LUNG DISEASE CLASSIFICATION USING CHEST X-RAY IMAGES

Final report

1. Introduction

Respiratory diseases are leading causes of death and disability in the world. We take our breathing and our respiratory health for granted, but the lung is a vital organ that is vulnerable to airborne infection and injury About 65 million people suffer from chronic obstructive pulmonary disease (COPD) and 3 million die from it each year, making it the third leading cause of death worldwide. About 334 million people suffer from asthma, the most common chronic disease of childhood affecting 14% of all children globally. Pneumonia kills millions of people annually and is a leading cause of death among children under 5 years old. Over 10 million people develop tuberculosis (TB) and 1.4 million die from it each year, making it the most common lethal infectious disease. Lung cancer kills 1.6 million people each year and is the most deadly cancer. Globally, 4 million people die prematurely from chronic respiratory disease. At least 2 billion people are exposed to indoor toxic smoke, 1 billion inhale outdoor pollutant air and 1 billion are exposed to tobacco smoke. The truth is that many of us are naïve to these stark realities.

Prevention, control and cure of these diseases and promotion of respiratory health must be a top priority in global decision-making in the health sector. These goals are achievable, and the control, prevention and cure of respiratory diseases are among the most important cost effective health interventions available.

The goal of this project is to demonstrate the utilisation of modern and quickly deployable deep learning techniques to analyse and evaluate chest-x-ray images to diagnose the disease of the patient, with reasonable accuracy. The ultimate goal is to build systems to automate the initial stages of respiratory disease diagnosis, thereby easing the immense pressure of doctors and medical professionals in the current times of crisis.

2. Current scenario:

CXRs are conventionally analysed by medical experts and doctors, to diagnose the patient. Though human evaluation is very accurate, in times of emergency, or while dealing with a large number of cases, medical staff come under extreme pressure to sieve through the huge number of images. Every case has to be carefully observed and analysed by doctors for correct diagnosis. This is a time consuming process. Moreover, human analysis is prone to errors. We need a model that can quickly filter the high criticality cases from the moderate ones, and can give doctors the opportunity to focus on critical cases.

3. Brief discussion about Radiography and X-Ray imaging

When imaging with X-rays, an X-ray beam produced by a so-called X-ray tube passes through the body. On its way through the body, parts of the energy of the X-ray beam are absorbed. This process is described as attenuation of the X-ray beam. On the opposite side of the body, detectors or a film capture the attenuated X-rays, resulting in a clinical image. In conventional radiography, one 2D image is produced. In Computed Tomography, the tube and the detector are both rotating around the body during the examination so that multiple images can be acquired, resulting in a 3D visualization. The most common methods of X-ray in medical imaging are X-ray radiography, computed tomography (CT), mammography, angiography and fluoroscopy. Different organs and tissues have a different sensitivity to radiation.

How does the procedure work?

Despite the development of newer technologies such as computed tomography (CT), ultrasound imaging and magnetic resonance imaging (MRI), plain film X-rays remain an important tool for the diagnosis of many disorders. In radiography, a beam of X-rays, produced by an X-ray generator, is transmitted through an object, for example, the chest of the patient. The X-rays are absorbed by the material they pass through, in differing amounts depending on the density and composition of the material. X-rays that are not absorbed pass through the object and are recorded on X-ray sensitive film. Such areas appear completely black.

While bones absorb X-rays particularly well, soft tissue such as muscle fiber, which has a lower density than bone, absorbs very few X-rays. This results in the familiar contrast seen in X-ray images, with bones shown as clearly defined white areas and darker areas of tissue.

This makes conventional X-rays very suitable for scans of bones and tissue dense in calcium such as in dental images and detection of bone fractures. Other uses of radiography include the study of the organs in the abdomen, such as the liver and bladder; chest radiography for diseases of the lung, such as pneumonia or lung cancer and mammography to screen for breast cancer. X-ray fluoroscopy is used to detect a number of diseases associated with the stomach and intestine, genitals and urinary tract.

Common uses of X-Ray Radiography

The chest x-ray is performed to evaluate the lungs, heart and chest wall.

A chest x-ray is typically the first imaging test used to help diagnose symptoms such as:

- breathing difficulties
- a bad or persistent cough
- chest pain or injury
- fever

Physicians use the examination to help diagnose or monitor treatment for conditions such as:

- Pneumonia

- Heart failure and other heart problems
- Emphysema
- Tuberculosis
- COVID-19 diagnosis
- Lung cancer
- Positioning of medical devices
- Fluid or air collection around the lungs
- Other medical conditions

4. Problem statement and goal of the project:

   We are given chest x-ray (CXR) images of patients (front view of the patient's chest). The images have been labelled as belonging to a normal patient or a diseased patient. In my project, I have collected CXR images belonging to 5 types of diseases.
   The task is to build a deep learning model using currently available deep learning tools that can analyse the image and label an x-ray as being healthy or diseased.

5. Value to client:

   Conventionally, the client in this type of project would be hospitals, clinics, radiology staff, doctors, etc. But considering the current crisis, this model has the potential to go beyond commercial purposes and help us in fighting the COVID pandemic. Deploying a model to carry out mundane tasks in the initial stages of diagnosis, like filtering out low priority cases (where the patients are clearly healthy), separating the critical cases from the mild ones, etc. can be carried out by a machine and the results can be presented to doctors for deeper analysis. This will reduce the stress on doctors, and they will have the opportunity to focus on demanding cases which need critical attention.

6. The Stakeholders:

   When such a project is implemented, the medical practitioners and hospitals are the stakeholders. The diagnosis and treatment of patients will be directly linked to how well the model performs. This will impact the reputation of the medical institution. But I believe that the patients are at the highest risk here. The accuracy of the model will directly and severely affect the life and general well being of patients. If an unhealthy person is incorrectly classified as healthy, that patient will not receive treatment, leading to further complications, or death in extreme cases. This will be a disaster.

7. Note on model performance: Importance of precision and recall

   Classification models are conventionally evaluated on the basis of the accuracy achieved. In medical classification, we can make either a type 1 or a type 2 error. When we incorrectly label a healthy person as diseased, this is a Type-1 error. Labelling a diseased person as healthy is a type 2 error. Clearly, committing type 2 errors are disastrous in the medical domain. Making type 1 errors

is forgivable, and easily corrected. For example, if a healthy person is labelled as having pneumonia, we can do another test, and compare the results of both tests to see if that is the truth. If a person is labelled as having pneumonia in 2 medical tests, then it is very likely that he has pneumonia. We can use Bayesian statistics here to find out the exact probability that the person actually has pneumonia, given that he has tested positive twice.

But the converse is not true. When a person is labelled as being healthy by a test, a second test is rarely conducted. The subsequent positivity of being healthy clouds mathematical judgement, and almost no one performs a second test. If a diseased person is described as being healthy, it's a DISASTER. There is no other term to describe it. The person will be unaware of the disease, and will go on with life as usual. This may lead to degradation in his/her medical condition, and even death in extreme cases. Hence, in medical classification, especially when we are trying to detect the presence of disease, having a high recall is extremely important. Precision is secondary. This becomes even more important when machine classification is the only mode of preliminary diagnosis. Although such a scenario rarely exists, machine evaluation is always adjudicated by human experts. In any case, for our model to be credible, it should not let any disease case go undetected, even if we need to handle a few false alarms along the way.

8. Data Sources and organization:

I have compiled the dataset from various Kaggle datasets as well as academic datasets. As medical data is protected information, it is a little difficult to find good quality CXR images with proper labelling. The same images are repeated across many datasets. I have sieved through multiple datasets and created a set of total 11276 images, with:

2530 CXR images Bacterial Pneumonia
288 CXR images COVID-19
1122 CXR images which are Normal (no disease)
5597 CXR images of Other Findings
394 CXR images of TB
1345 CXR images of Viral Pneumonia

As labelled CXR image data was not easily available, I am trying to use all images for training. As you can see, there is class imbalance in the dataset.

We will test our initial model on this set of 11276 images. Later, we will try other methods like data augmentation and image resizing. Also, we will create a smaller but more balanced random subset of this dataset, having about 300-500 images for each category. The number of images will be reduced to about 3000. This will reduce training time and we will be able to experiment with different methods easily. I will use the same training models and parameters as used for the original dataset, and compare the accuracy for both.

I have used the following datasets to compile mine:

For Normal and Pneumonia CXRs:

https://www.kaggle.com/paultimothymooney/chest-xray-pneumonia -

Chest X-ray images (anterior-posterior) were selected from retrospective cohorts of pediatric patients of one to five years old from Guangzhou Women and Children's Medical Center, Guangzhou. All chest X-ray imaging was performed as part of patients' routine clinical care.

For the analysis of chest x-ray images, all chest radiographs were initially screened for quality control by removing all low quality or unreadable scans. The diagnoses for the images were then graded by two expert physicians before being cleared for training the AI system. In order to account for any grading errors, the evaluation set was also checked by a third expert.

There are 5,863 X-Ray images (JPEG) and 2 categories (Pneumonia/Normal), and pneumonia images can be further classified into Viral or Bacterial Pneumonia.

Original dataset - https://data.mendeley.com/datasets/rscbjbr9sj/2
Citation - http://www.cell.com/cell/fulltext/S0092-8674(18)30154-5

For COVID-19 CXRs -

https://github.com/agchung/Figure1-COVID-chestxray-dataset

Citation:

- DarwinAI Corp., Canada and Vision and Image Processing Research Group, University of Waterloo, Canada
    - Linda Wang
    - Alexander Wong
    - Zhong Qiu Lin
    - Paul McInnis
    - Audrey Chung
    - Hayden Gunraj, COVIDNet for CT: Coming soon.
- Vision and Image Processing Research Group, University of Waterloo, Canada
    - James Lee
- Matt Ross and Blake VanBerlo (City of London), COVID-19 Chest X-Ray Model: https://github.com/aildnont/covid-cxr
- Ashkan Ebadi (National Research Council Canada)
- Kim-Ann Git (Selayang Hospital)
- Abdul Al-Haimi

https://www.kaggle.com/tawsifurrahman/covid19-radiography-database

The above dataset was developed with help of images from the Italian Society of Medical and Interventional Radiology (SIRM) COVID-19 DATABASE [1], Novel Corona Virus 2019 Dataset developed by Joseph Paul Cohen and Paul Morrison and Lan Dao in GitHub [2] and images extracted from 43 different publications. References of each image are provided in the metadata. Normal and Viral pneumonia images were adopted from the Chest X-Ray Images (pneumonia) database

Citation:

M.E.H. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M.A. Kadir, Z.B. Mahbub, K.R. Islam, M.S. Khan, A. Iqbal, N. Al-Emadi, M.B.I. Reaz, M. T. Islam, “Can AI help in screening Viral and COVID-19 pneumonia?” IEEE Access, Vol. 8, 2020, pp. 132665 - 132676.

https://www.kaggle.com/nabeelsajid917/covid-19-x-ray-10000-images
By Nabeel Sajid

For TB CXRs -

https://www.kaggle.com/kmader/pulmonary-chest-xray-abnormalities

Citation:

- Jaeger S, Karargyris A, Candemir S, Folio L, Siegelman J, Callaghan F, Xue Z, Palaniappan K, Singh RK, Antani S, Thoma G, Wang YX, Lu PX, McDonald CJ. Automatic tuberculosis screening using chest radiographs. IEEE Trans Med Imaging. 2014 Feb;33(2):233-45. doi: 10.1109/TMI.2013.2284099. PMID: 24108713
- Candemir S, Jaeger S, Palaniappan K, Musco JP, Singh RK, Xue Z, Karargyris A, Antani S, Thoma G, McDonald CJ. Lung segmentation in chest radiographs using anatomical atlases with nonrigid registration. IEEE Trans Med Imaging. 2014 Feb;33(2):577-90. doi: 10.1109/TMI.2013.2290491. PMID: 24239990

For CXRs of numerous other lung diseases -

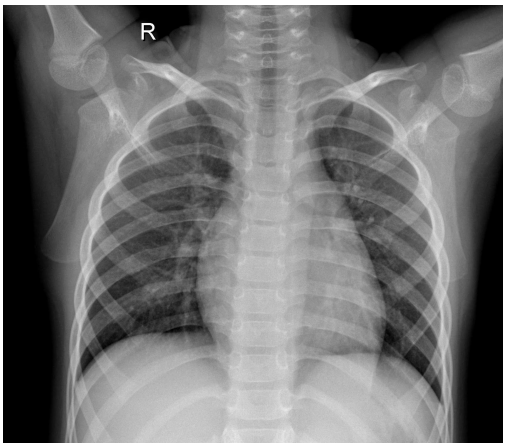
https://www.kaggle.com/nih-chest-xrays/sample (It is a randomly selected subset of the full NIH Chest X-Ray dataset (42 GB))

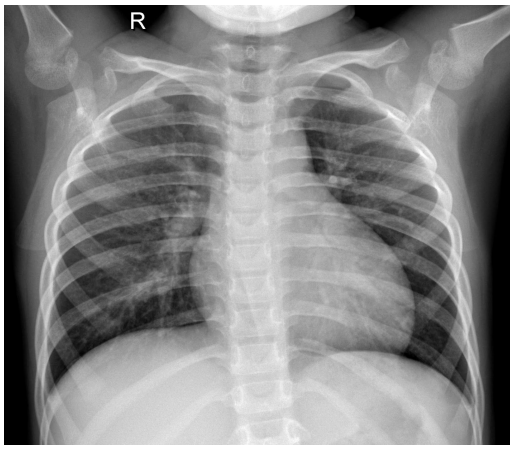
Citation:

- 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, ChestX-ray8*Hospital-Scale*Chest*CVPR*2017_paper.pdf
- NIH News release: NIH Clinical Center provides one of the largest publicly available chest x-ray datasets to scientific community
- Original source files and documents: https://nihcc.app.box.com/v/ChestXray-NIHCC/folder/36938765345

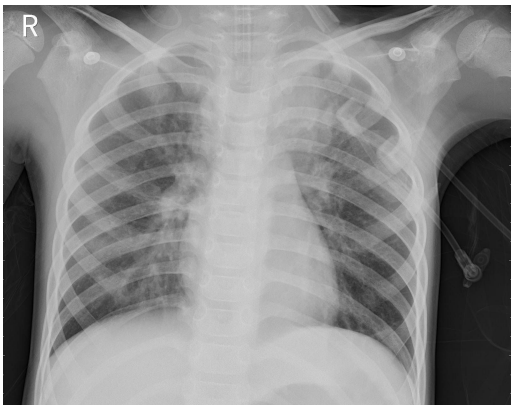
9. Image samples:

NORMAL:

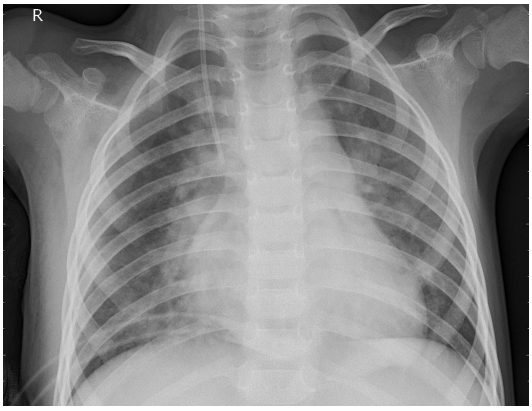


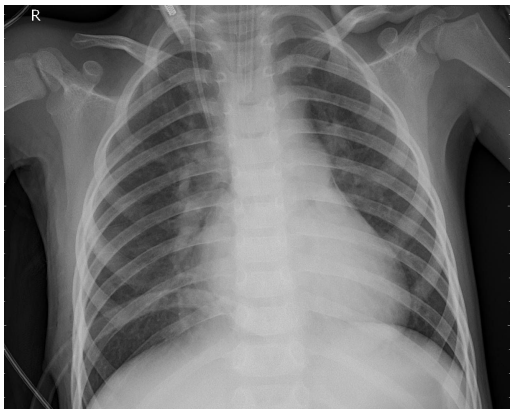


VIRAL PNEUMONIA:



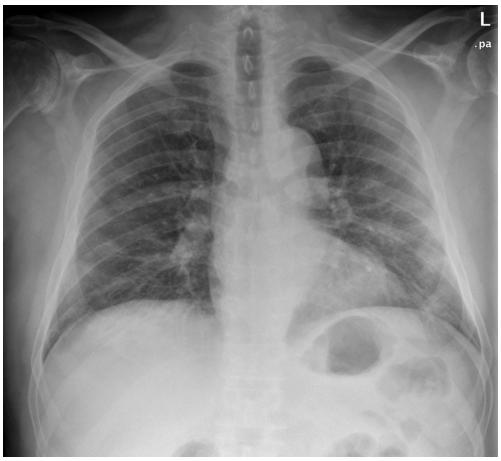
BACTERIAL PNEUMONIA:

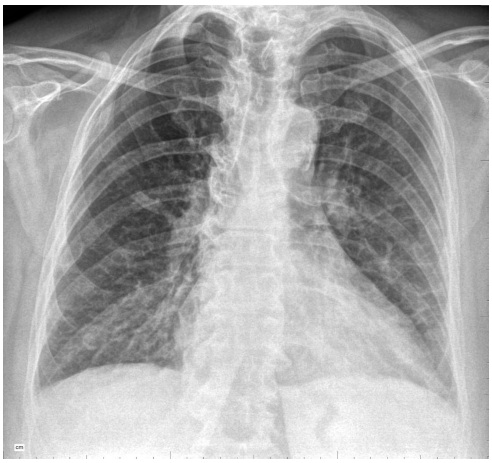




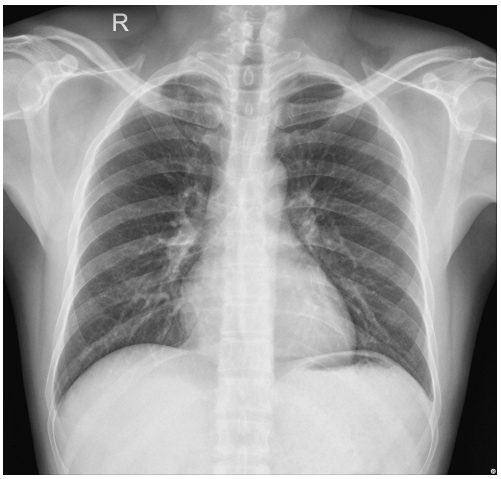
Taking a look at the images with naked eye shows that bacterial pneumonia typically exhibits a focal consolidation, .i.e. a region of aggregated “material”, whereas viral pneumonia (right) manifests with a more diffuse ‘‘interstitial’’ pattern in both lungs.

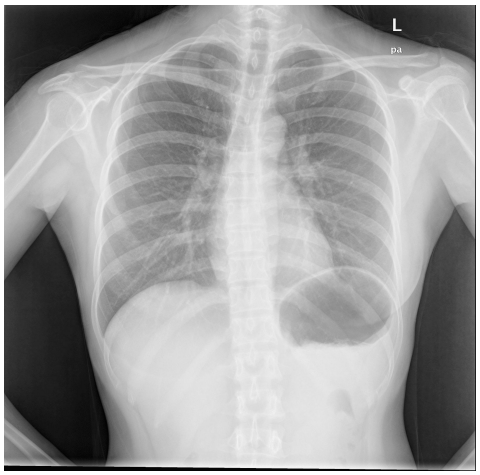
COVID-19:



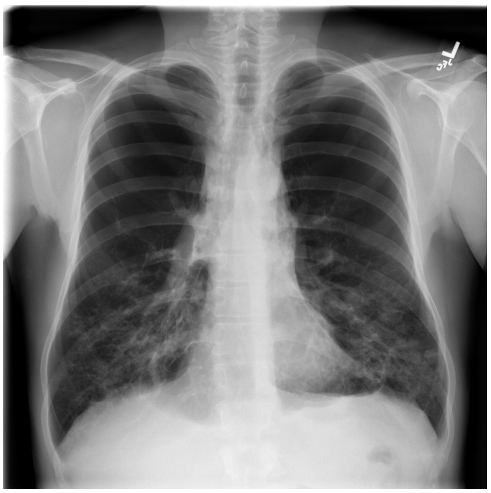


TUBERCULOSIS:





OTHER FINDINGS:



10. Preliminary Observations with naked eye

    We do not possess radiology knowledge to analyse these images and correctly diagnose the disease. But it is visible that ‘Normal’ CXRs are more “clear”, without any consolidated material or fluids obstructing the path of X-Rays. Diseases CXRs have more whitish or grayish regions spread throughout the lung area.
    For example, bacterial pneumonia typically exhibits a focal consolidation, whereas viral pneumonia manifests with a more diffuse ‘‘interstitial’’ pattern in both lungs.
    Likewise, CXRs of COVID-19 also show opacity inside the lung area, representing development of or aggregation of fluid-like substances.

11. Basics of Chest X-Ray Radiology:

    Regions of different opacity in the CXR images can be broadly divided into the following categories:

    a. Black: These areas offer no resistance to X-rays - mostly air and empty space
    b. Dark Grey (very slightly opaque): fat and subcutaneous tissue
    c. Light Grey (mildly opaque): soft tissue (heart, blood vessels, etc.)

d. Off white: Bones
e. Bright white: metal implants (buttons, heart stents, nuts, etc.)

12. Major milestones in model building:

    a. Training on the images as they are, without any preprocessing
    b. Using data augmentation to increase the number of image in COVID-19 (underrepresented class)
    c. Using a smaller subset to quickly experiment on learning rates
    d. Using progressive image resizing to increase accuracy.

13. Neural networks and importance of learning rates:

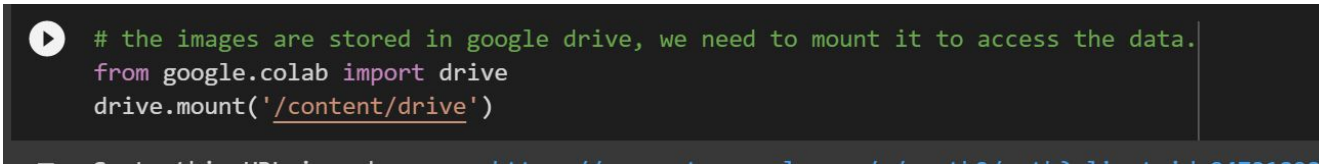
    In traditional machine learning models like linear regression, decision trees, etc. it's somewhat possible to visualise how the model is working. Also, there are a variety of hyperparameters which can be tuned to optimise training.

    Neural networks irk on a single simple principle, making a network of nodes, like a graph, optimising the weights of the graph to minimise loss. Although we know that the loss of the network with respect to a given weight is minimised using gradient descent, but its not easy to visualise this in multiple dimension. There are complex mathematical calculations involved. Hence, after a certain point, it is convenient to treat neural networks as black boxes. The only thing that we can decide explicitly is the learning rate at which the network is trained.

    In our project we will try to work with different learning rates and see how accuracy varies

14. Basic training without preprocessing:

    A. Gathering the and organising data:
    As explained above, the dataset is curated with the help of numerous medical imaging datasets available in Kaggle and other sources. The complete dataset contained about 11276 images. Storing such a huge number of images and loading them into RAM for processing is not feasible with general purpose desktop computers. Hence, I have uploaded the images to my google drive account.
    While modelling, I mount my drive and directly train on images from google drive.
    B. Loading the data for modelling:
    As training the model on a huge number of images is not possible locally, I am training building my model using a Google Colab notebook. It provides 12GB of RAM and an equivalent capacity GPU for processing. All these resources are available for free. We can mount the images in Colabinstance from google drive as follows:



After running this cell, we need to navigate to a link and login using our credentials to give drive access to Colab. After access is provided, we get a message like this:

```
Go to this URL in a browser: https://accounts.g

Enter your authorization code:
··········
Mounted at /content/drive
```

C. Loading the “fast.ai” module for training, and declaring the path to our data:

```
[ ] from fastai import *
    from fastai.vision import *
    from pathlib import Path

[ ] # path to parent folder
    data_path = '/content/drive/My Drive/Colab Notebooks/Chest X-Ray Classification'
    classes = ['Bacterial Pneumonia','COVID-19','Normal','Other Findings','TB','Viral Pneumonia']
```

D. Reading the image files from the directory:

As seen below, we have the following counts for various types of images:

```
We have 2530 CXR images of type Bacterial Pneumonia
We have 288 CXR images of type COVID-19
We have 1122 CXR images of type Normal
We have 5597 CXR images of type Other Findings
We have 394 CXR images of type TB
We have 1345 CXR images of type Viral Pneumonia
We have 11276 images in total.
```

E. Creating the local ‘ImageDataBunch’ for training:

The fast.ai library has an image data class for processing image data. It’s called an ImageDataBunch. While creating this, we need to specify the path here images will be found, the ratio of training to validation image, and the size of the images that will be used for training (original images will be scaled down to our desired size).

```
# creating the ImagedataBunch
np.random.seed(42)
data = ImageDataBunch.from_folder(data_path, train=".", valid_pct=0.2,
        ds_tfms=get_transforms(), size=224, num_workers=4).normalize(imagenet_stats)
```

```
[ ]  data.classes, data.c, len(data.train_ds), len(data.valid_ds)
```

```
(['Bacterial Pneumonia',
  'COVID-19',
  'Normal',
  'Other Findings',
  'TB',
  'Viral Pneumonia'],
 6,
 9021,
 2255)
```

As we can see, we are keeping 20% of the images for validation, and are using images of size 224 by 224. The 6 classes of images are visible. We have 9021 images for training and 2255 images for validation.

F. Creating the learner and fitting:

In fast.ai, every estimator is called a “learner”. For image processing tasks, we are going to use a learner made having convolutional layers, called Convolutional Neural Networks or CNNs. We are going to use the ResNet34 architecture for our model because in recent times, this architecture has given state of the art results for many popular image processing challenges. More complex and heavier architectures are available, but resnet34 is fairly suitable for our task.

```
[ ]  learn = cnn_learner(data, models.resnet34, metrics = [accuracy, error_rate])
```

The fit_one_cycle() method uses the cyclical learning rate methodology invented by Leslie Smith. In this approach, initially very high learning rates are used, and then the leaning rate is decreased when the loss gets out of control. The loss is plotted against learning rate, so we can use the optimum rate with the help of the plot.
We initially train for 4 epochs.

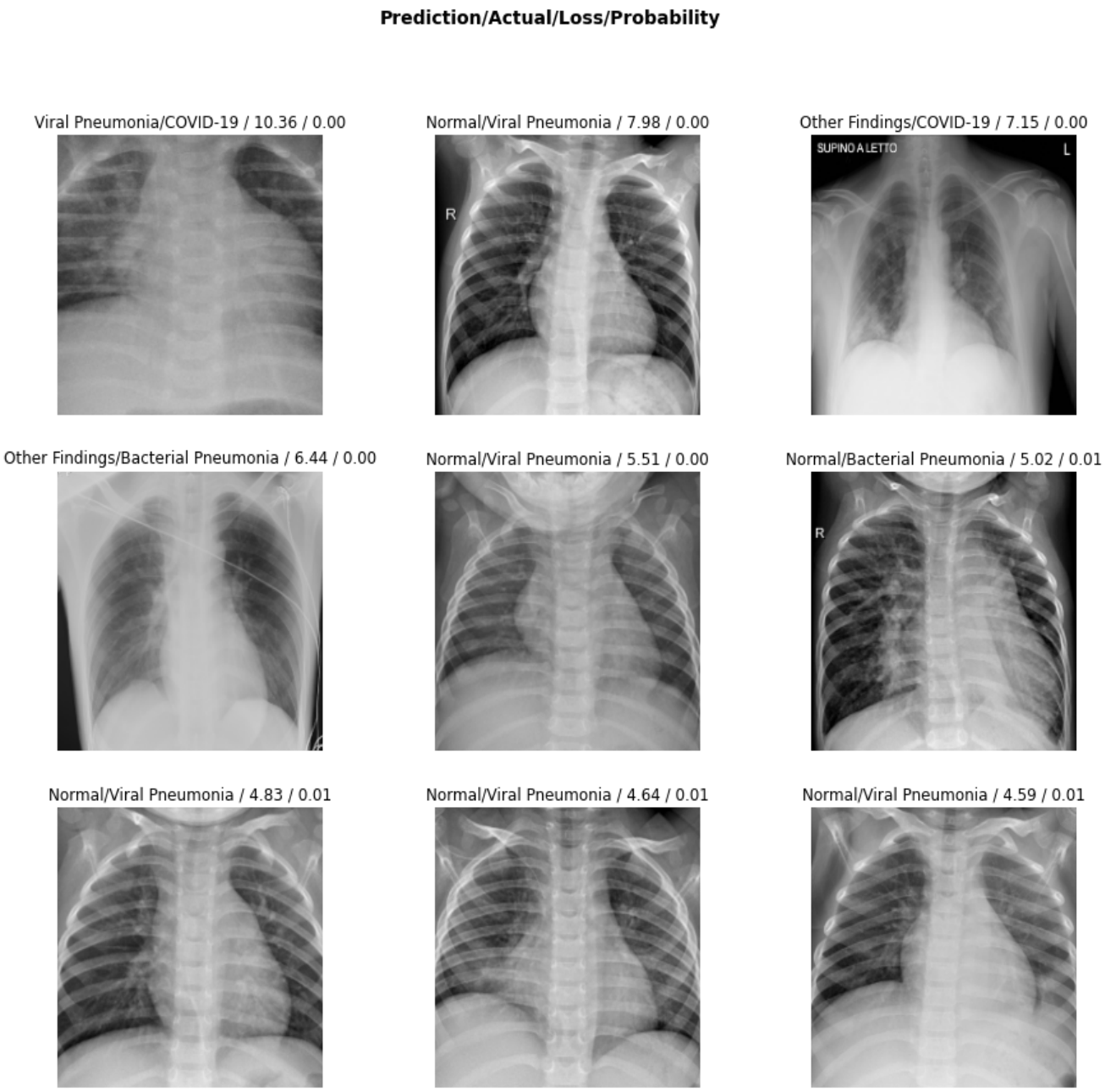
```
[ ] learn.fit_one_cycle(4)
```

epoch	train_loss	valid_loss	accuracy	error_rate	time
0	0.740564	0.447471	0.849667	0.150333	26:42
1	0.485865	0.331255	0.866962	0.133038	09:05
2	0.375516	0.339502	0.859867	0.140133	09:04
3	0.337040	0.300028	0.875388	0.124612	09:06

As we can see, the error rate has not started increasing drastically. Also, the training loss is more than the validation loss. So we are not overfitting yet.
This is a basic fit using a “cnn_learner” based on convolutional layers. Using this basic approach, we have achieved an accuracy of 87%, which is not bad.

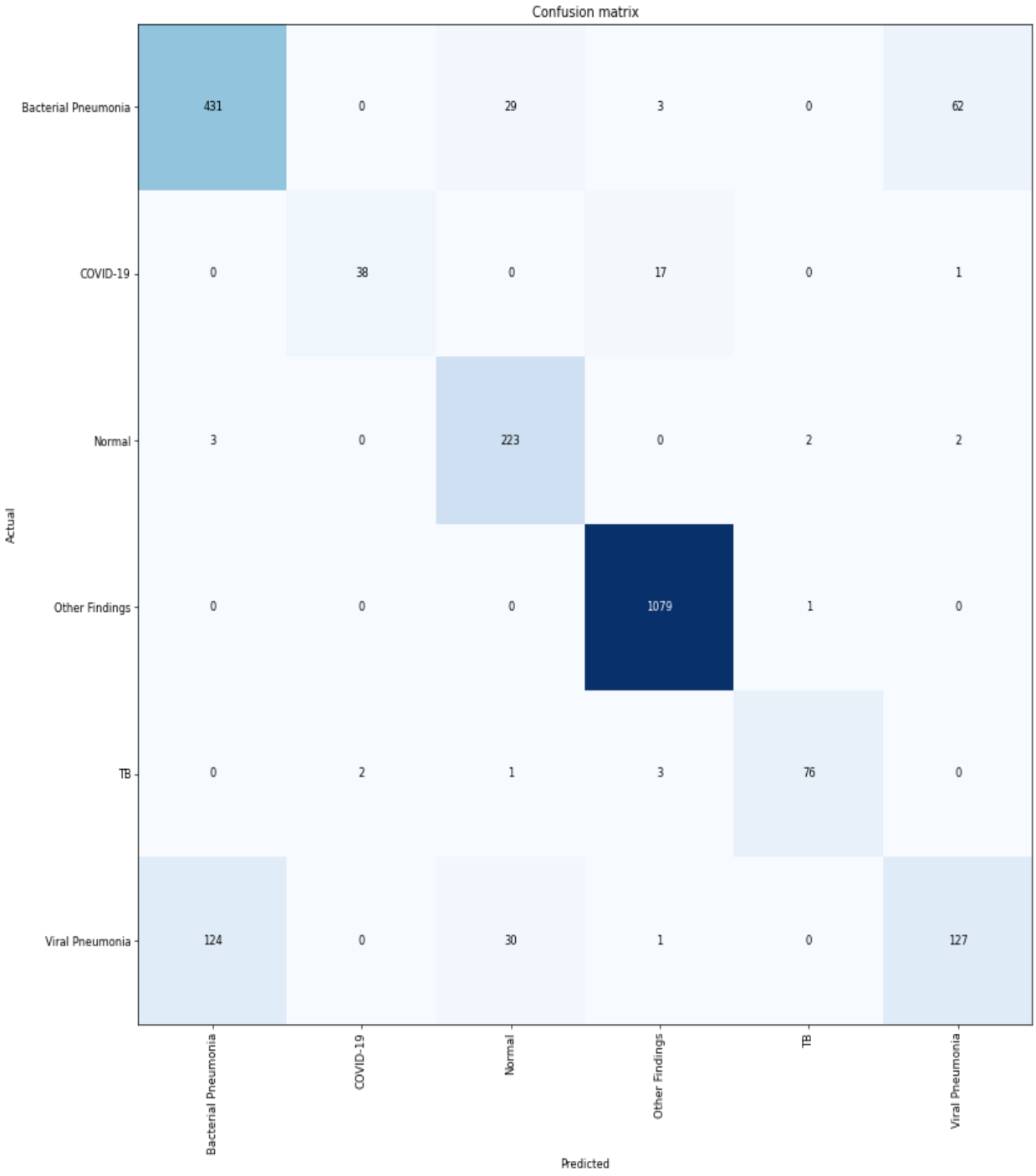
G. Model performance:

Lets now analyse the performance of our model in detail.



The above image shows the cases where our learner was the most confident about the wrong prediction. The first image is the one with highest loss, the second one next, and so on.

Let’s take a look at the confusion matrix:



As we can see, images of type “Other Findings” are classified very well. This may be due to the large number of images in that category. We will later create a more balanced dataset and train.

```
interp1.most_confused(min_val=2)
```

```
[('Viral Pneumonia', 'Bacterial Pneumonia', 124),
 ('Bacterial Pneumonia', 'Viral Pneumonia', 62),
 ('Viral Pneumonia', 'Normal', 30),
 ('Bacterial Pneumonia', 'Normal', 29),
 ('COVID-19', 'Other Findings', 17),
 ('Bacterial Pneumonia', 'Other Findings', 3),
 ('Normal', 'Bacterial Pneumonia', 3),
 ('TB', 'Other Findings', 3),
 ('Normal', 'TB', 2),
 ('Normal', 'Viral Pneumonia', 2),
 ('TB', 'COVID-19', 2)]
```

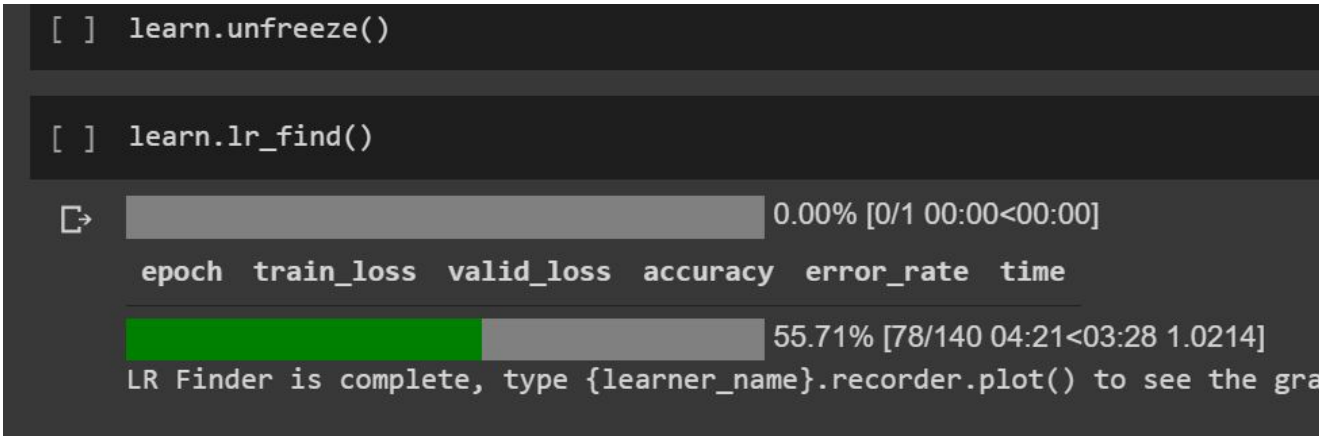
Interestingly, our learner seems to be the most confused between the 2 types of pneumonia. A possible reason for this might be similar manifestations of both diseases in the lung, so that CXR images might look similar.

H. Unfreezing and training:

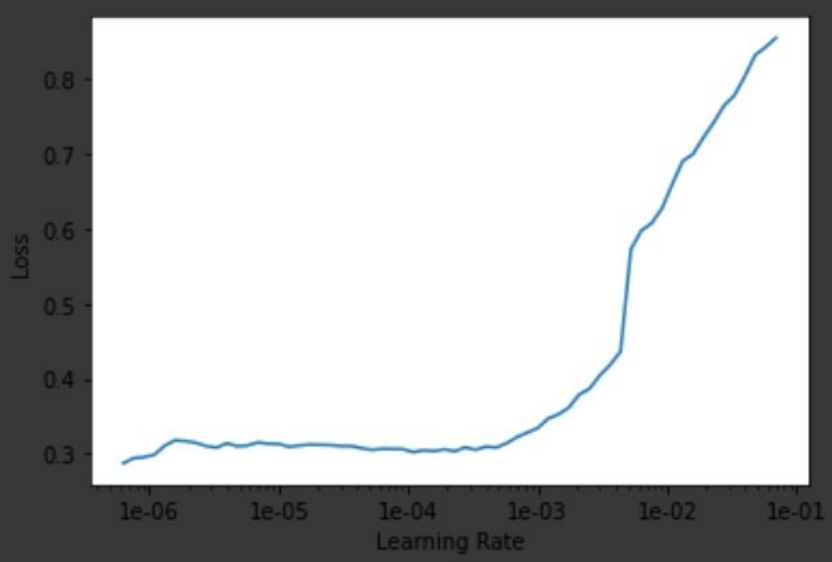
Until now, we were training a “frozen” model, which means that apart from the last few layers, most of the layers in the model were not trainable. This was done to protect the inner layers from learning unnecessary representations of the data, so that the model may generalise better.
Now, let’s unfreeze all layers and find the optimum learning rate.

```
[ ] learn.unfreeze()

[ ] learn.lr_find()
```



```
[ ]  learn.recorder.plot()
```



There is no region of steep descent in the curve.

As we can see, there is no region where the loss drops drastically. This type of a curve is typical of unfrozen models. It is a good practice to find the point where the learning rate starts to increase drastically, and take a minimum learning rate about 10 times smaller than that or lower.

```
[ ]  learn.fit_one_cycle(2, max_lr=slice(1e-5,1e-4))
```

epoch	train_loss	valid_loss	accuracy	error_rate	time
0	0.314130	0.271812	0.890466	0.109534	09:19
1	0.276294	0.260244	0.894900	0.105100	09:20

The initial training for 4 epochs took about 55 minutes, and the second stage took 20 minutes.
Thus, by training for just 6 epochs in total, and about 1 hour and 15 minutes, we have created a classification model from scratch that can classify x-ray images with close to 90% accuracy.

15. Training with augmented data:

As we saw above, the COVID-19 class seems to be severely under-represented. It has only 288 images, compared to 100s of images of other classes. Lets try some data augmentation, and see if it helps.
There are 288 COVID images. We will take every third image of this category from the original dataset, and create 5 augmented images from each original image.

After augmentation, the number of images in COVID19 category has increased.

```
We have 2530 CXR images of type Bacterial Pneumonia
We have 689 CXR images of type COVID-19
We have 1122 CXR images of type Normal
We have 5597 CXR images of type Other Findings
We have 394 CXR images of type TB
We have 1345 CXR images of type Viral Pneumonia
We have 11677 images in total.
```

If augmentation proves to be helpful, we will use this for other categories as well.
Lets now train a new “cnn_learner” on the new dataset and see the performance.

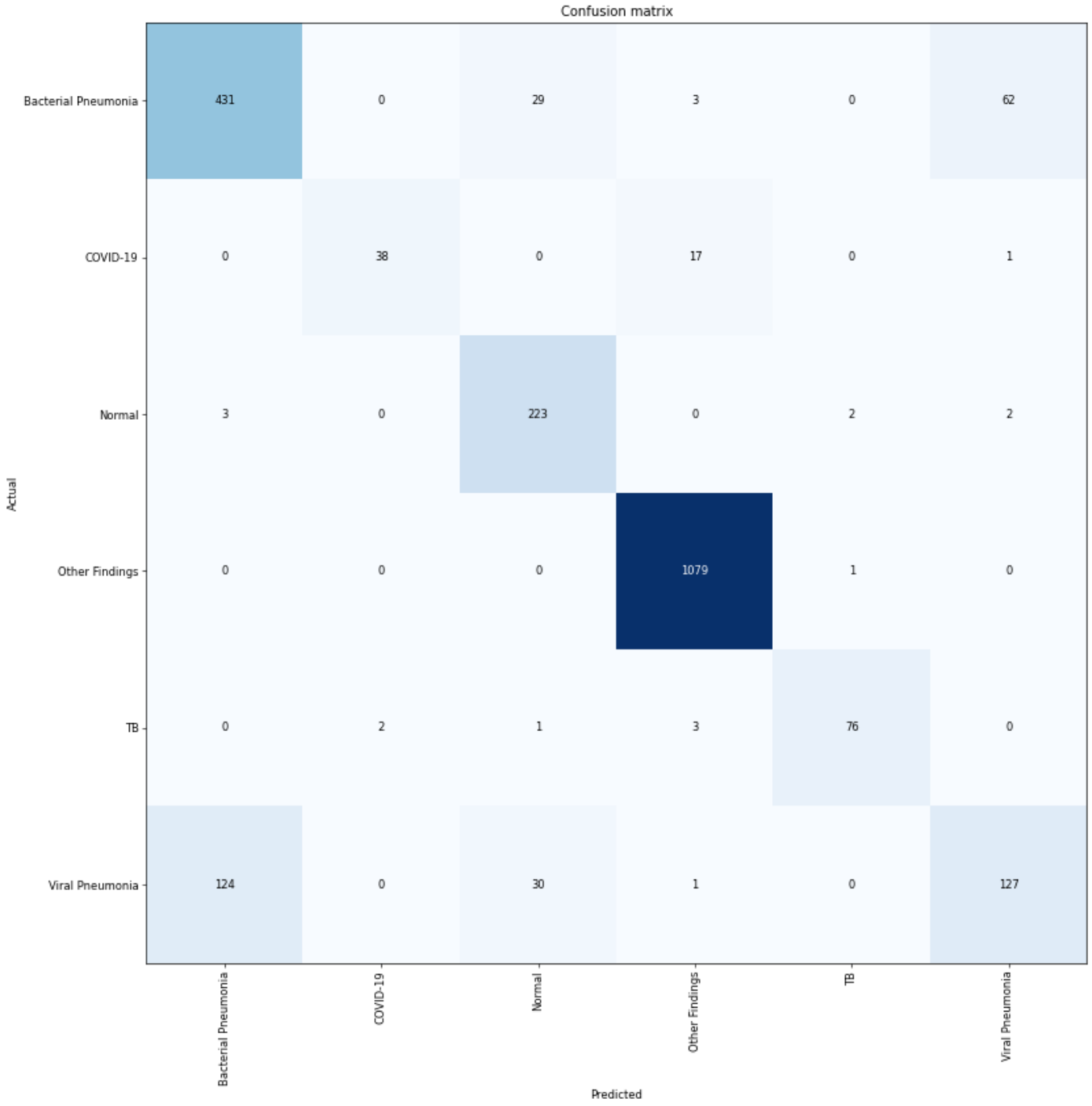
```
[ ] learn = cnn_learner(data, models.resnet34, metrics = [accuracy, error_rate])

learn.fit_one_cycle(4)   # training for 4 epochs, as above, so that we can compare accuracy
```

epoch	train_loss	valid_loss	accuracy	error_rate	time
0	0.751406	0.443958	0.837259	0.162741	20:34
1	0.472283	0.359046	0.852677	0.147323	07:38
2	0.372025	0.326270	0.866381	0.133619	07:37
3	0.334746	0.315116	0.871520	0.128480	07:36

Interestingly, we don’t see much change in accuracy. In fact, our error rate has increased from 0.125 to 0.128.

Lets see if there is any change in the confusion matrix.



Interestingly, our confusion matrix has not changed at all.

```
[ ] interp1.most_confused(min_val=2)

[('Viral Pneumonia', 'Bacterial Pneumonia', 121),
 ('Bacterial Pneumonia', 'Viral Pneumonia', 70),
 ('Bacterial Pneumonia', 'Normal', 36),
 ('Viral Pneumonia', 'Normal', 29),
 ('COVID-19', 'Other Findings', 22),
 ('TB', 'Other Findings', 6),
 ('Bacterial Pneumonia', 'Other Findings', 5),
 ('Normal', 'Viral Pneumonia', 3),
 ('Normal', 'Bacterial Pneumonia', 2),
 ('Viral Pneumonia', 'Other Findings', 2)]
```

After augmentation, more images of bacterial pneumonia are being mis-classified as viral pneumonia and normal category.
Thus augmentation does not seem to have helped. Even after unfreezing, we don't see much improvement in accuracy.
Let us unfreeze and see if accuracy has improved.

```
[ ] learn.unfreeze()

learn.fit_one_cycle(2, max_lr= 1e-2)
```

epoch	train_loss	valid_loss	accuracy	error_rate	time
0	0.332290	0.307160	0.874518	0.125482	13:36
1	0.334163	0.306714	0.873662	0.126338	08:23

After unfreezing, the accuracy has not improved much.

16. SUMMARY OF OBSERVATIONS TILL NOW:

    a. With no preprocessing and a basic model of resnet34 architecture, we achieved 89.4% accuracy in 6 epochs
    b. With data augmentation, accuracy has not improved.
    c. The learner seems to be most confused between CXRs of bacterial and viral pneumonia. These categories have the highest number of misclassifications.
    d. Unfreezing the learner seems to have helped
    e. A learning rate of 1e-5 to 1e-4 seems to have worked well for this task.

WHY HAS AUGMENTATION NOT HELPED ?

CXR images are highly specific visual representations of the human body. Every tiny detail is important, and medical diagnosis is generally based on very minute irregularities observed in the images.

This is very different from a scenario where we are doing broad classification, for example, classifying pets. A dog image will resemble a dog, even if we rotate it slightly, shift horizontally, or flip it horizontally. Also, cats, in general, look quite different from dogs, and we can see the difference with our naked eyes. If we change the background of an image (without distorting the object). it will not make much of a difference.
But chest-x-ray images contain minute details that are necessary for correct diagnosis. Even slight changes will distort the image, and the image may not resemble an x-ray at all. Using heavy augmentation for medical image analysis is not a good idea.

17. Training on a subset of images:

Our original dataset is highly imbalanced, with CXR counts ranging from about 300 to about 5000. Although neural networks are known to handle class imbalance pretty well, we should try with a more balanced subset to see if we can get any better accuracy.

For single-label multiclass classification problems, using about 500-600 images for each category should be sufficient.
I have randomly selected about 600 images for each category, and stored them in google drive for training. Here is the count of images.

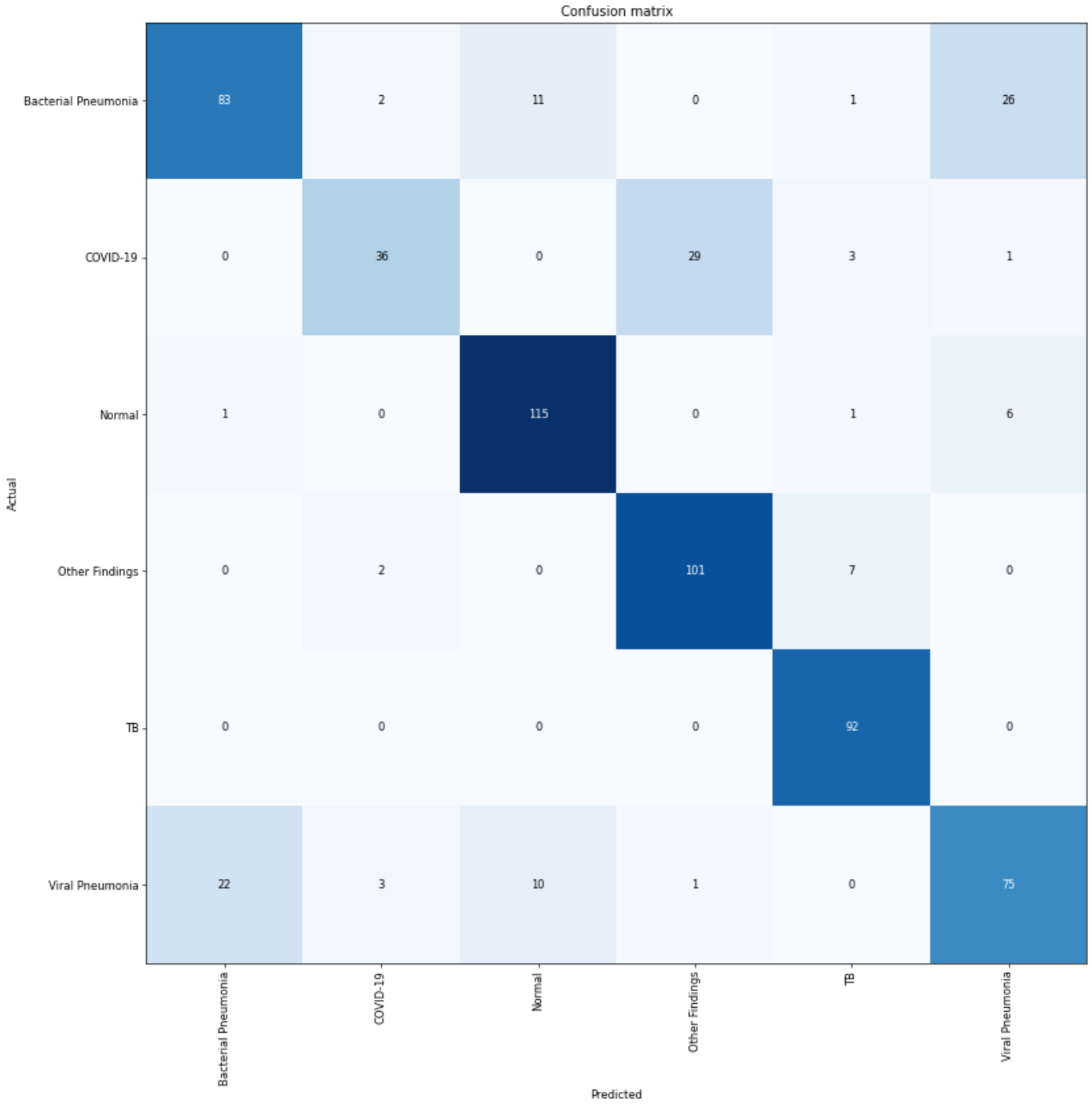
```
We have 616 CXR images of type Bacterial Pneumonia
We have 288 CXR images of type COVID-19
We have 613 CXR images of type Normal
We have 615 CXR images of type Other Findings
We have 394 CXR images of type TB
We have 616 CXR images of type Viral Pneumonia
We have 3142 images in total.
```

Lets fit for 4 cycles as usual and see the accuracy.

```
learn.fit_one_cycle(4)
```

epoch	train_loss	valid_loss	accuracy	error_rate	time
0	1.397551	0.967099	0.716560	0.283439	10:07
1	0.990396	0.662225	0.765924	0.234076	03:38
2	0.787333	0.564793	0.781847	0.218153	03:37
3	0.660789	0.533343	0.799363	0.200637	03:35

After 4 epochs, we have achieved accuracy of ~80%. The accuracy has decreased, which is expected because we have also heavily reduced the number of images. (We are now using less than a third of the images we were originally using).



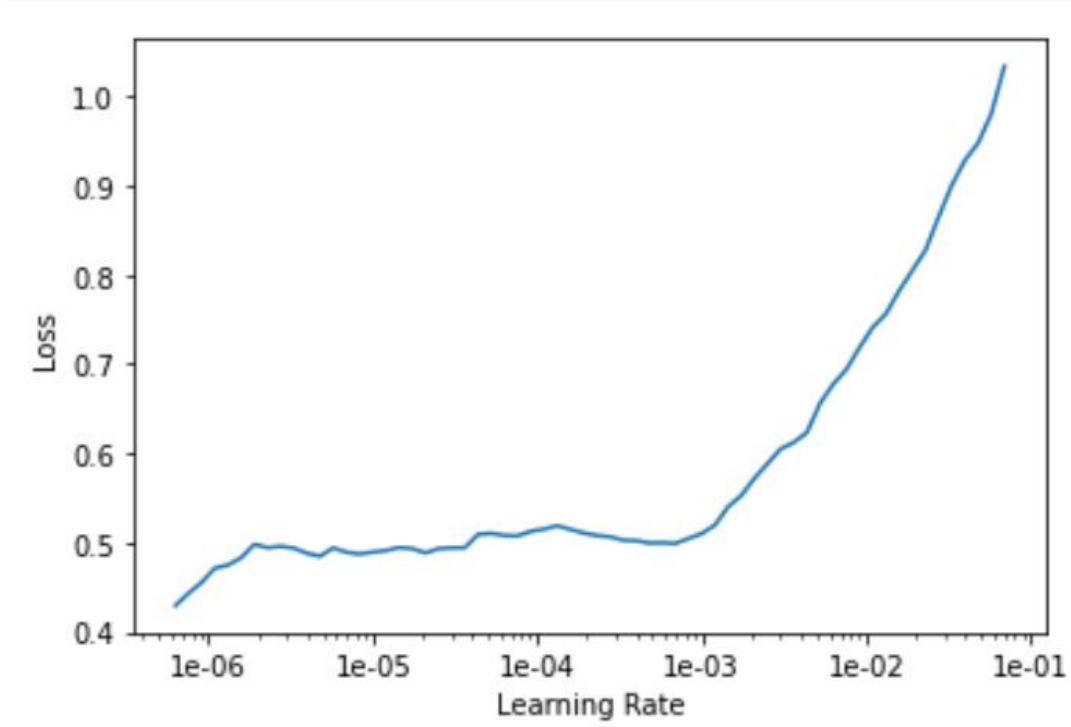
The learner seems to make the same mistakes as before. Although it is now also confused between COVID images and “Other Findings” images. Also, it seems more confused between Normal and Pneumonia images.

```
interp1.most_confused(min_val=2)
```

```
[('COVID-19', 'Other Findings', 29),
 ('Bacterial Pneumonia', 'Viral Pneumonia', 26),
 ('Viral Pneumonia', 'Bacterial Pneumonia', 22),
 ('Bacterial Pneumonia', 'Normal', 11),
 ('Viral Pneumonia', 'Normal', 10),
 ('Other Findings', 'TB', 7),
 ('Normal', 'Viral Pneumonia', 6),
 ('COVID-19', 'TB', 3),
 ('Viral Pneumonia', 'COVID-19', 3),
 ('Bacterial Pneumonia', 'COVID-19', 2),
 ('Other Findings', 'COVID-19', 2)]
```

Interestingly, our learner is now the most confused between COVID and Other images.

Let us try to find the optimum learning rate.



Again, there is nor well defined region of steep descent in loss.
Let us fit some more with optimum learning rate.

```
learn.fit_one_cycle(3, max_lr=slice(1e-5,1e-4))
```

epoch	train_loss	valid_loss	accuracy	error_rate	time
0	0.524212	0.543582	0.820064	0.179936	03:34
1	0.463744	0.466108	0.835987	0.164013	03:32
2	0.421807	0.416512	0.856688	0.143312	03:32

We have achieved an accuracy of 85% after 7 epochs of training, which completed in 32 minutes.

In the original dataset, we trained for 75 minutes with 11276 images and achieved ~90% accuracy. Here, we got just a 6% drop in relative accuracy, but our dataset size and training time is drastically reduced. For quick testing and deployment, choosing a smaller dataset seems pretty reasonable.

18. PROGRESSIVE IMAGE RESIZING

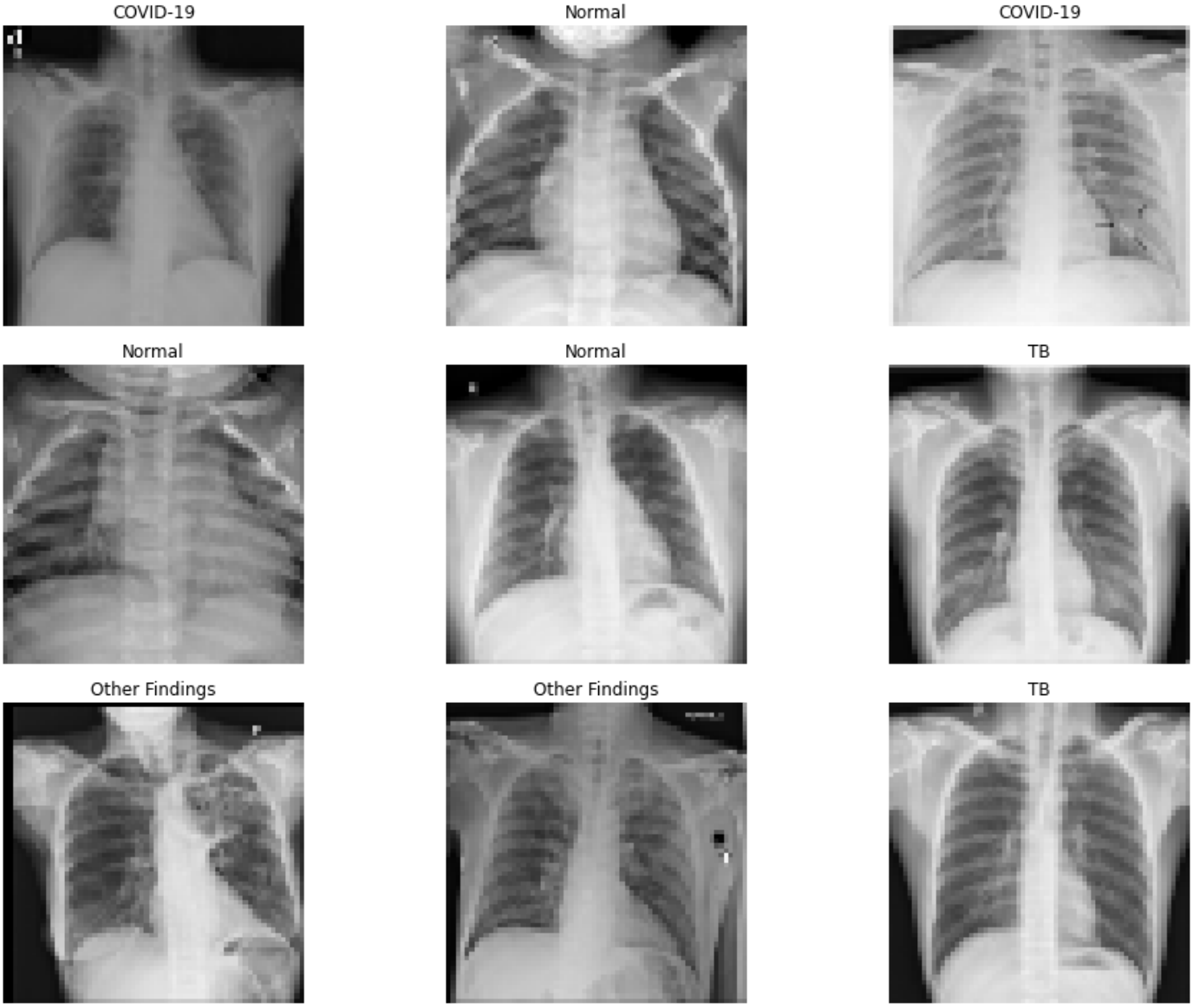
Now, let us use the final tool at our disposal, progressive image resizing. We will initially train on a very small image. Then we will use transfer learning to train the same model on bigger images, and see the performance.
(Here, we have included a few random images TB and COVID, and marked them as belonging to another category - “random subset”, to try to throw the learner off)

STAGE-1:

First, we are going to train on images of size 64 X 64.

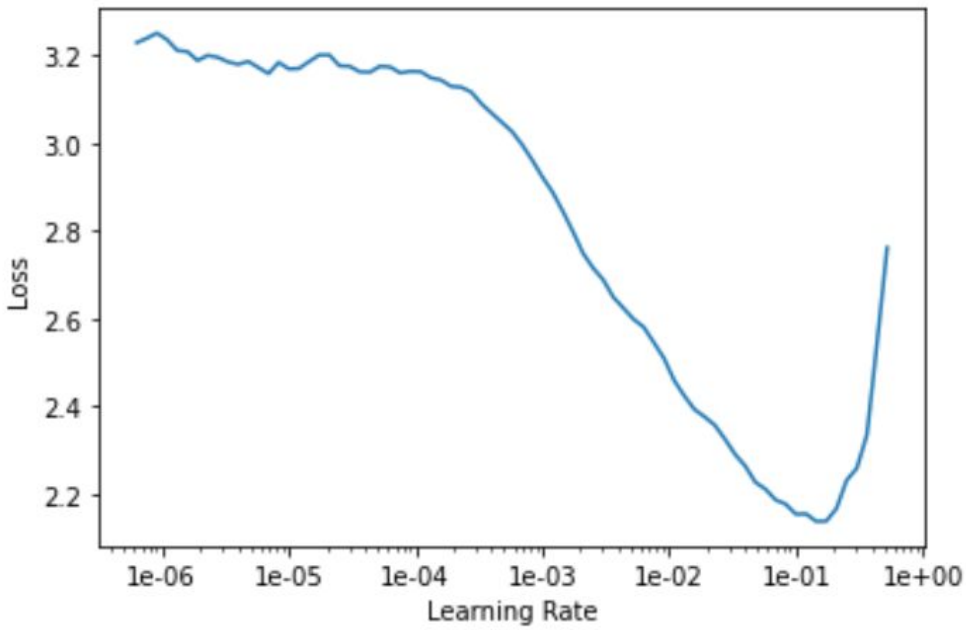
```
# using images of size 64 X 64
path = data_path
data = (ImageList.from_folder(path)
        .split_by_rand_pct(0.2)
        .label_from_folder()
        .transform(size=64) # to use images of size 64 X 64
        .databunch())
```

We have applied a transform on the images to reduce the image size. Let us take a look at the transformed images.



As expected, the images appear blurry and low-resolution.

For training this batch of images, as we are using low-resolution images, let us plot the learning rate vs loss before any training.



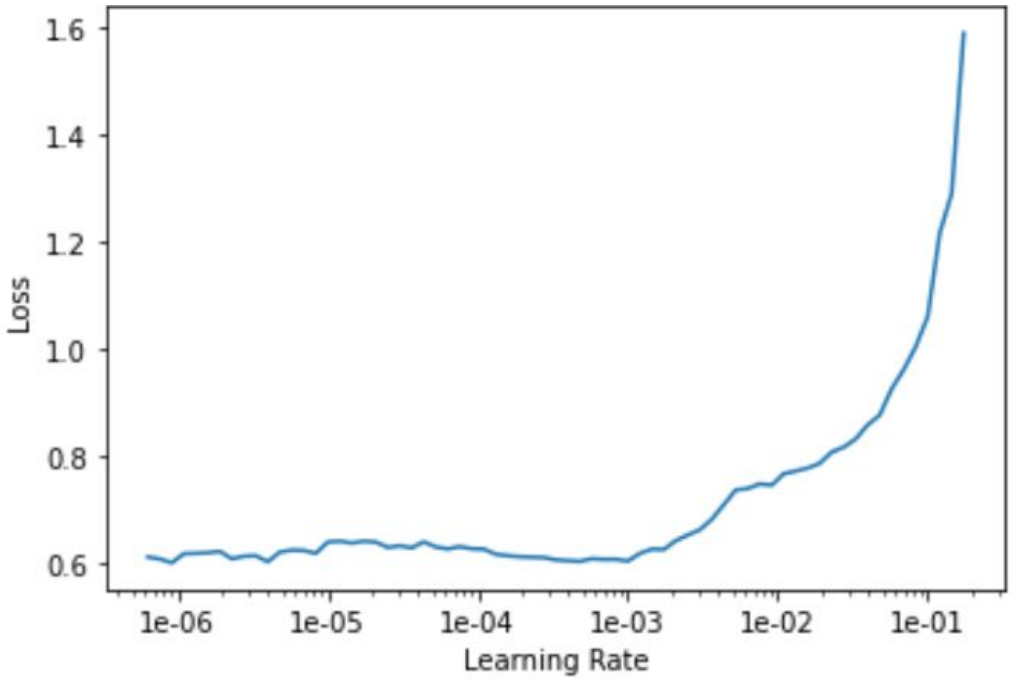
A steep descent is visible in the curve. It is a rule of thumb to use the region of the steepest descent, or a value somewhere in the middle of the descent region, as the optimum learning rate.

```
learn.fit_one_cycle(5, slice(lr))
```

epoch	train_loss	valid_loss	accuracy	time
0	2.317661	1.524154	0.523885	03:51
1	1.513407	1.022265	0.665605	02:19
2	1.112880	0.767682	0.702229	02:15
3	0.865325	0.742865	0.726115	02:12
4	0.748434	0.744711	0.730892	02:11

After 5 epochs, we are getting an accuracy of 73%. Let us unfreeze and train more with optimum learning rate.

After unfreezing:

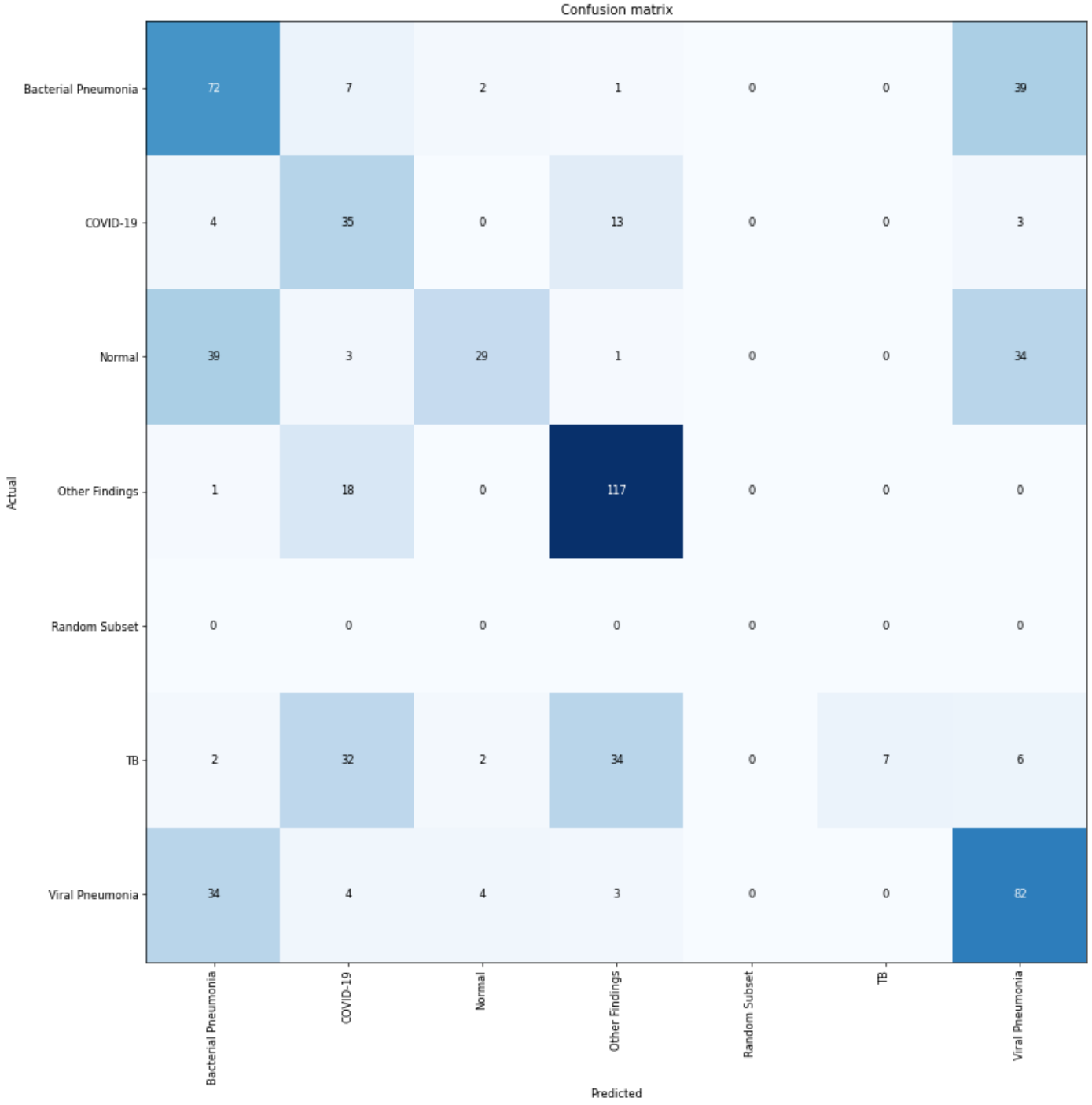


Training using optimum learning rate,

```
learn.fit_one_cycle(5, slice(5e-5, lr/2))
```

epoch	train_loss	valid_loss	accuracy	time
0	0.608370	0.606352	0.781847	02:16
1	0.482793	0.565122	0.799363	02:14
2	0.350067	0.609955	0.816879	02:13
3	0.215173	0.516967	0.839172	02:14
4	0.133340	0.502570	0.845541	02:17

Within a combined training time of 26 minutes, we are able to achieve 84% accuracy.


Confusion matrix
Bacterial Pneumonia
COVID-19
Normal
Other Findings
Random Subset
TB
Viral Pneumonia
Actual
Predicted
72 7 2 1 0 0 39
4 35 0 13 0 0 3
39 3 29 1 0 0 34
1 18 0 117 0 0 0
0 0 0 0 0 0 0
2 32 2 34 0 7 6
34 4 4 3 0 0 82

```
[('Bacterial Pneumonia', 'Viral Pneumonia', 39),
 ('Normal', 'Bacterial Pneumonia', 39),
 ('Normal', 'Viral Pneumonia', 34),
 ('TB', 'Other Findings', 34),
 ('Viral Pneumonia', 'Bacterial Pneumonia', 34),
 ('TB', 'COVID-19', 32),
 ('Other Findings', 'COVID-19', 18),
 ('COVID-19', 'Other Findings', 13),
 ('Bacterial Pneumonia', 'COVID-19', 7),
 ('TB', 'Viral Pneumonia', 6),
 ('COVID-19', 'Bacterial Pneumonia', 4),
 ('Viral Pneumonia', 'COVID-19', 4),
 ('Viral Pneumonia', 'Normal', 4),
 ('COVID-19', 'Viral Pneumonia', 3),
 ('Normal', 'COVID-19', 3),
 ('Viral Pneumonia', 'Other Findings', 3),
```

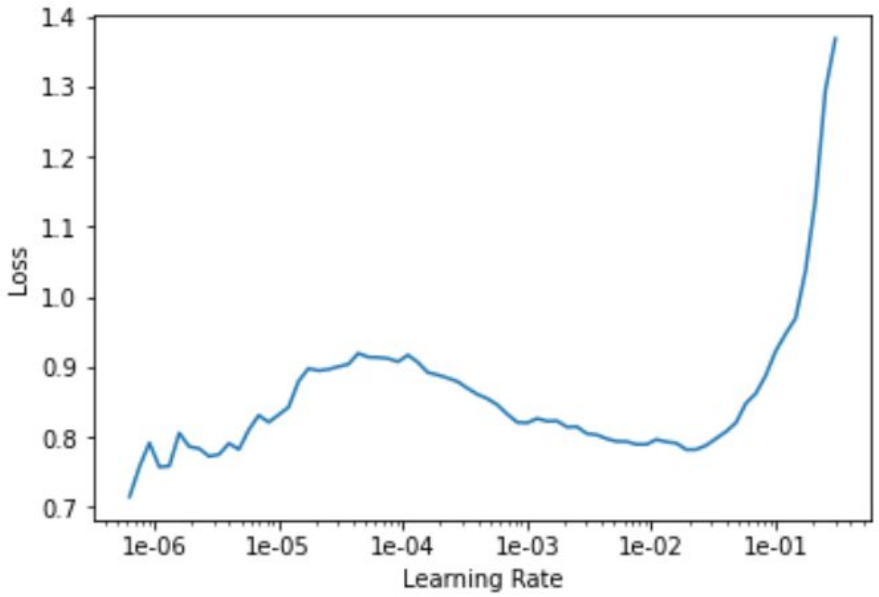
As usual, our model is confused between pneumonia and normal images. This time, many TB CXRs are also misclassified.

STAGE-2 :

Now let us train on images of size 128 X 128.
For this, we will have to create a new ImageDataBunch, using the same process as above, using transformation of “size = 128”.

NOTE: We are not discarding our ‘learner’. We will use the same learner we trained earlier, and train it on a new dataset of image size 128 X 128. We are utilising Transfer Learning.

Finding the optimum learning rate:



Fitting:

```
learn.fit_one_cycle(5, slice(lr))
```

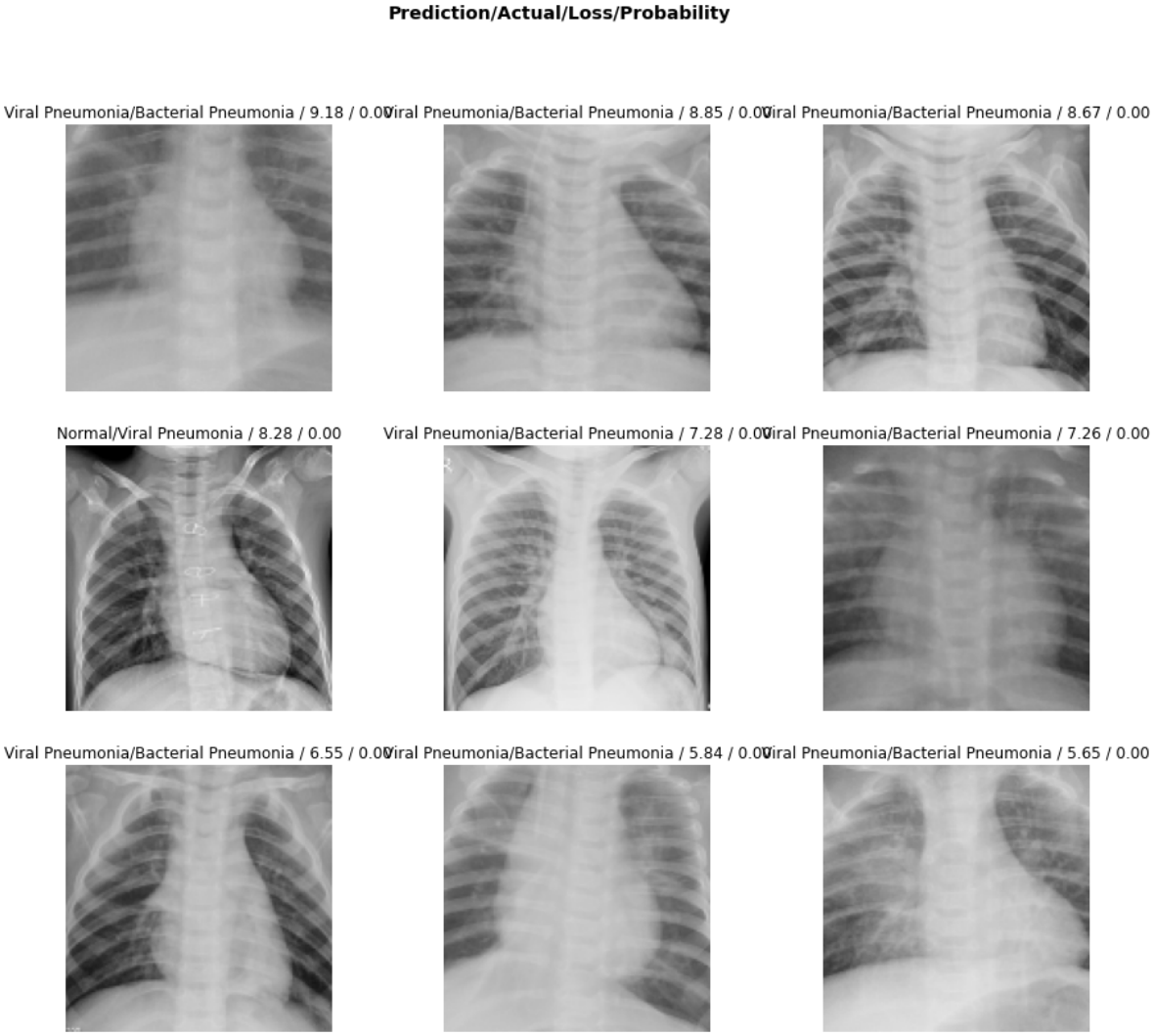
epoch	train_loss	valid_loss	accuracy	time
0	0.777712	0.560995	0.821656	02:29
1	0.628239	0.443297	0.835987	02:28
2	0.490273	0.432373	0.837580	02:24
3	0.405591	0.416886	0.837580	02:26
4	0.346249	0.416776	0.839172	02:29

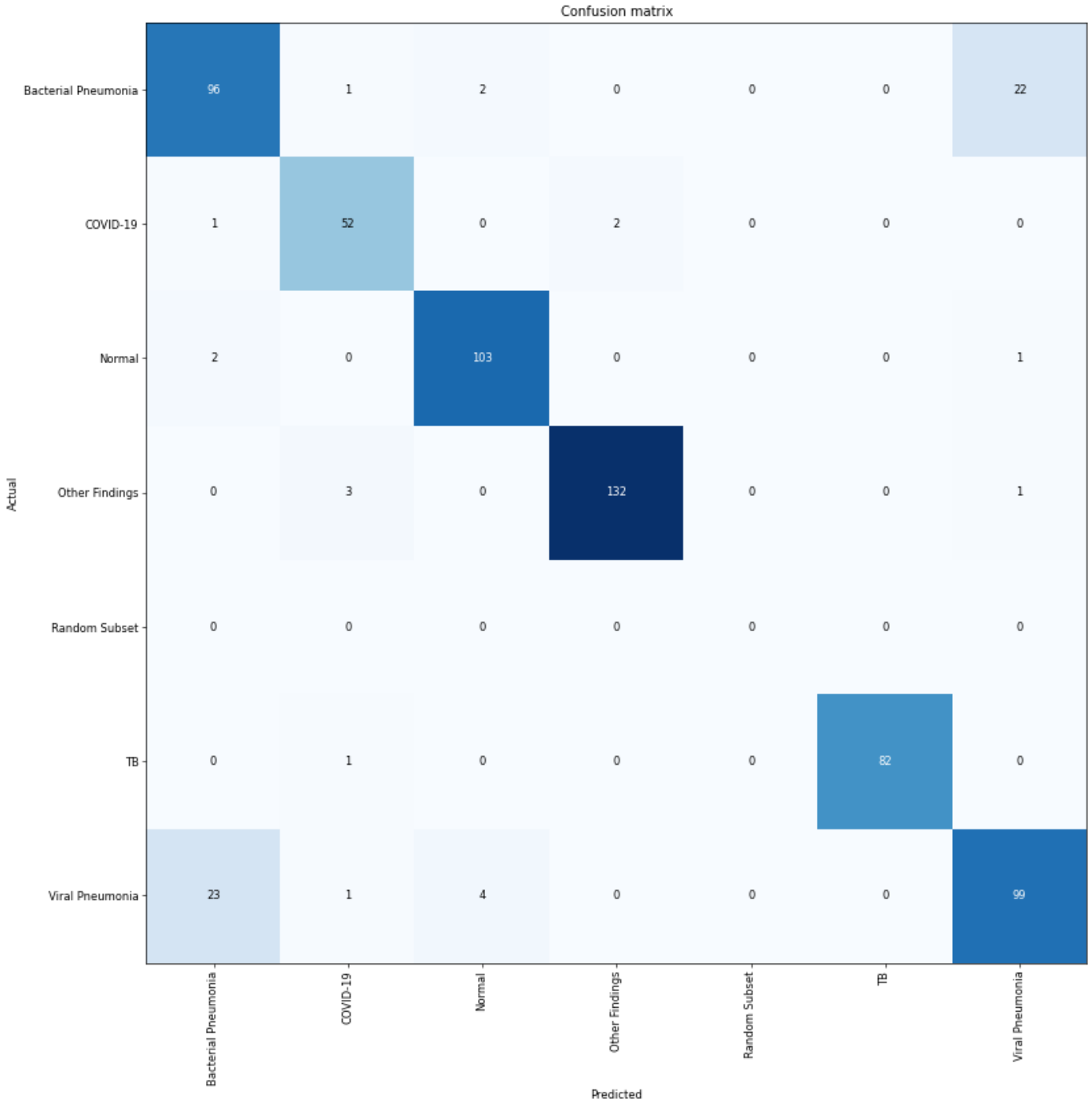
Unfreezing and fitting again:

```
learn.fit_one_cycle(5, slice(5e-5, lr/4))
```

epoch	train_loss	valid_loss	accuracy	time
0	0.317811	0.413152	0.853503	02:32
1	0.235875	0.386799	0.861465	02:33
2	0.179713	0.430592	0.869427	02:36
3	0.101160	0.366705	0.890127	02:33
4	0.056134	0.368156	0.898089	02:36

Within a total combined training time of about 55 minutes, and using low-resolution images of size 64 and 128, we are able to achieve 89.8% accuracy !!





The confusion matrix now looks much cleaner, with very few misclassifications.

At this point, we have a very good model that can classify the images of Chest-X-ray with 90% accuracy.

19. FINAL OBSERVATIONS-

    a. Training on a smaller subset of 3142 images (compared to 11276 images in the original dataset) has drastically reduced training time. We are still able to achieve accuracy of 85%.
    b. For quick model testing and employment, training with a smaller subset is a great idea.

c. Using low resolution 64X64 images, we are achieving an accuracy of 84%
d. The learner is most confused between pneumonia and normal images. With small image size, TB images are being more misclassified.
e. creating a dummy category for testing does not seem to impact the performance much
f. A learning rate of 5e-5 to 0.001 seems to be optimum for training low-resolution images
g. In the LR vs loss plot, a steep descent is visible before unfreezing the model. After unfreezing, it disappears.
h. There is a steep increase in loss when learning rate is more than 0.001, in all cases
i. Taking our current learner (trained on size-64 images) and training on size-128 images helps achieve accuracy ~90%. Transfer learning has slightly reduced training time.
j. In the last 6-7 epochs of stage-2(size-128) , the accuracy is increasing very slowly. Also, validation loss has surpassed training loss.

20. MODEL PERFORMANCE SUMMARY

Size of dataset (no. of images)	Model state (Frozen/ Unfrozen)	Image size	Learning rate (Max or range)	Training time (in minutes)	Number of epochs	Augment ation used ?	Accuracy (%)
11276	Frozen	224	0.003 (max)	54	4	No	87.5
11276	Unfrozen	224	1e-5 to 1e-4	73	6	No	89.4
11677	Frozen	224	0.003 (max)	44	4	Yes	87.1
11677	Frozen	224	1e-4 to 1e-2	67	7	Yes	90.0
11677	Unfrozen	224	0.01 (max)	88	9	Yes	87.3
3142	Frozen	224	0.003 (max)	21	4	No	79.9
3142	Unfrozen	224	1e-5 to 1e-4	33	7	No	85.6
3142	Frozen	64	0.001 (max)	12	5	No	73.0
3142	Unfrozen	64	5e-5 to 5e-4	23	10	No	84.5
3142	Frozen	128	0.001 (max)	13	5	No	83.9
3142	Unfrozen	128	5e-5 to 2.5e-4	26	10	No	89.8

21. CONCLUSION:

Neural networks are the backbone of modern artificial intelligence. In the past, carrying out Deep Learning experiments has been severely time consuming and resource intensive. Hence, the reach of neural networks and deep learning was restricted to people in large corporations with access to high-end computers, or to researchers and academicians. With the rapid advancements in storage technology and computational power, deep learning has come within the reach of the masses. As more researchers are working on developing tools and libraries with ease-of-use and simplicity in mind, building world-class AI models is now a matter of a few hours of work, and writing a few tens of lines of code. In this project, we were able to build a model that could classify very similar looking medical radiography images with 90% accuracy, without using any specialised GPUs or complex coding logic. We have used freely available online coding platforms, free-tier GPUs, and a simple library for image classification. We manually altered some parameters, tried out variations of a basic model and found out the optimum learning rate for this task. For quick model deployment and testing in the medical imaging domain, we can develop more robust and advanced models with the knowledge gained in this project. In the years to come, as technology advances, and AI education penetrates every strata of society, we will learn to use deep learning to solve problems in ways unknown.