Machine Learning Engineer Nanodegree

Capstone Project

Duy Do Van
January 25st, 2018

Diseases Detection from Chest X-ray data

1. Definition

a) Project Overview

The world is changing so fast that the pressure on health is increasing, the bad changes of climate, the environment, the life way of human, ... also increase the risk as well as diseases for people. One of the issues that we will focus on in this article is lung diseases.

About 3.2 million people succumbed in 2015 to chronic obstructive pulmonary disease (COPD), caused mainly by smoking and pollution, while 400,000 people died from asthma [1].

With so many lung diseases people can get, here is just one example of diseases we can save if we find them out earlier.

With the technology machine and computer power, the earlier identification of diseases, particularly lung disease, we can be helped to detect earlier and more accurately, which can save many many people as well as reduce the pressure on the system. The health system has not developed in time with the development of the population.

I would also like to thank the scientists who have gone ahead in their research to offer to humanity, applying machine learning to the problem of X-ray image prediction [2345678].

With the power of computers as well as the large amount of data being released to the public, this is a good time to contribute to solving this problem. Wishing to contribute more to the community, helping those who are not able to pay for medical expenses, I hope that my solution can contribute to reducing medical costs, the development of computer science for medical projects.

I am very fortunate to know that there is a huge set of X-ray image data on Kaggle.

Sample dataset [11]**:**

- File contents: this is a random sample (5%) of the full dataset:

    • sample.zip: Contains 5,606 images with size 1024 x 1024

    • sample_labels.csv: Class labels and patient data for the entire dataset

- Class descriptions: there are 15 classes (14 diseases, and one for "No findings") in the full dataset, but since this is drastically reduced version of the full dataset, some of the classes are sparse with the labeled as "No findings": Hernia - 13 images, Pneumonia - 62 images, Fibrosis - 84 images, Edema - 118 images, Emphysema - 127 images, Cardiomegaly - 141 images, Pleural_Thickening - 176 images, Consolidation - 226 images, Pneumothorax - 271 images, Mass - 284 images, Nodule - 313 images, Atelectasis - 508 images, Effusion - 644 images, Infiltration - 967 images, No Finding - 3044 images.

Full dataset [12]**:**

- File contents:

    • images_00x.zip: 12 files with 112,120 total images with size 1024 x 1024

    • README_ChestXray.pdf: Original README file

    • BBox_list_2017.csv: Bounding box coordinates. *Note: Start at x,y, extend horizontally w pixels, and vertically h pixels*

    • Data_entry_2017.csv: Class labels and patient data for the entire dataset

- Class descriptions: there are 15 classes (14 diseases, and one for "No findings"). Images can be classified as "No findings" or one or more disease classes: Atelectasis, Consolidation, Infiltration, Pneumothorax, Edema, Emphysema, Fibrosis, Effusion, Pneumonia, Pleural_thickening, Cardiomegaly, Nodule Mass, Hernia.

From the knowledge and current technology conditions, I think I can contribute some parts to the community in analyzing and building a model based on this useful data set.

The data set contains useful information for the model we will build as: age, gender, patient data, snapshot data as well as X-ray images. From this main information I will use it to model the model.

In diagnosis from X-ray data, the physician can diagnose a part of the patient's medical condition, so I think that with the X-ray chest image data, the intelligent machine can support the physician in the diagnosis of the disease, some data on age and gender will also be considered to increase the accuracy of this system.

b) Problem Statement

Recently a large dataset of X-ray lung data was public on Kaggle followed by labeled lung disease data. This is a good condition for me to implement this project.

In this project I will conduct a study and analysis of this data set, then apply Machine Learning and Deep Learning to predict that the patient has a lung disease. This project is a binary classification with input is patient's data (age, gender, X-ray images, View Position) and output is found diseases or not.

The difficulty is a new dataset, and I will be one of the pioneers to learn it, my analysis is that this is a large dataset but has never been processed full, data has a lot of noise, and X-ray of the lung is not likely to provide enough information to assess whether a patient may be ill.

I will use Machine Learning as well as Deep Learning to process data as well as create models for diagnosing patients. My keys point here will be: combining the processing of patient information with data from X-rays, using CNN with the well-known pre-trained model, first time using the CapsNet [910] network for data this form.

c) Metrics

The evaluation metrics used here will be precision, recall and F-beta scores (beta is 0.5) for binary classification – found diseases or not. In this case F score is better than accuracy because with binary classification found diseases or not, the classes are imbalanced. For example, consider you have a trivial classifier that just guesses the majority class, it will obtain 80% accuracy when there is an 80/20 split and 50% accuracy when there is a 50/50 split.

These indexes will be evaluated on a separate testing data set from the original dataset.

These indicators will be evaluated for all diseases – found disease or not.

If the case is positive and not negative, then the indicators:

tp: True Positive - The number of people affected is expected to be affected

fp: False Positive - The number of people who are sick is predicted to be unwell

fn: False Negative - The number of people without the disease is predicted to be wrong

$$\text{Precision} = \frac{tp}{tp + fp} \qquad \text{Recall} = \frac{tp}{tp + fn}$$

Precision and Recall will focus on the number of people expected to be affected, thereby overcoming the skew data status and the importance of predicting a person's illness.

Precision represents the proportion of people who correctly predicted the disease in the total number of people who were predicted to be sick. Recall represents the proportion of people who correctly predicted illness on the total number of people actually infected. Both of these indicators are very important in predicting the disease, and we need an index that can be both Precision and Recall.

From here we have a combination of Precision and Recall F score:

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{precision} \cdot \text{recall}}{\beta^2 \cdot \text{precision} + \text{recall}}$$

With different β will show the importance between different and precision different. There are two main ideas for choosing the importance of precision and recall:

- Models need to be sure that a patient is expecting a disease, which means that each expectation is highly confident because when a patient is diagnosed with a disease, it is very shocking. It is very important, as this is a method to assist doctors with other diagnostic procedures. So we want high precision and low recall, corresponding to small β, taking $\beta = 0.5$ for F beta score.
- Models need to avoid mispronouncing sick people to avoid illness, avoiding missing patients at risk. This case will select low precision and high recall, corresponding to large β, taking $\beta = 2$ for F beta score.

For the purpose of this project is to prove the technology as well as support the doctors in diagnosing the disease because to determine the disease will need many tests on the patient. The patient will be anxious before knowing further results, so I choose F-0.5-score is $\beta = 0.5$.

With $\beta = 0.5$ to represent precision will be more important than recall in this case.

2. Analysis

a) Data Exploration

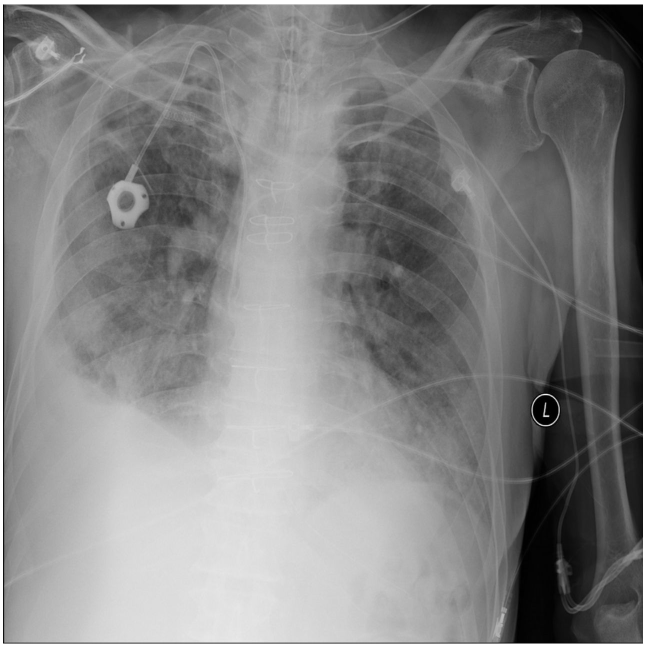
NIH Chest X-ray Dataset (National Institutes of Health Chest X-Ray Dataset)

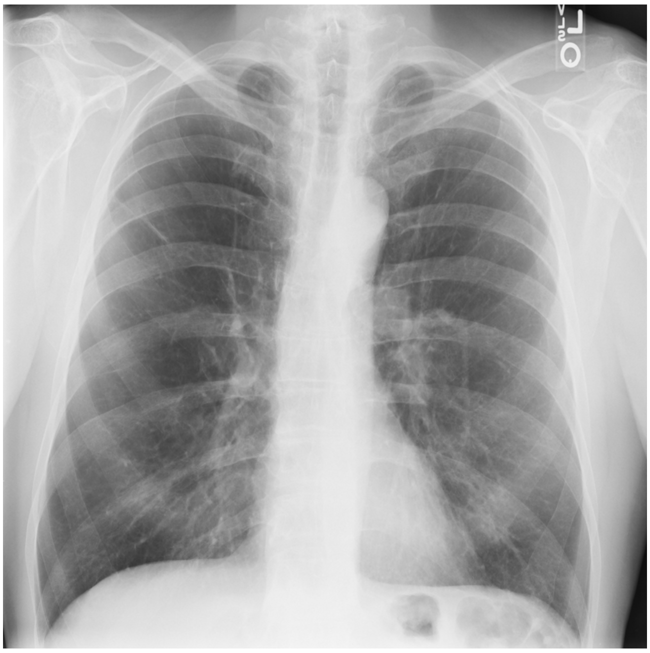
Chest X-ray exams are one of the most frequent and cost-effective medical imaging examinations available. However, clinical diagnosis of a chest X-ray can be challenging and sometimes more difficult than diagnosis via chest CT imaging. The lack of large publicly available datasets with annotations means it is still very difficult, if not impossible, to achieve clinically relevant computer-aided detection and diagnosis (CAD) in real world medical sites with chest X-rays. One major hurdle in creating large X-ray image datasets is the lack resources for labeling so many images. Prior to the release of this dataset, Openi was the largest publicly available source of chest X-ray images with 4,143 images available.

This NIH Chest X-ray Dataset is comprised of 112,120 X-ray images with disease labels from 30,805 unique patients. To create these labels, the authors used Natural Language Processing to text-mine disease classifications from the associated radiological reports. The labels are expected to be >90% accurate and suitable for weakly-supervised learning. The original radiology reports are not publicly available but you can find more details on the labeling process in this Open Access paper: "ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases." (Wang et al.)

<u>In this sample dataset (about 5% of the full dataset)</u>:

✗ Contains 5,606 images with size 1024 x 1024





✗ Class labels and patient data for the entire dataset:

- Image Index: File name
- Finding Labels: Disease type (Class label)
- Follow-up #
- Patient ID
- Patient Age
- Patient Gender
- View Position: X-ray orientation
- OriginalImageWidth
- OriginalImageHeight
- OriginalImagePixelSpacing_x

- OriginalImagePixelSpacing_y

The data set contains useful information for the model we will build as: age, gender, view positiont, snapshot data as well as X-ray images. From this main information I will use it to model the model.

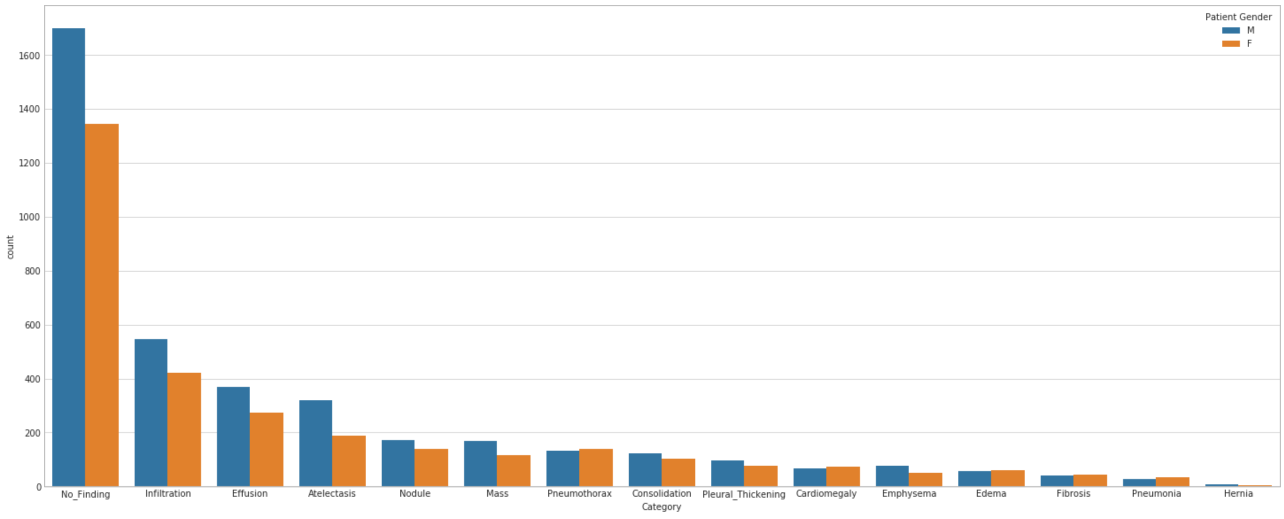
b) Exploratory Visualization

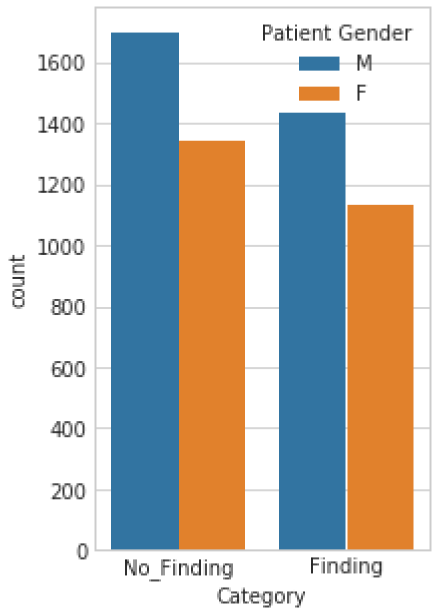
Notebook: “Data analysis – FullDataset”, “Data analysis – SampleDataset”.

Here are a few plots that will show us more about the disease data we will use.

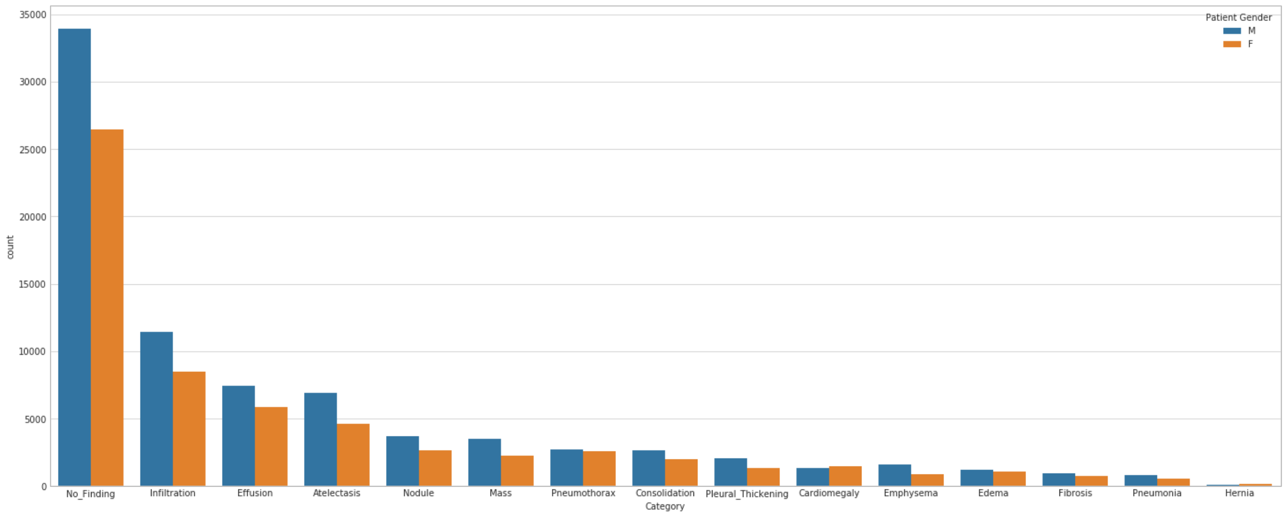
Diagram of the number of patients by disease and sex:

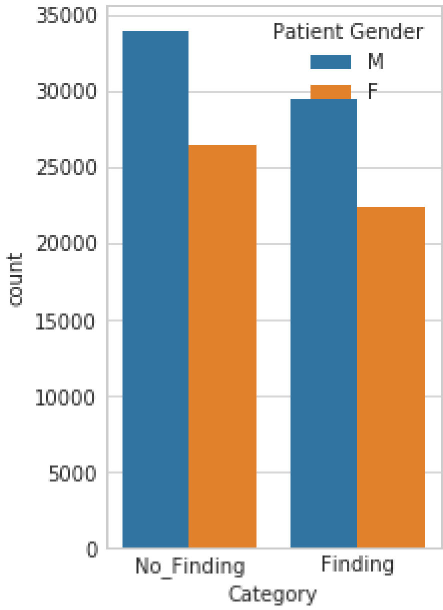
Sample dataset:





Full dataset:



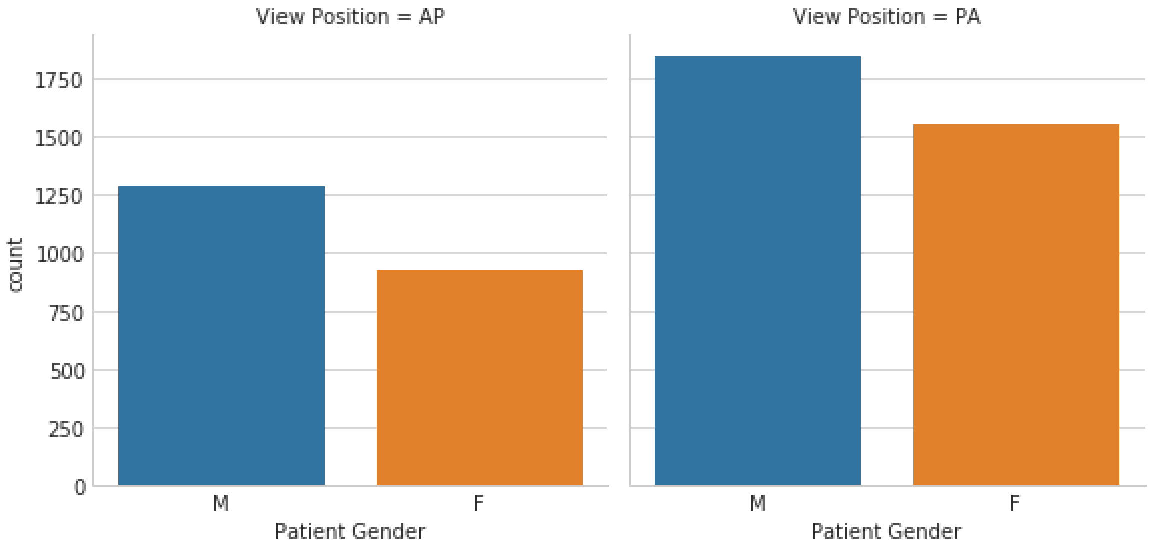


The pictures show some diseases with very few cases such as Hernia, Pneumonia, Fibrosis and some more common lung diseases such as Infiltration, Effusion, Atelectasis. Disease distribution is indeed uneven.

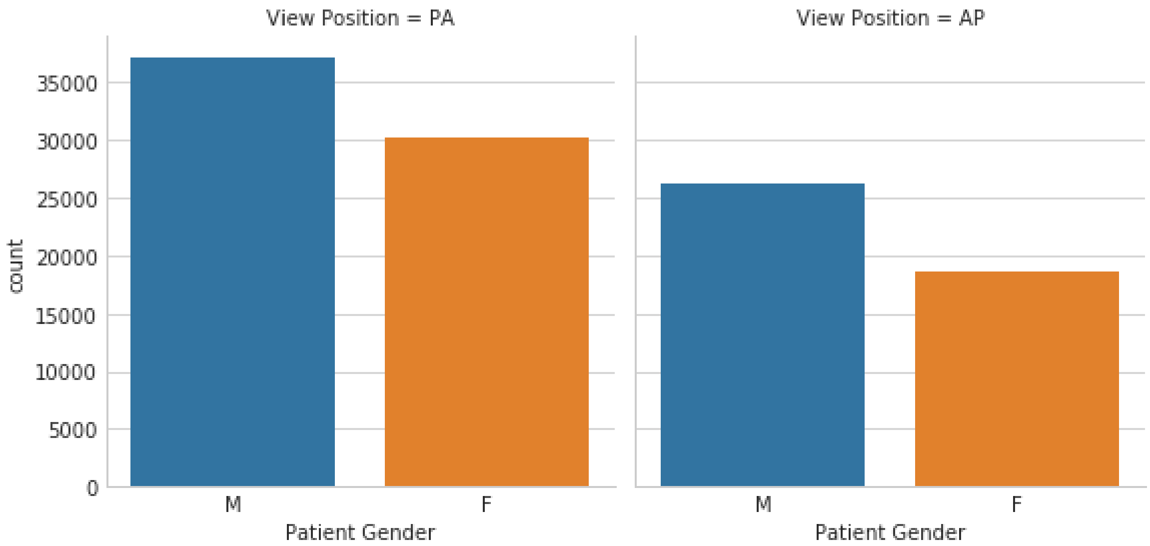
In this list, the total number of men is higher than that of women, the number of confirmed cases is higher than the number of men diagnosed with lung disease. In cases where the disease is not confirmed, the difference shall not be so great.

Diagram of the distribution of patients by sex and view position:

Sample dataset:



Full dataset:



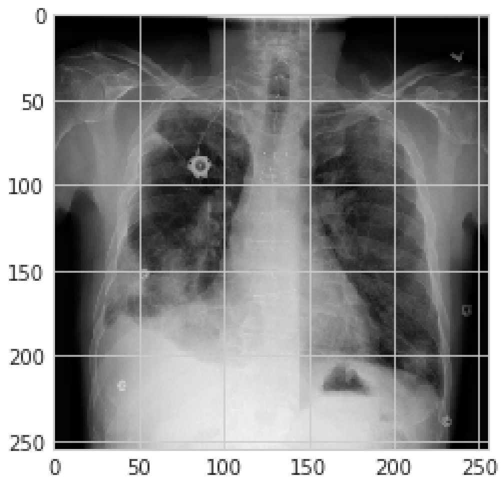
View Position:

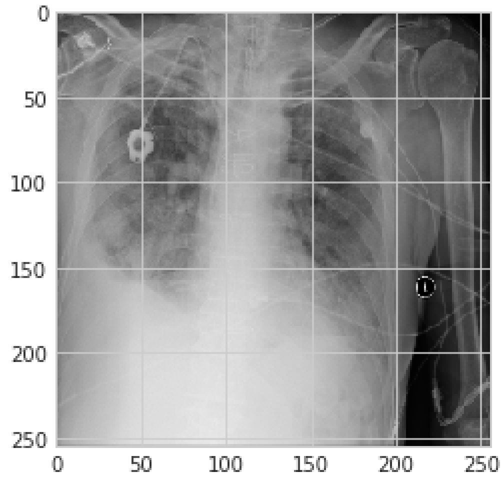
Posterior-anterior (PA) Position: The standard position for obtaining a routine adult chest radiograph. Patient stands upright with the anterior wall of chest placed against the front of the film. The shoulders are rotated forward enough to touch the film, ensuring that the scapulae do not obscure a portion of the lung fields. Usually taken with the patient in full inspiration. The PA film if viewed as if the patient is standing in front of you.

Anterior-posterior (AP) Position: Used when the patient is debilitated, immobilized, or unable to cooperate with the PA procedure. Film is placed behind the patient's back with the patient in a supine position. Heart is at a greater distance from the film hence appear more magnified than in a PA. The scapulae are usually visible in the lung fields because they are not rotated out of the view as they are in a PA.

It can be seen that these two types of position will show the information in the chest differently according to the points stated. So this is also a powerful feature for model building.

An example from an image of two types of position of the same patient that we clearly see is very different.

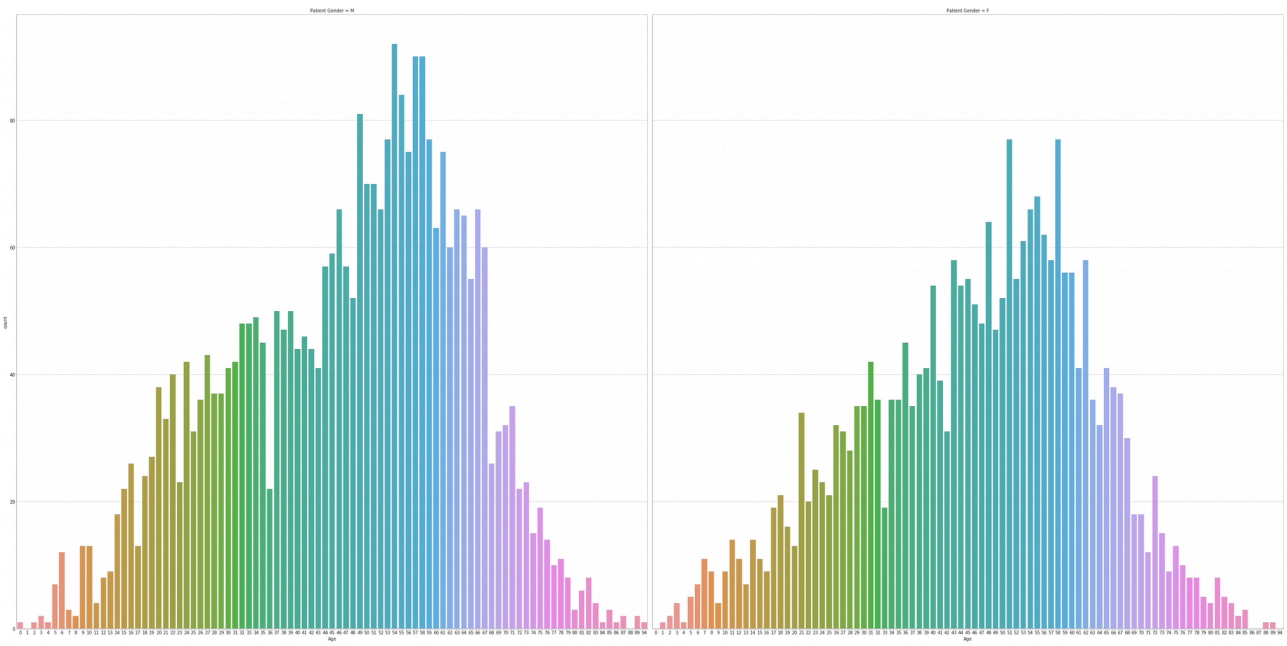




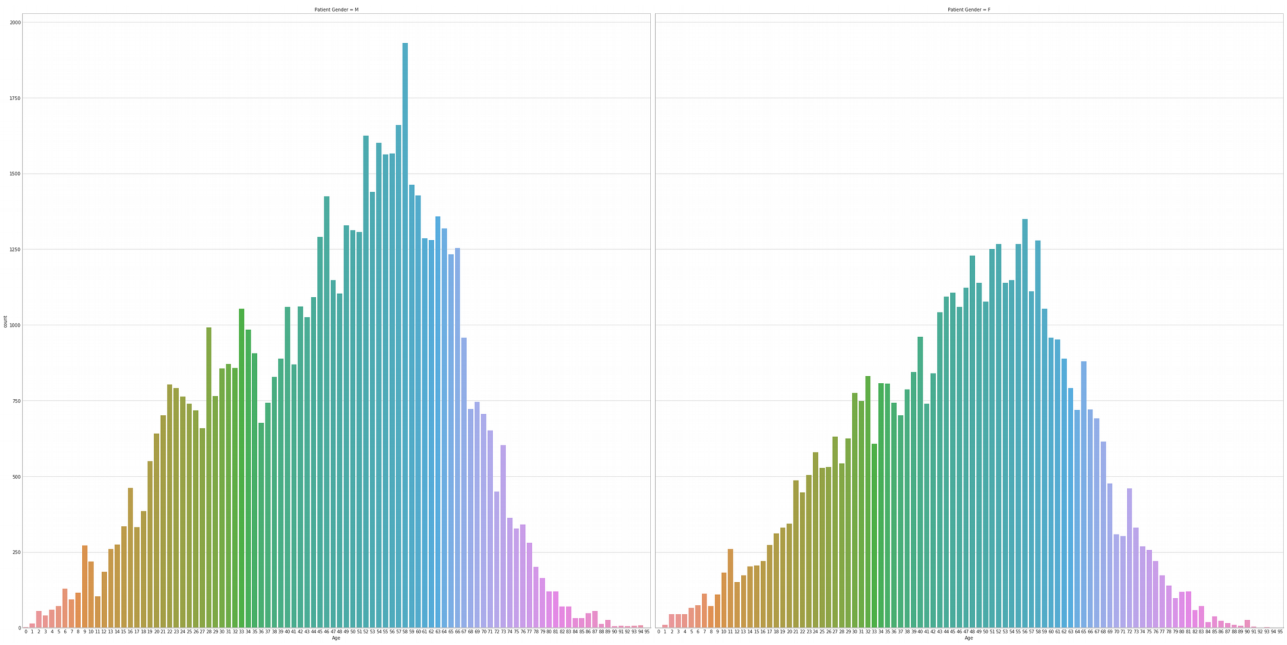
PA AP

Age distribution

Sample dataset:



Full dataset:



From this distribution it can be seen that middle-aged people are more likely to develop lung disease and that they are more likely to go for medical examinations. Young people are also focusing on early diagnosis.

For the purpose of this project is to distinguish whether a person has a lung disease or not, the data needed to build the model will be: X-ray, X-ray view position, Age, Gender.

c) Algorithms and Techniques

This is a problem with the new datasets that have never been fully modeled, so I do not want to get stuck, and so I come up with the following approach:

✗ Conclusion Neural Network

This is a powerful algorithm for processing image data like this. Given the huge data in the full dataset, this is indeed the appropriate method to apply, some parameters are considered and used as follows:

- Neural network architecture: Choose the appropriate architecture
- Preprocessing parameters
- Fine tuning
- Spacial transformer
- Training parameters
- Add more data in network not only images

✗ Capsules Network

With the power to distinguish many objects from different perspectives, I find it useful because our image data has two types of View Position. Just like CNN I have some things to do as follows:

- Capsules Networks architect: Choose the appropriate architecture
- Preprocessing parameter

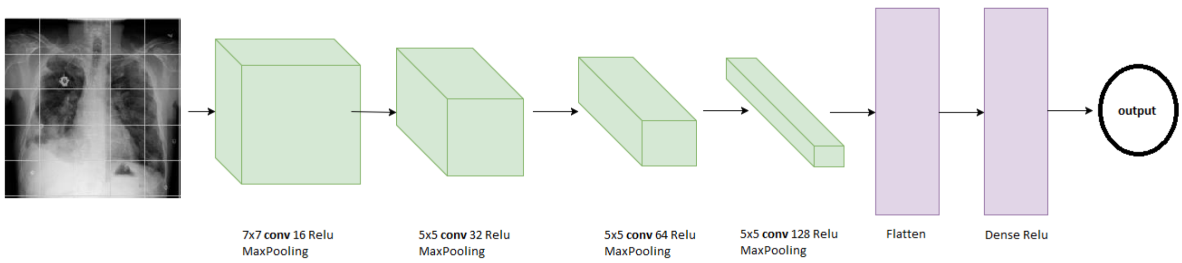
- Training parameters

d) Benchmark

For this problem, the benchmark model will be vanilla CNN model in notebook “vanilla CNN - FullDataset” and “vanilla CNN - SampleDataset”.

Since no team has ever built a complete model for this dataset just like what I intended to do in this project so a basic CNN model would be appropriate to compare to what's optimal and apply the Advanced algorithms.

Architecture for the CNN model vanilla as shown below



A basic CNN model with 4 Convolutional layers following each Convolution layer is MaxPooling. Convolution layers are increasing in depth. The next two classes to classify are Flatten and Dense.

CNN is the standard for handling such visual problems. With this simple model I will try to capture the pattern of diseases by ensuring that CNN will work with X-ray images.

3. Methodology

a) Data Preprocessing

My preprocessing data for both sample dataset and full dataset in “Data preprocessing” notebook consists of following steps:

- For images:
  - The images are rescaled, reducing the size of the image reduces the feature that makes training faster.
  - The images are converted to gray and rgb. Both are used for different models.
  - Read the image to numpy array then normalize by dividing the image matrix by 255
- For extra data:
  - Separate individual features into individual features.
  - Standardize the age field to the numeric form and, according to the year, then normalization field

- Remove the outliers point due to data about age too large.
- One hot two important attributes they will use are 'Patient Gender' and 'View Position'
- random both dataset

All data after processing is stored for later use.

The preprocessing process has the following adjustable parameters: Images resize shape

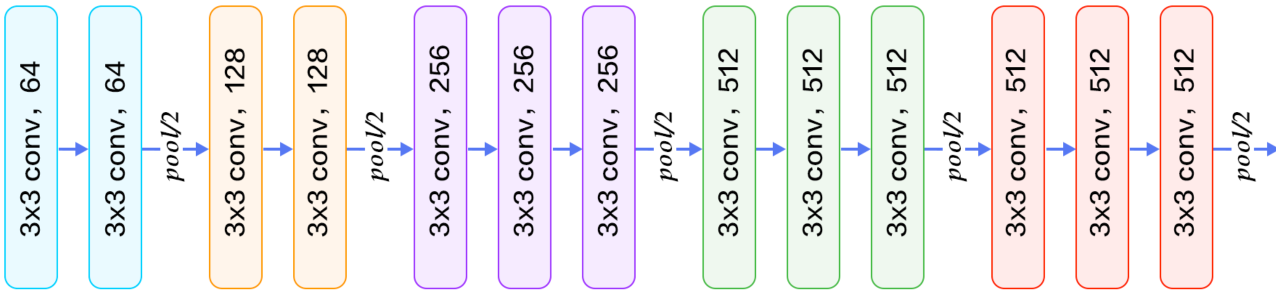
b) Implementation

We will go into the analysis of the two major options that will apply as follows:

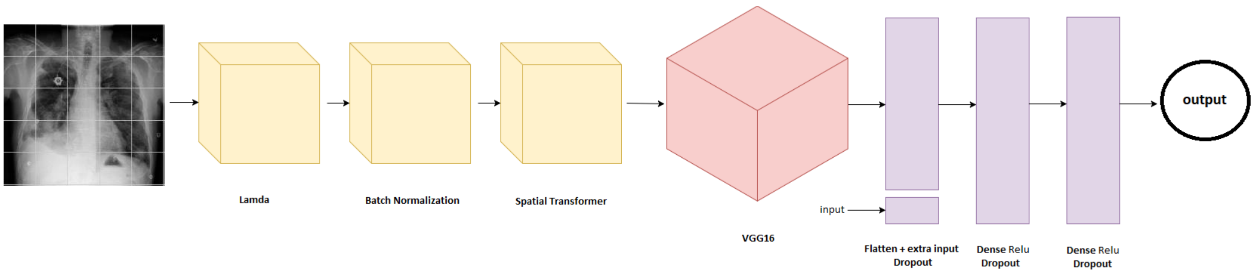
Optimized CNN

This is the main method of this project and can be seen on notebooks: “optimized CNN – FullDataset” and “optimized CNN – SampleDataset”.

Here is the Architecture for this approach



VGG16 architect extract feature



Full architecture

The architecture consists of three main layers in the following order:

- Spatial transformer layers (The first three layers)
  - The first is lamda to transfer the routing features [-0.5: 0.5], which means that the features of the image have an average value of 0.
  - The second is Batch Normalization.
  - The third layer is Spatial Transformer, which is used to extract the most valuable features for classification.

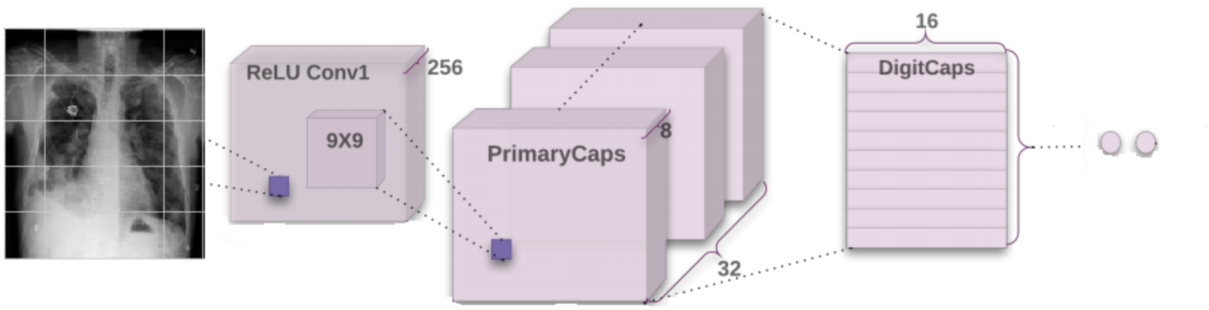
- Extract features layers (VGG16 pretrained model)
  - A set of 13 layers as shown in the first image of the VGG is the extract features, there are many pretrained model but now I am trying before with VGG16 because this is a simple model for learning time and training faster.
- Classification layers (Last 3 layers)
  - The first layer is the Flattened layer from the output of the layers VGG16 and 5 features plus 'Age', 'Gender Male', 'Gender Female', 'View position AP', 'View position PA'. These additional features will also affect the sorting, as we have seen above, so they are added to this layer. Following this layer is the dropout layer.
  - The next two layers are Dense after each Dropout layer, with a gradual decrease in depth.

The order of steps is as follows:

- Load data has been processed into ram, data is processed as before. Images are in rgb format.
- Implement Network architecture as designed by the architect.
- Implement Metric function including Binary accuracy, Precision score with threshold, Recall score with threshold, Fbeta score with beta and threshold.
- Implement data generator, model checkpoint, model loss function.
- Training model with training parameters, logging training / validation loss and training / validation accuracy.
- Save best model and test model on testingset.

Capsule Network

With the Capsule Network I had a slight change from the original Hinton architecture so that it could work well with this data set. Here is the architecture taken from the article by Hinton, I will clarify the changes right below:



Main parts of this model:

- Convolution layer with filters = 256, kernel_size = 9, strides = 2, padding = 'same', activation = 'relu'. This layer was changed from the original model from strides = 1 to strides = 2, the reason being that with the MNIST data set Hinton tested CapsNet, the image was 28x28, and the data set was 64x64, With strides = 2, the output of this model will be significantly reduced, and we will accept that we will get fewer features than strides = 1, since we have increased the strings so I think the output has been significantly reduced. , so I change from padding = 'valid' to padding = 'same'.

- PrimaryCaps with dim_capsule = 8, n_channels = 32, kernel_size = 9, strides = 2, padding = 'same', only changes with Hinton's achitecture in that the padding from 'valid' is replaced with 'same'.
- DigitCaps (I leave the same name that Hinton put) with num_capsule = n_class, dim_capsule = 16, fixed set routings of 3.

Thanks to the community who has created the layers for Capsnet on Keras for the community, so that I can change the architecture quickly.

Similar to CNN, the implementation steps are implemented in the following order:

- Load data has been processed into ram, data is processed as before. Photos are in gray format.
- Implement Network architecture as architect designed above with the parameters discussed.
- Implement Metric function including Binary accuracy, Precision score with threshold, Recall score with threshold, Fbeta score with beta and threshold. There is a slight change from CNN to the output shape (None, 2) instead of CNN with the output shape (None, 1).
- Implement data generator, model checkpoint, model loss function.
- Training model with training parameters, logging training / validation loss and training / validation accuracy.
- Save best model and test model on testingset.

c) Refinement

The above section says through some of my tweaks, let's be repeated and add:

All are implemented by Keras with the backend is tensorflow-gpu

Optimized CNN

Changing and experimenting with a lot of image sizes, I found that the 64x64 image size was small enough and good enough for the model to capture the pattern of the image.

Use the Spatial transformer with some layers supporting the front as lamda layer. The spatial transformer layer uses a fairly simple locnet (localization network) model to separate key features from the image.

Non-complementary data has been tested in many places on the architecture, and the first layer of the classification is most appropriate.

Tweaks the thresholds of precision, recall, and Fbeta score

Refine the index of the dropout layer in the classification

Parameter of optimezer Gradient descent with momemtum decay and learning rate

CapsNet

Convolution layer with filters = 256, kernel_size = 9, strides = 2, padding = 'same', activation = 'relu'. This layer was changed from the original model from strides = 1 to strides = 2, the reason being that with the MNIST data set Hinton tested CapsNet, the image was 28x28, and the data set in this project was 64x64, With strides = 2, the output of this model will be significantly reduced, and we will accept that we will get fewer features than strides = 1, since we have increased the strings so I think the output has been significantly reduced. , so I change from padding = 'valid' to padding = 'same'.

PrimaryCaps with dim_capsule = 8, n_channels = 32, kernel_size = 9, strides = 2, padding = 'same', only changes with Hinton's architecture in that the padding from 'valid' is replaced with 'same'.

Implement Metric function including Binary accuracy, Precision score with threshold, Recall score with threshold, Fbeta score with beta and threshold. There is a slight change from the CNN to the output shape (None, 2) instead of the CNN with the output shape (None, 1).

Training parameters are also available to suit the machine configuration such as batch size = 32, learning rate, ...

4. Results

I would like to use some abbreviations for the models I tested:

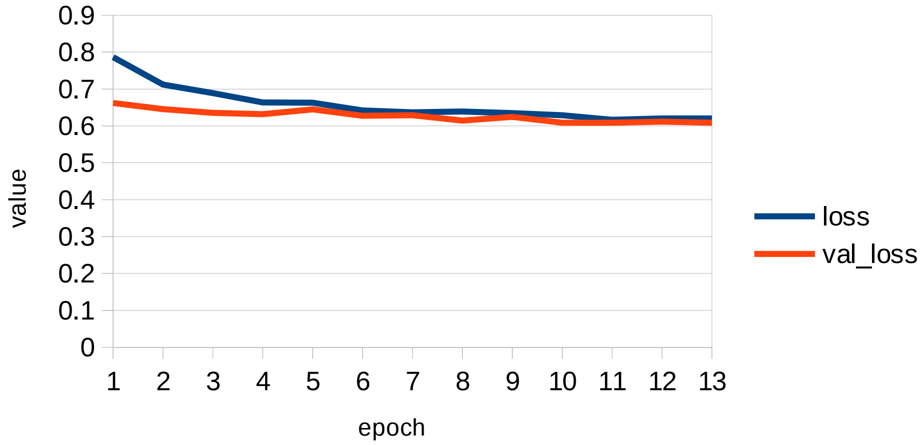
- Vanilla rgb: Vanilla CNN model for rgb images
- Vanilla gray: Vanilla CNN model for gray images
- CNN + VGG: optimized model with VGG16 pretrained model
- CNN + VGG + data: optimized model with VGG16 pretrained model and extra data
- CNN + VGG + data + STN: optimized model with VGG16 pretrained model and extra data and Spatial Transformer
- CapsNet basic: Capsule Network with Hinton’s architecture
- CapsNet changed: Capsule Network with my changed architecture

a) Model Evaluation and Validation

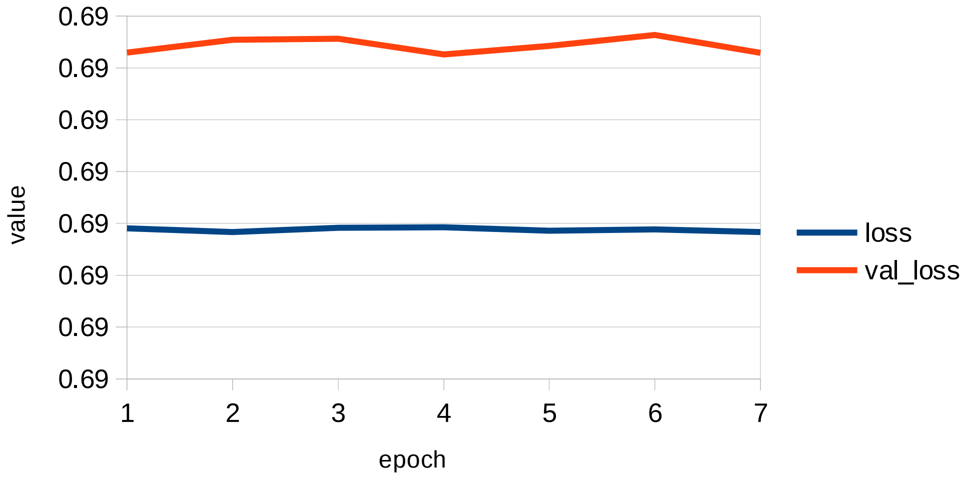
During development, a validation set was used to evaluate the model.

The following is a change chart of loss in training algorithms corresponding to sample dataset and full dataset

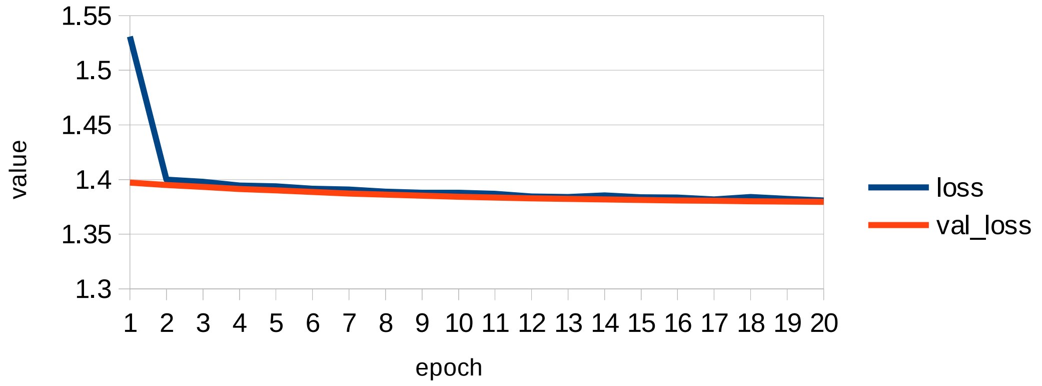
Sample dataset: CNN + VGG + data + STN



Sample dataset: Vanilla CNN

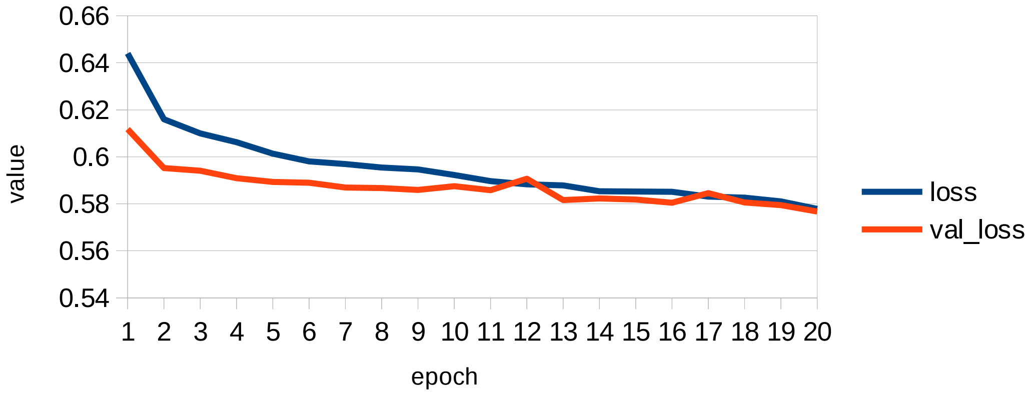


Sample dataset: CapsNet

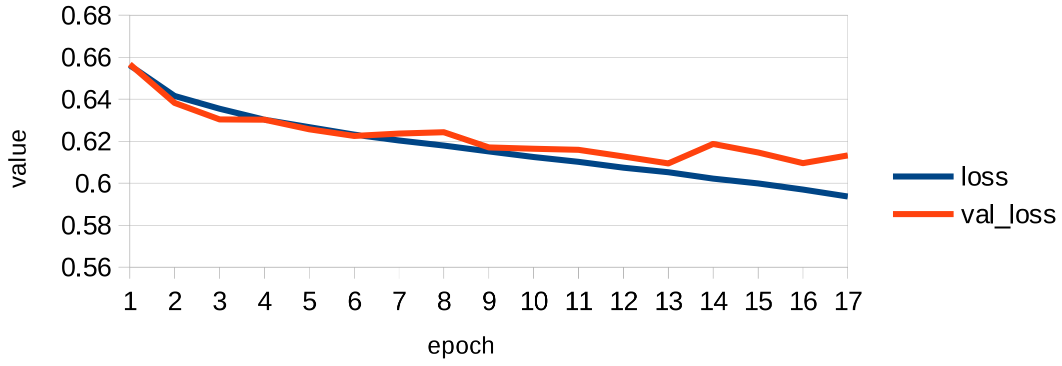


From the above three charts, it is easy to see that the vanilla CNN is running the worst, it overfilled very early and stopped because of my EarlyStopping checkpoint model. CapsNet seems to be working but convergence is too slow. CNN + VGG + data + STN seems to work well but is converging too slowly, perhaps due to too little data on the number of features from the image is too large, so we need more data in the full dataset .

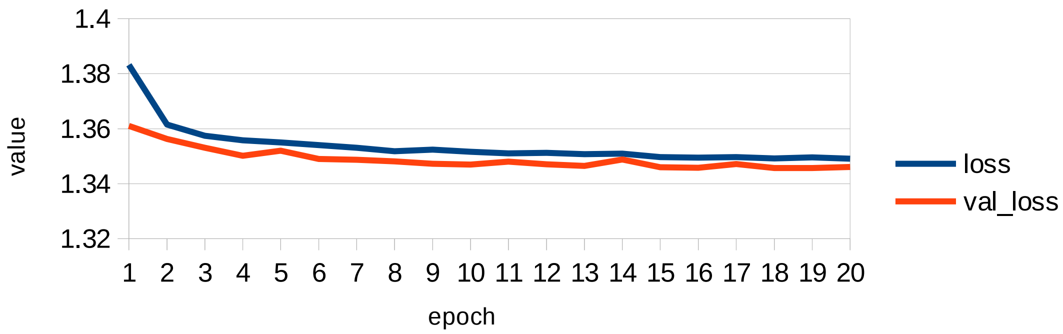
Full dataset: CNN + VGG + data + STN



Full dataset: vanilla CNN



Full dataset: Capsnet



All three of these models performed better than the sample dataset, and found that the CNN vanilla was overfilled and stopped by the Earlystopping model, CNN + VGG + data + STN, with more sophisticated techniques as demonstrated. The convergence is fast and is still very good convergence, will also have better results if I train this model with more epoch. CapsNet is better, but slower convergence and almost impossible to change the value of losses.

From the above charts, it was found that CNN + VGG + data + STN is the best with the specific parameters as mentioned above, please be supplemented.:

The architecture consists of three main layers in the following order:

- Spatial transformer layers (The first three layers)
    - The first is lamda to transfer the routing features [-0.5: 0.5], which means that the features of the image have an average value of 0. The lamda function is "lambda x: 2 * x - 1."
    - The second is Batch Nomalization
    - The third layer is the Spatial Transformer for extracting the most valuable features for classification with the net localization of a small CNN as follows:
        - The 3 Convolution layers followed by Maxpooling, the Convolution layers have a doubling depth of 16 - 32 - 64 with a kernel size of 7 - 5 - 3, padding = 'valid'
        - Layer Flatten followed by 3 dense layers with diminished depth, activation = 'elu'

- Extract features layers (VGG16 pretrained model)
  - A set of 13 layers as shown in the drawing of the VGG is the task of extracting features, there are many pretrained model but now I am trying before with VGG16 because this is a simple model for learning time and training faster.
- Classification layers (Last 3 layers)
  - The first layer is the flattened layer from the output of the layers VGG16 and 5 features plus 'Age', 'Gender Male', 'Gender Female', 'View position AP', 'View position PA'. These additional features will also affect the sorting, as we have seen above, so they are added to this layer. Following this layer is the dropout layer.
  - The next two layers are Dense after each layer. Dropout is 0.2, with decreasing depth.

The selected model has not really had small loss and rapid convergence, so still need more improvement can be resolved to identify lung disease in the hospital. I still look forward to more time, computer power to continue to improve this model much better.

b) Justification

A comparison of the accuracy of the methods on two sets of data sets: sample dataset and full dataset, the following evaluations were performed on the data set for testing differed from training and validation:

Dataset	Architecture	Precision	Recall	F 0.5 score	Accuracy	training time/ epoch	no. parameters
Sample Dataset	Vanilla rgb	0.617	0.589	0.611	0.503	2 s	322793
	Vanilla gray	0.577	0.48	0.555	0.517	2 s	321225
	CNN + VGG	0.645	0.555	0.624	0.667	16 s	15252133
	CNN + VGG + data	0.647	0.588	0.634	0.675	16 s	15240769
	CNN + VGG + data + STN	0.642	0.614	0.636	0.677	19 s	15488051
	CapsNet basic	0.614	0.599	0.611	0.581	75 s	14788864
	CapsNet changed	0.735	0.073	0.261	0.575	37 s	12167424
Full Dataset	Vanilla rgb	0.672	0.594	0.655	0.672	53 s	322793
	Vanilla gray	0.672	0.572	0.649	0.667	51 s	321225
	CNN + VGG	0.675	0.619	0.663	0.688	384 s	15252133
	CNN + VGG + data + STN	0.684	0.621	0.67	0.693	431 s	15488051
	CapsNet basic	0.64	0.498	0.605	0.635	1815 s	14788864
	CapsNet changed	0.625	0.474	0.588	0.625	856 s	12167424

My best model is CNN + VGG + data + STN, which is better than the benchmark Vanilla CNN. It can be seen that the important indicator that they have defined is the F 0.5 score of the best model is 0.67. Training time is bigger than Vanilla CNN, but my best model is still focused and can be improved by continuing training with more epochs as the above analysis.

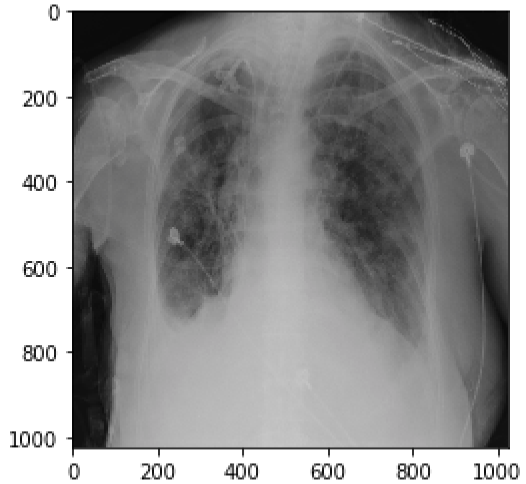
The CapsNet model does not seem to work well, the number of parameters is only equivalent to my best model but the training time is much longer.

As a result of my best model - CNN + VGG + data + STN is F 0.5 score = 67% with accuracy of 69.3%, it still does not meet the requirement to use in hospitals, need more time and computer power to further analyze the data, improve the algorithm can meet the requirements. However, this is also a good first step, and this result is very good when the normalized data set is public and there are many mistakes in labeling.

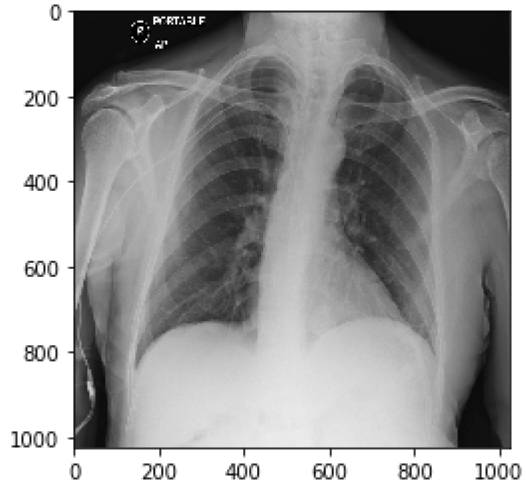
5. Conclusion

a) Free-Form Visualization

In the "Demo" notebook, I tested with 20 random cases, the user, either a physician or a patient, just filled in information about X-rays, age, gender, and view position. Basically judging that the case is ill, before proceeding with the analysis on more important trials, remember that we have chosen F beta score with small beta means that we are focusing on Be sure to diagnose the condition in case of shock and sadness before formal examinations.

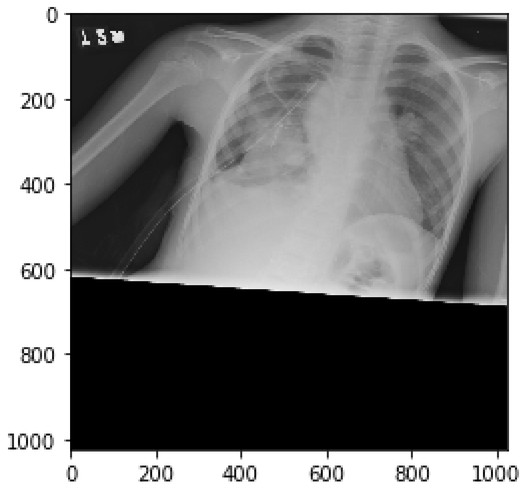


True: Fibrosis, Predict: Finding, confident: 0.58584183



True: No Finding, Predict: No Finding, confident: 0.074103214

Most of the results are exactly the same as above, but there are also some cases that are wrong as follows



True: Pneumothorax, Predict: No Finding, confident: 0.48332688

But I'm still glad that this misadventure predicts that the sick person is not sick, avoid shocking people and need more tests before the doctor gives the final diagnosis. The Confident of this case is 0.48 which is close to the threshold I have chosen as 0.5, meaning that there is almost half the chance that this person is ill.

b) Reflection

In this project I proposed the diagnosis of lung disease from the patient's X-ray data plus some additional information. The best solution is to have a complex CNN with the following data processed:

- Research for resolved issues, domain information, support data, methods, and solution data for similar projects. Some potential techniques are listed and investigated.
- Sample data is downloaded and analyzed, preprocessing, metric selection
- Testing multiple architectures, optimizing and testing on a sample dataset.
- Use good architects to test the full dataset, continue optimizing and statistics.

This project is based on a very new set of data and not many people find out, this is a very good problem and if done well it will make a big contribution to the community. This project has tested many new and interesting methods such as Spatial Transformer and Capsule Network and has shown that they have recorded remarkable results.

This project is hardly new, and X-rays are difficult to see clearly, the data is not standardized, and NLP labeling can be used to obtain the disease. It is also difficult to apply a very new method of Capsule Network so there is not much documentation to optimize it. Big data on the full dataset is also a big challenge for me being limited to computer power.

The results of this project have achieved my initial expectation, but to be able to apply in hospitals, more improvements are needed to increase the precision of the model.

c) Improvement

I think with this project there may be more to do to increase our results like:

- Testing the model to distinguish each type of disease rather than just having the disease, so one of the important things is that I will have to solve the data problem for each disease is very skew.
- Training with more epoch, change some parameters to faster convergence models such as learning rate.
- Increasing the size of training shots, this will increase the chance of getting important features, but it also means that the model will be more complex and the training time and prediction will be longer.
- For optimized CNN using VGG, we can experiment on many other pretrained models.
- For the Spatial Transformer layer I will experiment with more complex localization network.
- Some metric parameters of the metrics will also be tested more.
- For CapsNet I want to try adding some more layers so that it can extract more features, but the training time will be very long.

There are so many things to do so I need support for a very powerful server, I will have to seek help now.

References

1. https://medicalxpress.com/news/2017-08-lung-diseases-million.html
2. Andrew Ward, Nicholas Bambos. *Quantum Annealing Assisted Deep Learning for Lung Cancer Detection*. http://cs231n.stanford.edu/reports/2017/pdfs/534.pdf
3. Albert Chon, Niranjan Balachandar. *Deep Convolutional Neural Networks for Lung Cancer Detection*. http://cs231n.stanford.edu/reports/2017/pdfs/518.pdf
4. Kingsley Kuan, Mathieu Ravaut, Gaurav Manek, Huiling Chen, Jie Lin, Babar Nazir, Cen Chen, Tse Chiang Howe, Zeng Zeng, Vijay Chandrasekhar. *Deep Learning for Lung Cancer Detection: Tackling the Kaggle Data Science Bowl 2017 Challenge*. https://arxiv.org/abs/1705.09435
5. https://www.nature.com/articles/srep46479
6. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5569872/
7. https://stephenson.pure.elsevier.com/en/publications/computer-aided-lung-cancer-diagnosis-with-deep-learning-algorithm
8. https://www.researchgate.net/publication/301651102_Computer_aided_lung_cancer_diagnosis_with_deep_learning_algorithms
9. *Matrix capsules with EM routing.* https://openreview.net/forum?id=HJWLfGWRb&noteId=HJWLfGWRb
10. Sara Sabour, Nicholas Frosst, Geoffrey E Hinton. *Dynamic Routing Between Capsules.* https://arxiv.org/abs/1710.09829
11. *NIH sample Chest X-rays dataset*, https://www.kaggle.com/nih-chest-xrays/sample
12. *NIH full Chest X-rays dataset*, https://www.kaggle.com/nih-chest-xrays/data
13. Max Jaderberg, Karen Simonyan, Andrew Zisserman, Koray Kavukcuoglu. *Spatial Transformer Networks*. https://arxiv.org/abs/1506.02025
14. https://www.womenshealth.gov/a-z-topics/lung-disease
15. https://www.cdc.gov/cancer/lung/statistics/
16. https://www.nih.gov/news-events/news-releases/nih-clinical-center-provides-one-largest-publicly-available-chest-x-ray-datasets-scientific-community
17. https://www.slideshare.net/drvikrampatil/approach-to-chest-xray-and-interpretation
18. Traffic Sign Recognition using CNN with Learned Color and Spatial Transformation. https://github.com/hello2all/GTSRB_Keras_STN/blob/master/capstone_report.md
19. CapsNet-Keras. https://github.com/XifengGuo/CapsNet-Keras