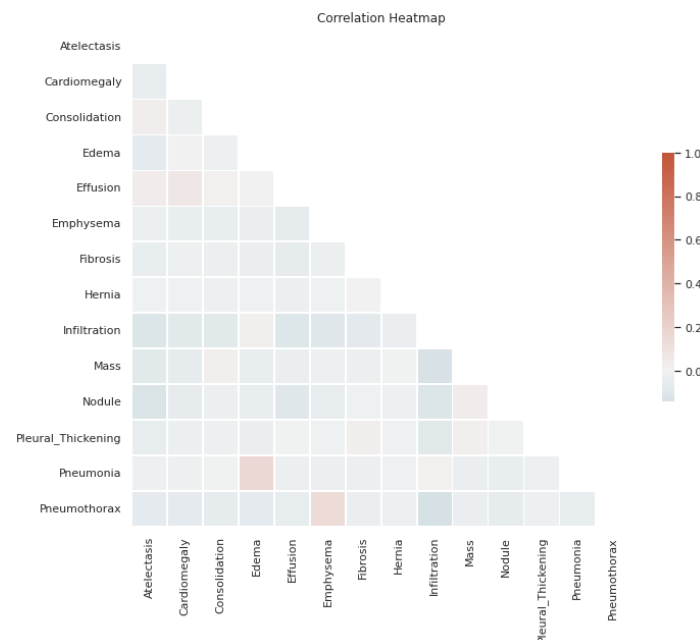


Fig 3. Correlation of unprocessed data (Heatmap)

III. Modeling

Our project is mainly focus on realizing image classification, so we firstly consider to build a CNN model (Convolutional Neural Networks) to achieve the classification. We chose 80% of data as training dataset and 20% as test dataset and used ImageDataGenerator to get the images with their labels and do data augmentation to the images. The results are visualized as shown below.



Fig 4. ImageDataGenerator result

We used Keras to build CNN model. This model that has 4 convolutional layers and 2 fully connected layers with dropout. Input size of image is 512/512, output feature map is 14 probabilities. Kernel size is 3*3 and activation function is Relu. The batch size we used for CNN is 64 because we successfully run it on AWS. The learning curve by training on the first sub dataset is shown in Figure 5. We can see finally the loss converges to 0.3 and accuracy converges to 0.9.

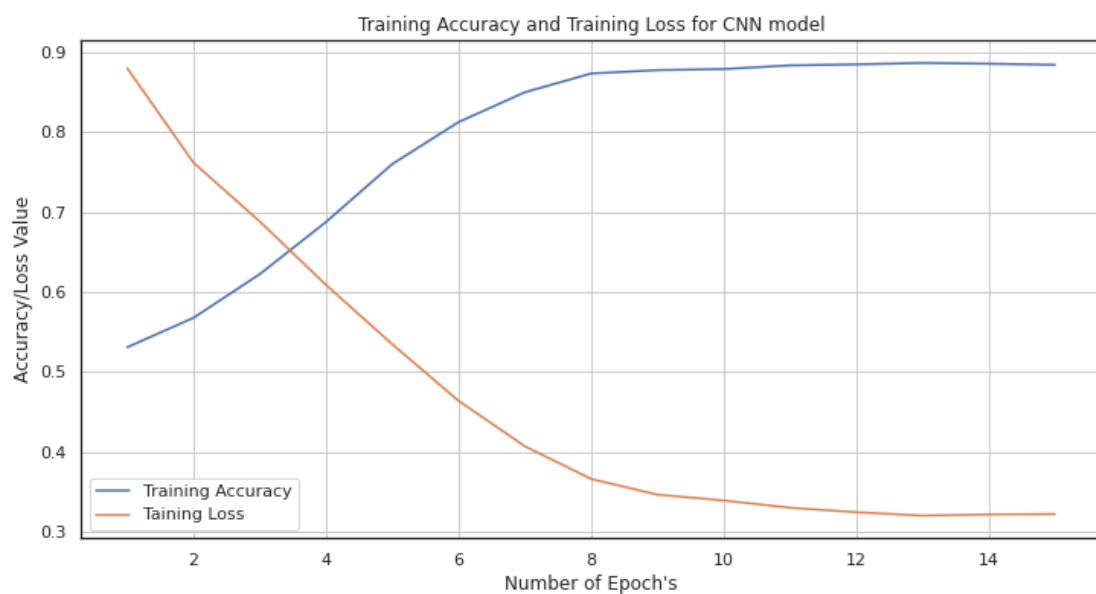


Fig 5. Learning curve for cnn

After predicting using test set, we calculated the confusion matrix of our CNN model (Fig 5).

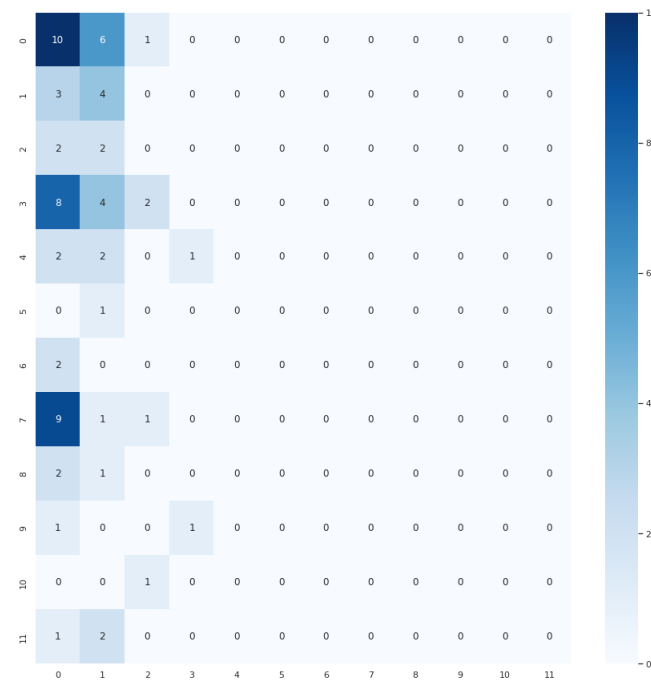


Fig 5. Confusion Matrix of CNN

Then we save the model and load it before training on other sub datasets. The result is that there is no significant decrease of loss because we already have a nice model for the first train. But we still do this trying to see if it is possible to improve our model.

We also use another network, DenseNet121, combining transfer learning with deep learning. Although the batch size this time is only 8, we can see the model almost learned everything just in the second epoch. Thanks to DenseNet, the model converges faster. Besides, we also use other sub datasets to train the DenseNet model. The final loss and accuracy still converge to 0.3 and 0.9 respectively.

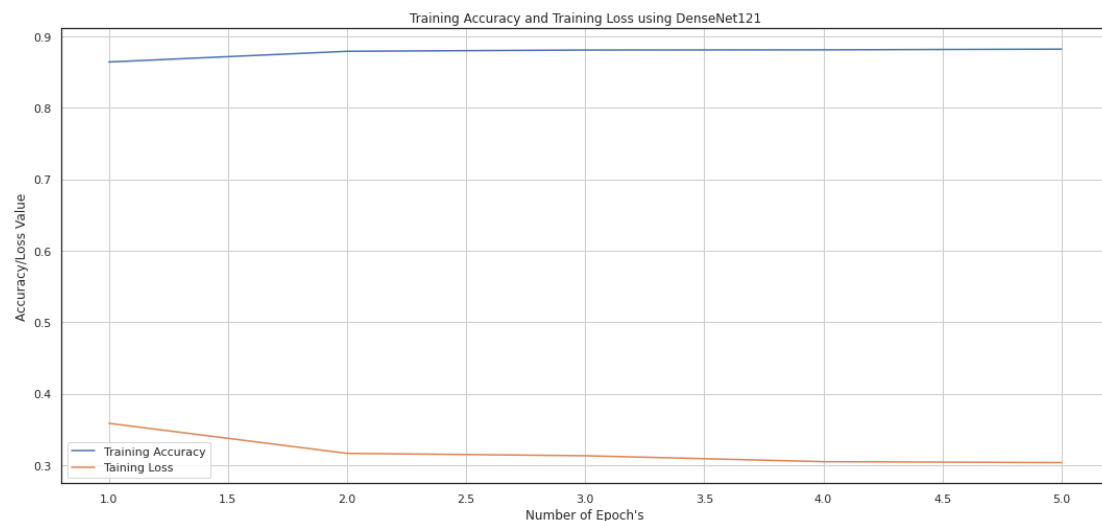
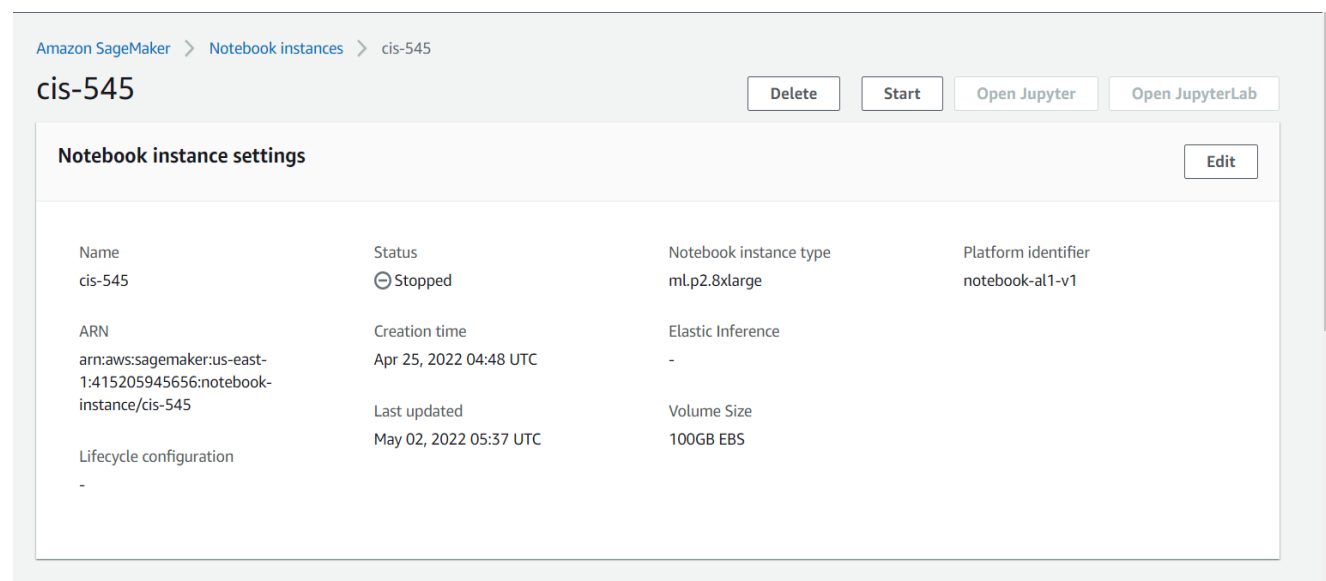


Fig 6. Learning curve for DenseNet121

IV. AWS sageMaker usage for training CNN and DenseNet model.

The whole dataset is too large for storage of colab and usage of GPU is limited. In order to train all dataset images, we choose to use AWS platform and establish a sageMaker notebook for our project with the type of ml.p2.8xlarge (100GB for SSD and 128 RAM), also, we stored our trained model as h5 file.

AS mentioned above, because of help AWS we choose **64 as batch size** and **successfully run** DenseNet with 8 batch size.(failure on colab because of limited RAM usage.)



The screenshot displays the Amazon SageMaker console interface for a specific notebook instance. At the top, the breadcrumb navigation shows 'Amazon SageMaker > Notebook instances > cis-545'. The instance name 'cis-545' is prominently displayed on the left, with action buttons 'Delete', 'Start', 'Open Jupyter', and 'Open JupyterLab' on the right. Below this, the 'Notebook instance settings' section is visible, featuring an 'Edit' button. The settings are organized into a table with four columns: Name, Status, Notebook instance type, and Platform identifier. The instance 'cis-545' is currently 'Stopped'. Additional details like ARN, Creation time (Apr 25, 2022 04:48 UTC), Last updated (May 02, 2022 05:37 UTC), Elastic Inference status, and Volume Size (100GB EBS) are also provided.

Name	Status	Notebook instance type	Platform identifier
cis-545	⏻ Stopped	ml.p2.8xlarge	notebook-al1-v1

ARN	Creation time	Elastic Inference
arn:aws:sagemaker:us-east-1:415205945656:notebook-instance/cis-545	Apr 25, 2022 04:48 UTC	-

Lifecycle configuration	Last updated	Volume Size
-	May 02, 2022 05:37 UTC	100GB EBS

V. Post-trained EDA

In this part, we will show how we use the model trained result to do data visualization and statistical data analysis. First, we care about the total number of patients and healthy people in this dataset, because the distribution of dataset is directly related to the training result of our model.

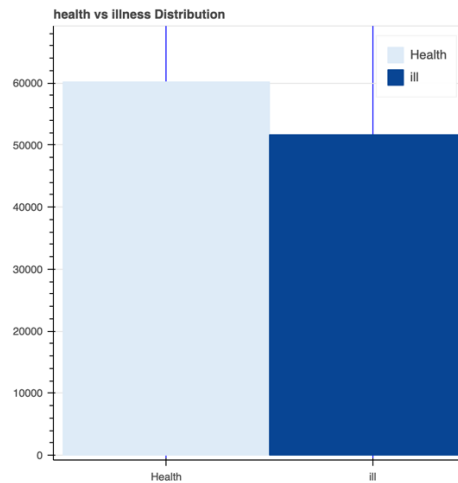


Fig. Heathy people and patients

We want check whether the patient's diagnostic label is linked to the patient's ID. We drew the diagnosis distribution on patients' ID, which is shown below.

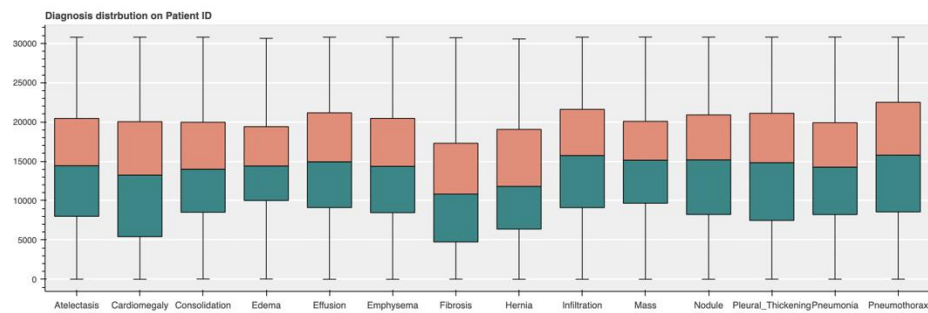


Fig. Diagnosis distribution of patients' ID

We also want to see the number of times each labels appears. In order to make the result more intuitive, we will draw it into a histogram and a pie chart and result are as follow.

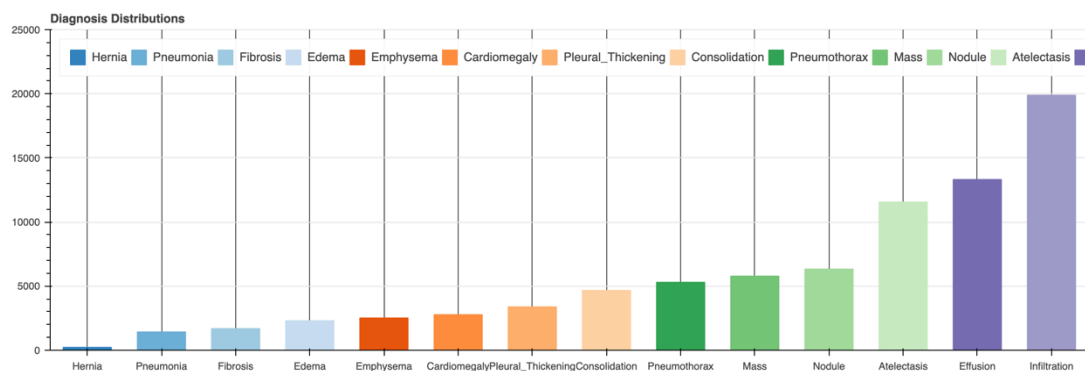


Fig. Diagnosis Distribution (Histogram)

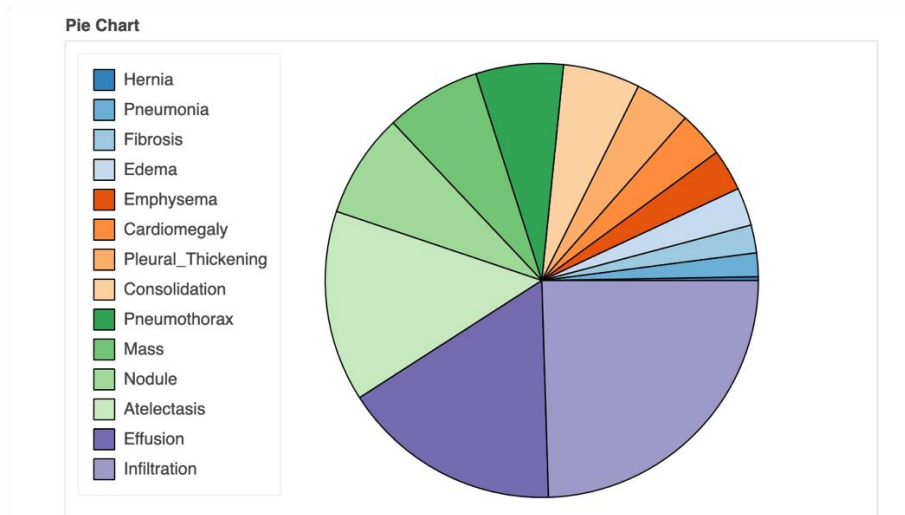
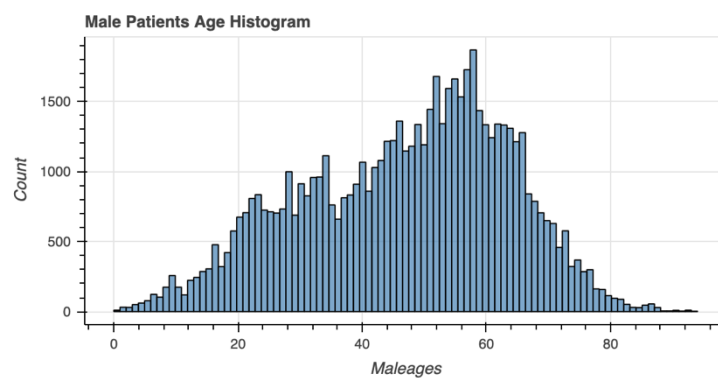
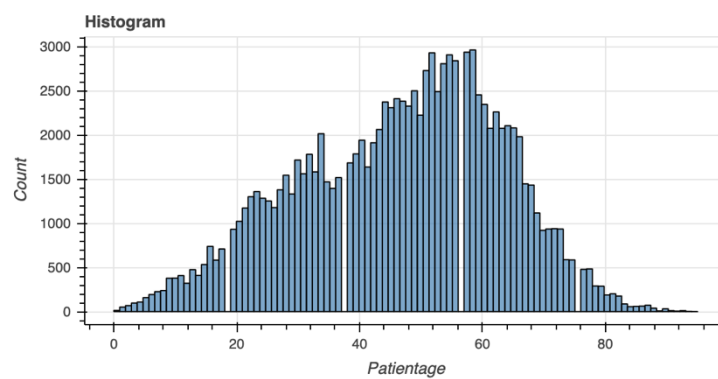
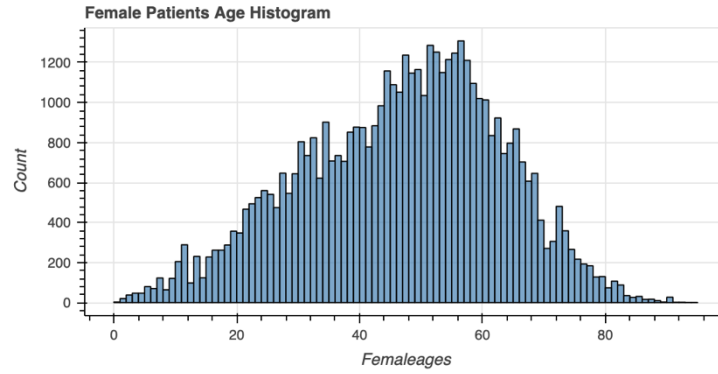


Fig. Diagnosis Distribution (Pie chart)

In the previous Pre-trained EDA, we deleted the column of patient age in the original data. In order to count the relationship between the number of samples and patient age and patient gender, we drew the corresponding histogram, as shown in the following figures.





We want to use pairwise bivariate distributions to analyze the relation between each and every variable in this dataset, which is shown below.

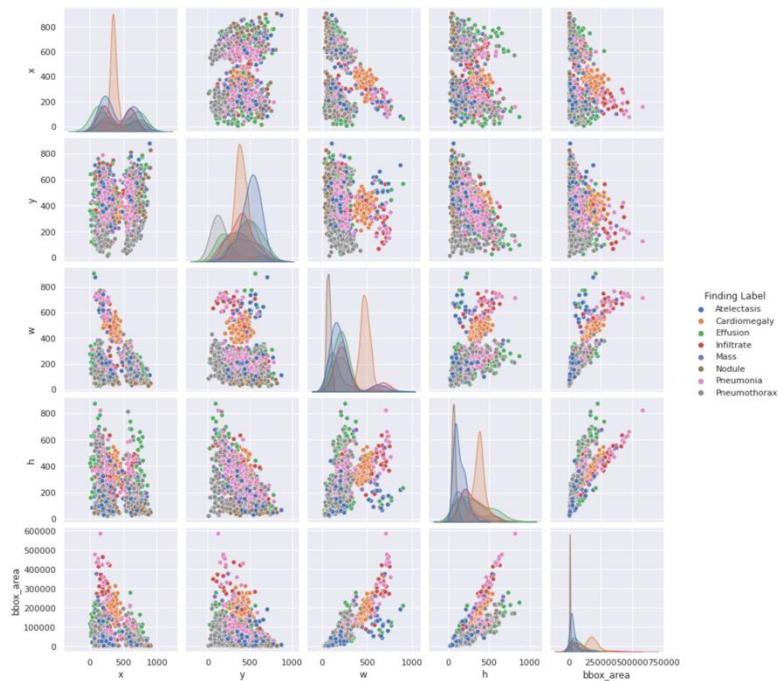


Fig. Visualizing Pairwise Relationship

VI. Key question response

- Is your model useful?

There is no doubt for using CNN (convolutional neural network) and DenseNet for doing image classification as this is basic models for processing images. Finally, after trained model, we get high accuracy on training accuracy and loss.

- Is your model implemented correctly?

Yes, for CNN, we correctly add convolutional layer, max-pooling layer, batch normalization and dropout layers as one stage of CNN and we build 4 stages and output state using ReLU and sigmoid to flatten and do classification.

- The models that you choose should be justified.

For capturing every feature inside of model, we choose DenseNet to train model because In DenseNet, each layer obtains additional inputs from all preceding layers and passes on its own feature-maps to all subsequent layers. Concatenation is used. Each layer is receiving a “collective knowledge” from all preceding layers.

VII. Challenges and Obstacles

- Have you attempted challenging analysis?

Yes, deep learning with machine learning on CV (computer vision) is the hardest part of our courses, we try our best to practice on the knowledge have learned from course.

- How much time would have been required to complete your project?

We use nearly about two months to do this project and even that we cannot cover all materials from our lectures, there is still a large potential for our projects.

VIII. Potential Next Steps Future Direction

- **K-fold validation implement.**

Obviously, our model has high possibility of being overfitting, to prevent we need add k-fold validation to keep balance (no high bias underfitting and no high variance with overfitting)

- **Over-sampling our dataset.**

From the heatmap of confusion matrix, clearly can be seen the diagnosis called Herina consist even the 30% of the whole dataset. To keep the sample numbers as large as possible, we do over-sampling instead of under-sampling.

- **Transfer Learning**

one of interest criteria in deep learning, as we can use pretrain model of Vgg16 or MobileNet...etc, change the output layer to do classification on 14 different diagnosis.

ⁱ Wang X, Peng Y, Lu L, Lu Z, Bagheri M, Summers RM. ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. IEEE CVPR 2017,
http://openaccess.thecvf.com/content_cvpr_2017/papers/Wang_ChestX-ray8_Hospital-Scale_Chest_CVPR_2017_paper.pdf