**Pathology specimen analysis: Chest X Ray pathology analysis using CNN and Transfer learning**

A PROJECT REPORT

**Submitted by**

Priyadharshini A R - CH.EN. U4AIE20051

Venkata Ramana Murthy. Polisetty - CH.EN. U4AIE20077

Bachu Ganesh - CH.EN. U4AIE20003

Krithika S - CH.EN. U4AIE20034

Chereddy. Spandana - CH.EN. U4AIE20010

**Under the Guidance of**

**Dr. Oviya I.R**

***In partial fulfilment for the award of the degree***

***Of***

**BACHELORS OF TECHNOLOGY**

***In***

**COMPUTER SCIENCE ENGINEERING**



Amrita Vishwa Vidyapeetham

Chennai – 601 103, Tamil Nadu, India.

December 2021

# **Acknowledgment**

We offer our sincere pranams at the lotus feet of Universal guru, **MATA AMRITANANDAMAYI DEVI** who blessed us with her grace to make this a successful project.

We express our deep sense of gratitude to **Dr. P Shankar,** Principal, Amrita School of Engineering, Chennai for his kind support. We are grateful to our Course faculty **Dr.** **Oviya I.R**, for her invaluable support and guidance during the course of the project work. We are indebted to **Dr. Prasanna Kumar**, Chairperson of Department of AI, ASE, Chennai for his guidance.

We are also thankful to all other staff members of the Department of Computer Science Engineering for their valuable help, all our classmates who have always been a source of strength, for always being there and extending their valuable helps to the successful completion of this work.

**Abstract**

Chest radiographs are widely used in the detection and diagnosis of the lung diseases, such as pulmonary nodules, tuberculosis, and interstitial lung disease. Chest radiography contains a large amount of information about a patient’s health. However, correctly interpreting the information is always a major challenge for the doctor. The overlapping of the tissue structures in the chest X-ray greatly increases the complexity of the interpretation. For example, detection is challenging when the contrast between the lesion and the surrounding tissue is very low or when the lesion overlaps the ribs or large pulmonary blood vessels. Even for an experienced doctor, it is sometimes not easy to distinguish between similar lesions or to find very obscure nodules. Therefore, the examination of the lung disease in chest X-ray will cause a certain degree of missed detection. The wide application of chest X-rays and the complexity of reading them make Neural network-based computer-aided detection systems a hot research topic since the system can help doctors improving the accuracy of their detection which is exactly the objective of our project. Our project aims to develop a Machine learning model trained with CNN, Mobile Net and Inception V3 models to provide the analysis and detection of pathologies from Chest X Ray images. The Image sample dataset used for training contains about 5606 images with each image having pathologies that fall under 15 class labels namely Hernia, Pneumonia, Fibrosis, Edema, Emphysema, Cardiomegaly, Pleural Thickening Consolidation, Pneumothorax, Mass Nodule, Atelectasis, Effusion, Infiltration, No Finding. The models are trained with several hyperparameters and tested. The Evaluation of model is done by plotting confusion Matrix and ROC curves.

**Keywords: CNN, Mobile Net, Inception V3 models, Hyperparameters, Confusion matrix, ROC curve.**

**DATASET**

**SOURCE**: Kaggle

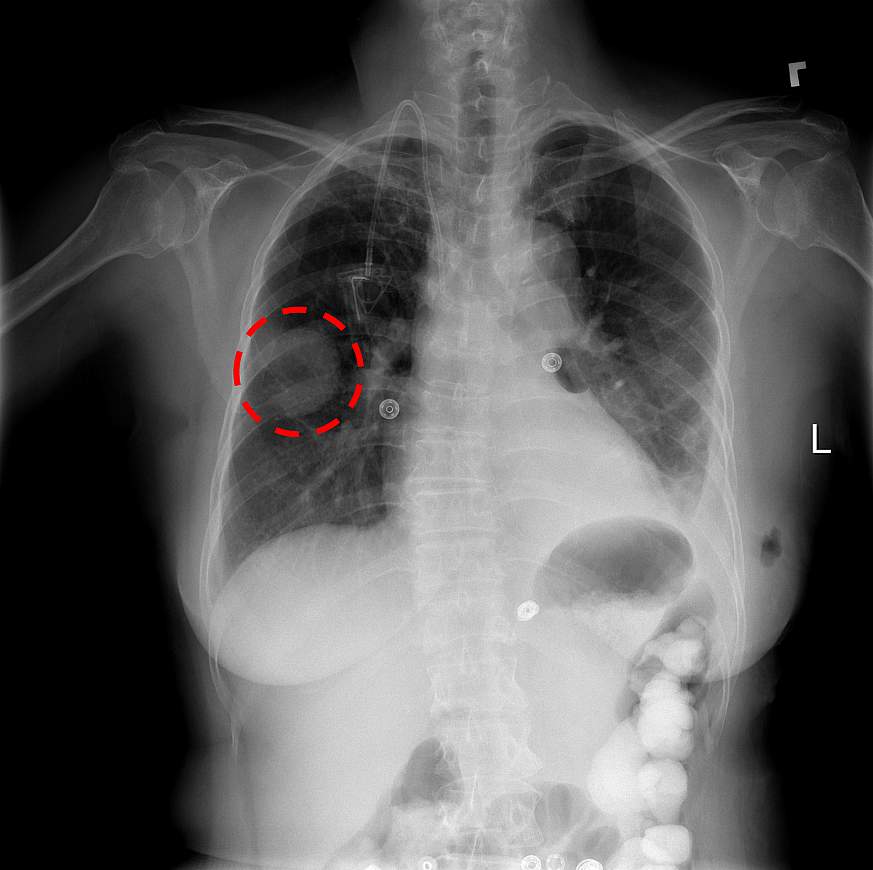
**RELEASED BY**: NIH Clinical Centre

**DESCRIPTION:**

The NIH Clinical Center released over 100,000 anonymized chest x-ray images and their corresponding data to the scientific community. The release will allow researchers across the country and around the world to freely access the datasets and increase their ability to teach computers how to detect and diagnose disease. Ultimately, this artificial intelligence mechanism can lead to clinicians making better diagnostic decisions for patients.

NIH compiled the dataset of scans from more than 30,000 patients, including many with advanced lung disease. Patients at the NIH Clinical Center, the nation’s largest hospital devoted entirely to clinical research, are partners in research and voluntarily enroll to participate in clinical trials. With patient privacy being paramount, the dataset was rigorously screened to remove all personally identifiable information before release.

Reading and diagnosing chest x-ray images may be a relatively simple task for radiologists but, in fact, it is a complex reasoning problem which often requires careful observation and knowledge of anatomical principles, physiology and pathology. Such factors increase the difficulty of developing a consistent and automated technique for reading chest X-ray images while simultaneously considering all common thoracic diseases.

By using this free dataset, the hope is that academic and research institutions across the country will be able to teach a computer to read and process extremely large amounts of scans, to confirm the results radiologists have found and potentially identify other findings that may have been overlooked.

Example: A chest x-ray identifies a lung mass.

**FEATURES OF THE DATA SET:**

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.

Dataentry2017.csv: 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

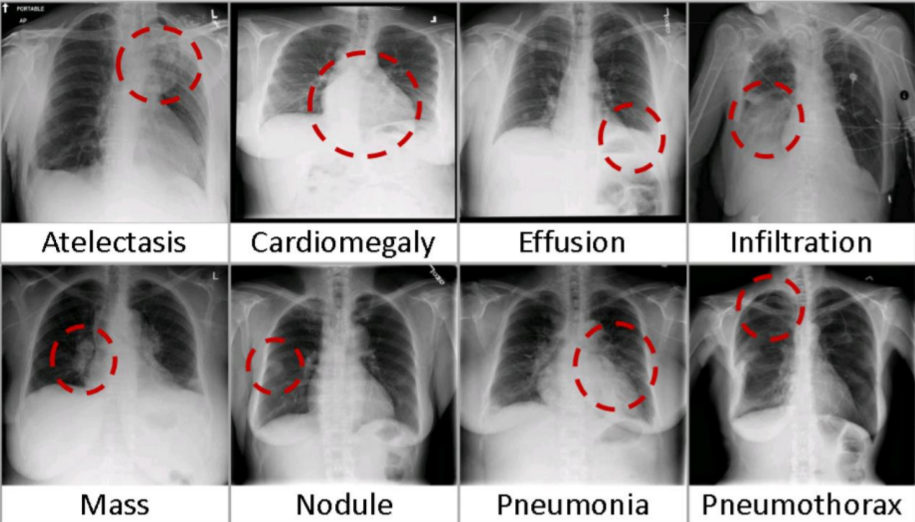


FIG 2 : Eight visual examples of common thorax diseases

Due to computation load factors, limitation in physical memory, the original dataset containing 112,120 X-ray images(45 GB) is not compatible for training our developed model. Therefore, a sample dataset which contains **(5%) of the full dataset** is used for our computation.

### A sample dataset (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
  + - * 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

### 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

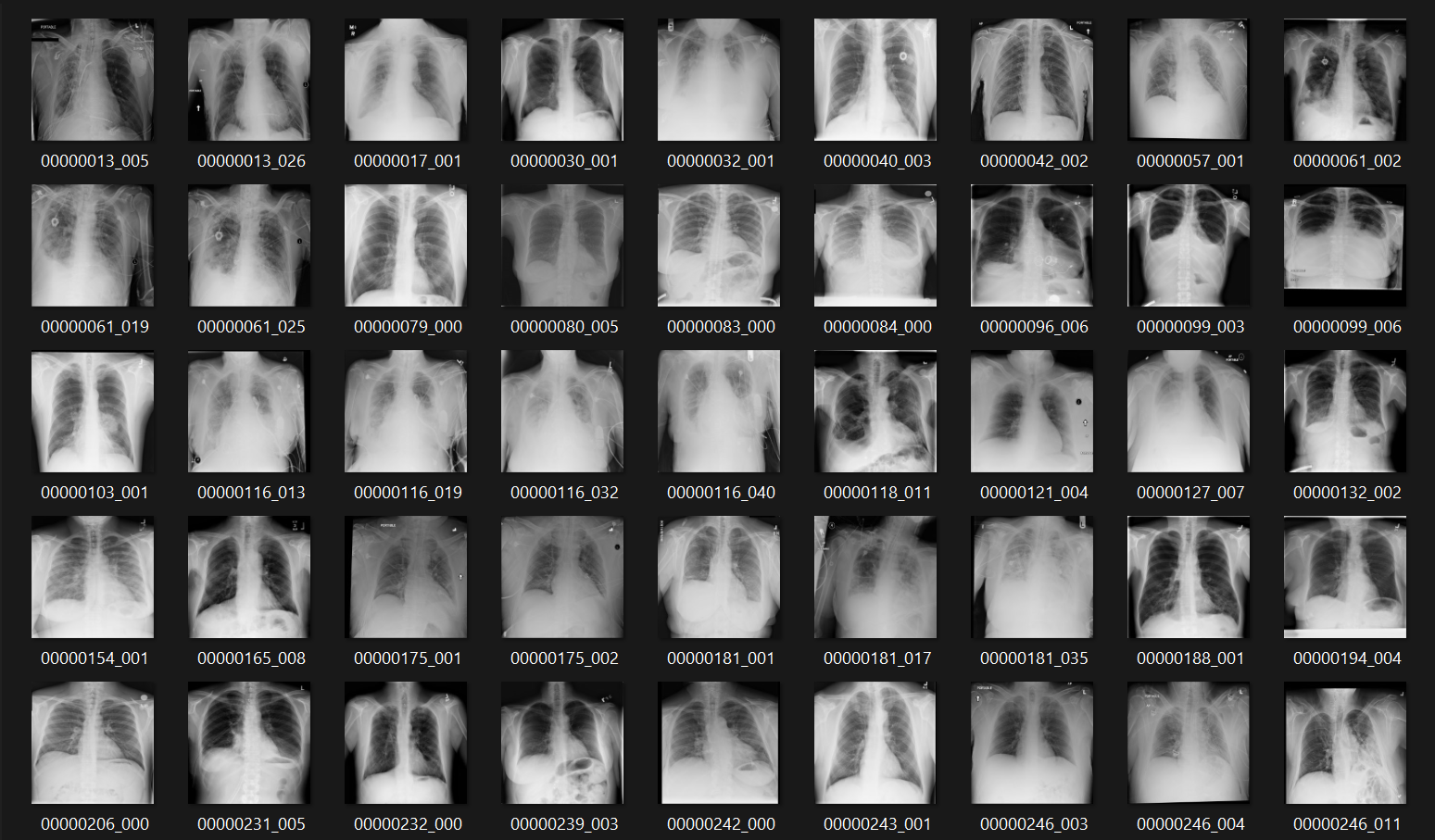


FIG 3 : Portion of our sample dataset

**Literature Review:**

Chest radiography (chest X-ray or CXR) is an economical and easy-to-use medical imaging and diagnostic technique. The technique is the most commonly used diagnostic tool in medical practice and has an important role in the diagnosis of the lung disease. Well-trained radiologists use chest X-rays to detect illnesses, such as pneumonia, tuberculosis, interstitial lung disease, and early lung cancer.

Chest X-rays is a painless, non-invasive test and is the most commonly preferred diagnostic examination to produce images of heart, lungs, airways, blood vessels and the bones of the spine and chest.

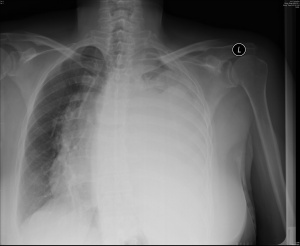
**Technique:**

An X-ray uses electromagnetic waves and ionizing radiation to create pictures of the inside of your body. The procedures involves positioning the body between the machine that produces the X-rays and a plate that that creates the image digitally or with X-ray film.

To obtain the front view:

1. [](https://www.physio-pedia.com/File:Chest_X-ray_2346.jpg)Posterior - Anterior (PA). This is the most common and preferred type of chest X-Ray. Posterior - anterior refers to the direction of the X-Ray beam travel. i e. X-Ray beams hit the posterior part of the chest before the anterior part. To obtain the image, the patient is asked to stand with their chest against the film, to hold their arms up or to the sides and roll their shoulders forward. The X-ray technician may then ask the patient to take few deep breaths and hold it for a couple of seconds. This techniques of holding the breath generally helps to get a clear picture of the heart and lungs on the image.

PA Chest X-Ray

1. [](https://www.physio-pedia.com/File:AP_Chest_X-Ray.jpg)Anterior - posterior (AP): This type of chest X-Ray is generally less preferred because the image of the heart and mediastinum is less clear and focused in this projection. To obtain AP image, the patient is asked to stand with their back against the film. If the patient is unable to stand, an AP image can also be taken with the patient sitting or supine on the bed.

AP Chest X-Ray

1. Decubitus: This type of X-Ray shows a frontal view of the chest when the patient is lying on their side(s). Decubitus X-Ray can be performed to assess the air inside the lung, the presence of free fluid, or airway obstruction.

2. To obtain the side-view (lateral): A lateral image is usually taken to complement the frontal view. A lateral image is useful to localise a lesion since, together with a frontal image, it allows a 3-dimensional analysis. To obtain a lateral image, the patient is asked to turn and place one shoulder on the plate and raise their hands over their head. The technician may again ask the patient to take a deep breath and hold it.

[](https://www.physio-pedia.com/File:472px-Chest_x-ray_-_lateral_view.jpg)

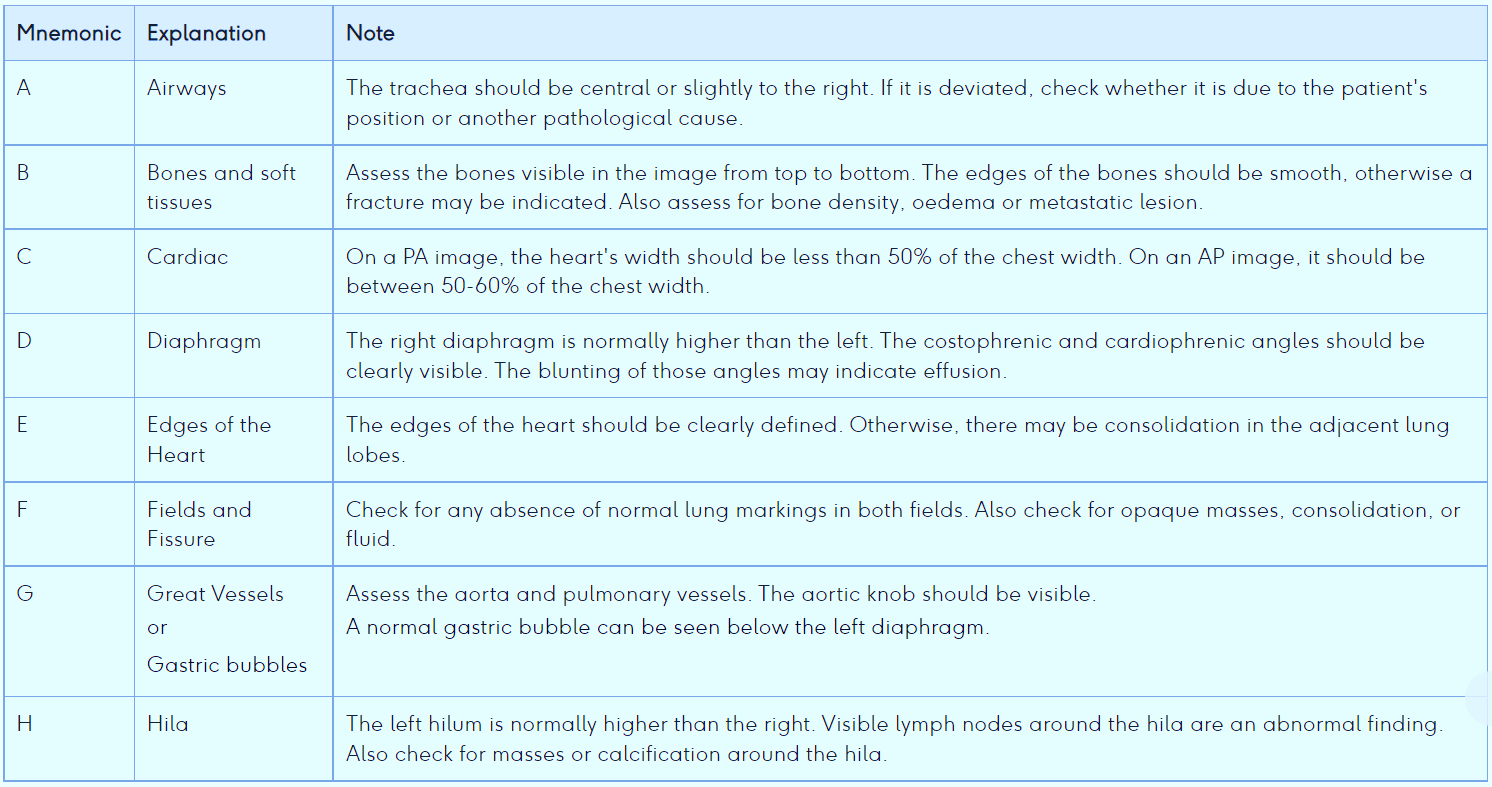
Lateral Chest X-Ray

Reading the chest X-ray systematically reduces the chance of missed diagnosis. There is no one recommended analysis methodology; some may choose to read a chest X-Ray in an anatomical order and some may choose to use a mnemonic. However, everyone should begin analysing an X-Ray by checking the following details:

* Patient's full name and date of birth
* Date and time of the X-Ray taken. There could be a few images taken on the same day.
* Image Annotation
* Right (R) or Left (left side)
* Image projection: PA (posterior-anterior) or AP (anterior-posterior) or lateral
* Patient's position. Check whether the patient is upright, semi-erect, or supine when the image was taken.
* Image Quality (R.I.P)
* R - Rotation. Check whether the patient's position is rotated. The patient is NOT rotated if the spinous processes are equidistant from the medial ends of the clavicles.
* I - Inspiration. Check whether the 5 - 7 anterior ribs intersect the diaphragm in the mid-clavicular line. If the intersection happens in anterior ribs 8 or more, then the lungs are hyperinflated.
* P - Penetration. Check whether the spine and the spaces between the ribs can be seen through the heart; which indicates that the X-ray exposure is adequate.
* Foreign objects such as lines, tubes, pacemaker, drains, etc. Foreign objects can also be non-medical such as bullets, glass, etc.

**ASSESSMENT OF X RAY:**

The most commonly used mnemonic approach to assess the details of Chest X-Ray is the following:

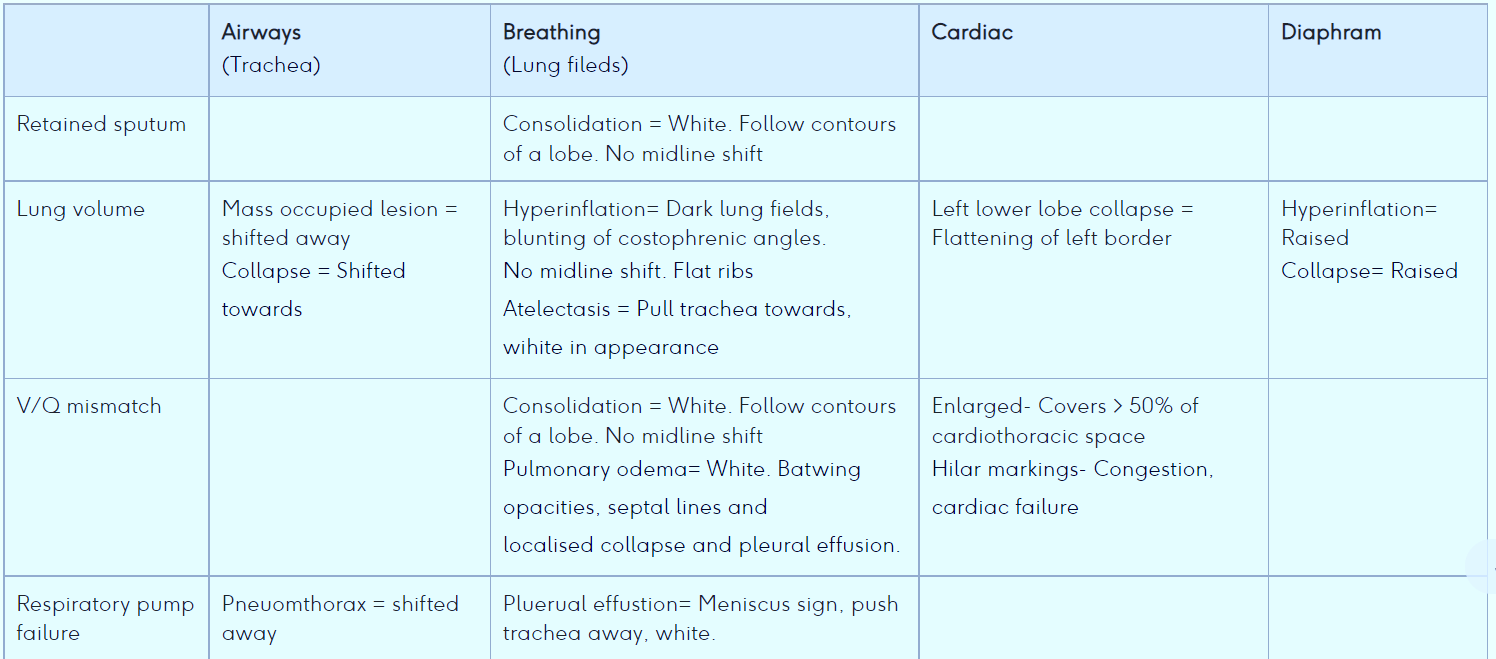
****

**Pathology Detection In X Rays:**

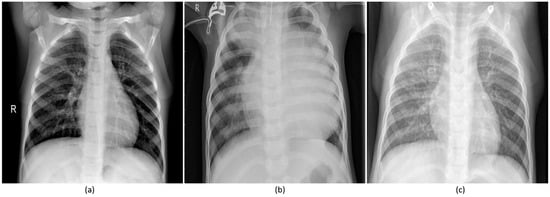
In chest radiographs, there are three main types of anomalies: texture abnormalities, which are characterized by diffuse changes in the appearance and structure of the area, such as interstitial lesions; focal abnormalities, which are manifested as isolated changes in density, such as pulmonary nodules; and abnormal shape, in which disease processes change the outline of the normal anatomy, such as cardiomegaly. Sometimes, the texture and shape of the chest changes at the same time as a certain disease, such as tuberculosis. The section describes common abnormalities in chest radiographs mainly caused by pulmonary nodules, tuberculosis, interstitial lesions, cardiomegaly, etc.

Atelectasis= Collapse, alveoli filled with fluid or collapse. Common post-surgery.

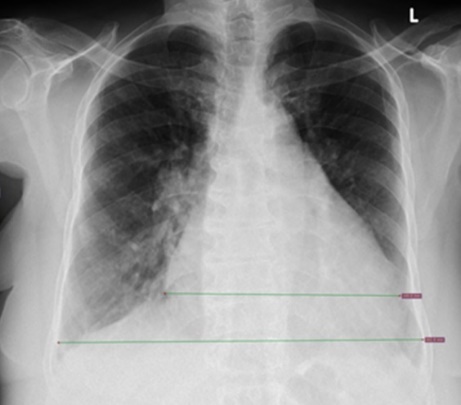
**X RAY PATHOLOGY DESCRIPTION OF EACH CLASS OF DISEASE:**

****

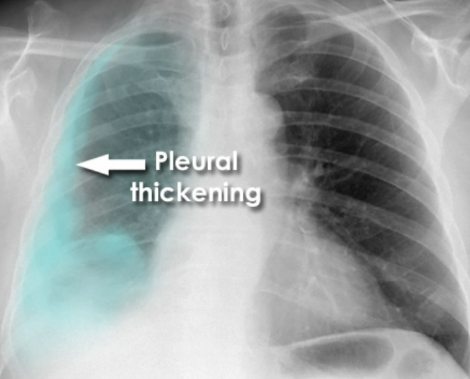
1. pneumonia



Pediatric CXRs: (**a**) Normal CXR showing clear lungs with no abnormal opacification; (**b**) Bacterial pneumonia exhibiting focal lobar consolidation in the right upper lobe; (**c**) Viral pneumonia manifesting with diffuse interstitial patterns in both lungs.

1. cardiomegaly.

An increase in the transverse cardiac diameter by 1.5cm on two consecutive chest radiographs taken at short intervals is considered abnormal, and a sign of cardiomegaly.

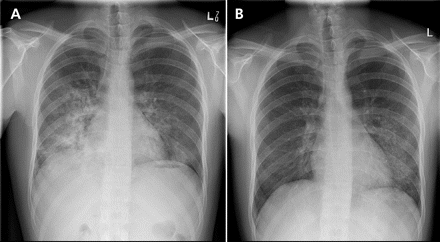
1. pleural thickening

* Lobulated peripheral shadowing on the right
* Loss of right lung volume
* Shadowing over the whole right lung due to circumferential pleural thickening.



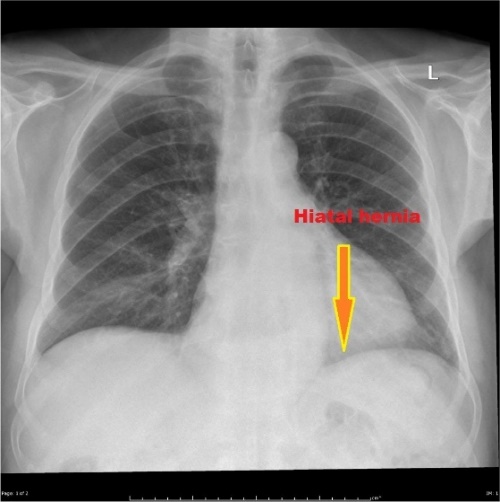
1. Nodule

Discrete, well-marginated, rounded opacity less than or equal to 3 cm in diameter that is completely surrounded by lung parenchyma, does not touch the hilum or mediastinum,

1. pulmonary infiltrates

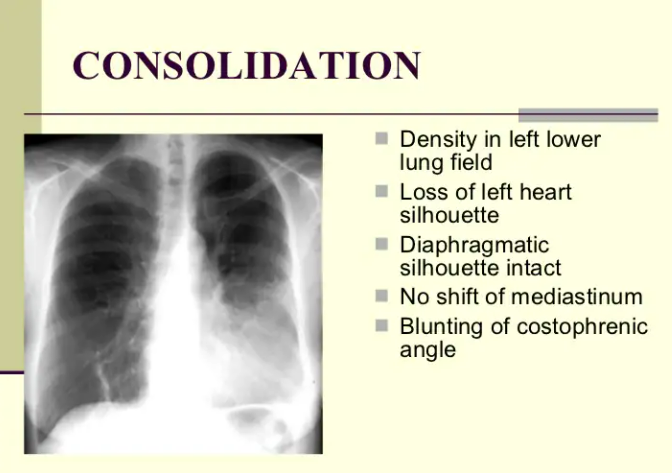
Chest radiographs taken (A) 2 weeks before evaluation showing pulmonary infiltrates in the lower regions of both lungs, and (B) during evaluation showing improvement in pulmonary infiltrates on the right side and worsening of infiltrates on the left side.

1. Hernia

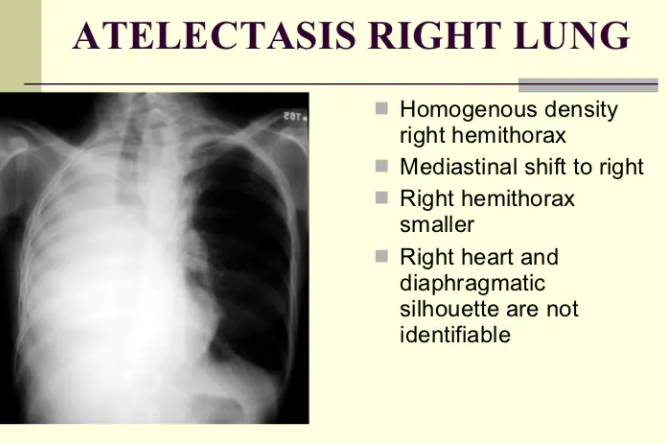


CT/MRI: supraclavicular protrusion of lung posterior to the subclavian vessels. Enlargement with Valsalva manoeuvre

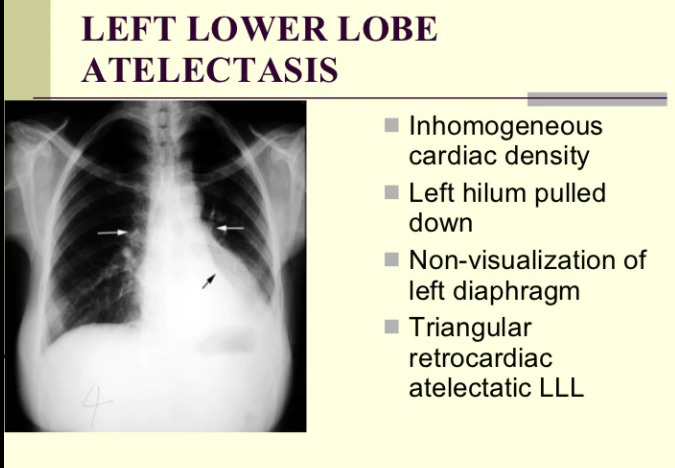
GE junction in normal position, fundus herniates into thorax

**CONSOLIDATION:**

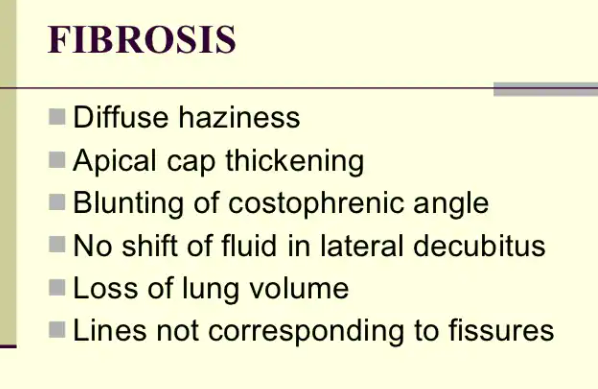
* Density in left lower lung field
* Loss of left heart silhouette
* Diaphragmatic silhouette intact
* No shift of mediastinum
* Blunting of costophrenic angle

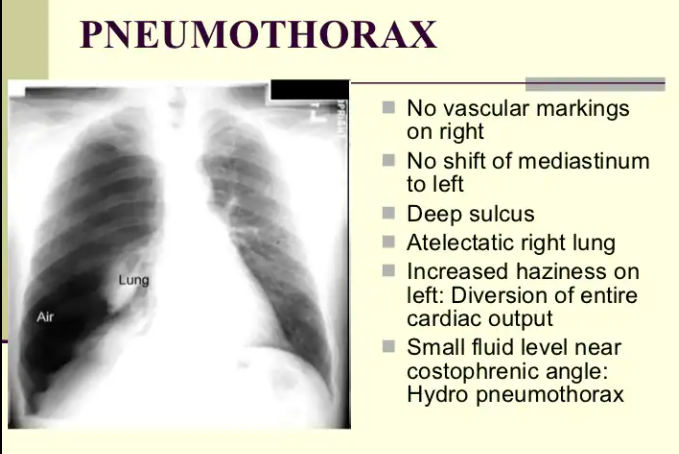
 **ATELECTASIS RIGHT LUNG:**

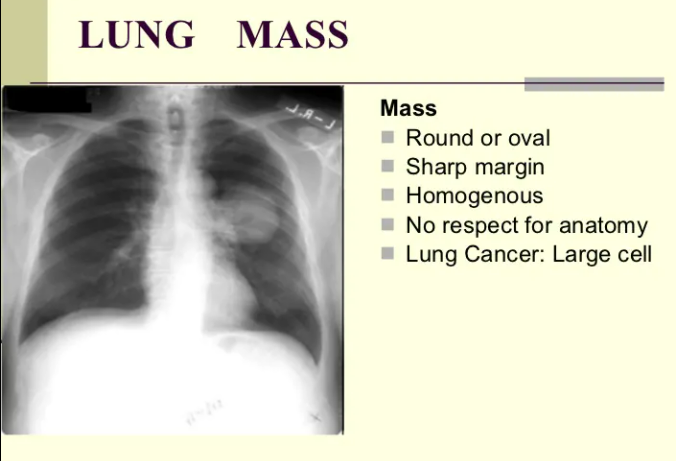
* Homogenous density right hemithorax
* Mediastinal shift to right
* Right hemithorax smaller
* Right heart and diaphragmatic silhouette are not identifiable

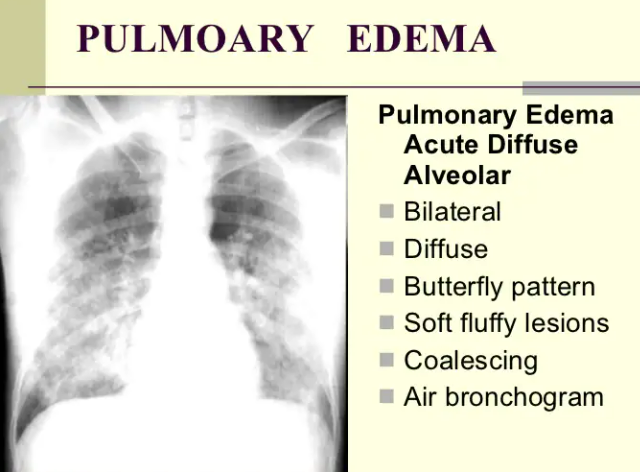


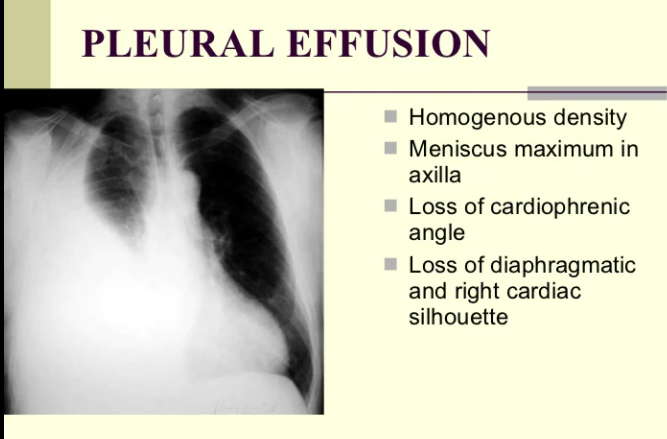
**Left Lower Lobe Atelectasis:**

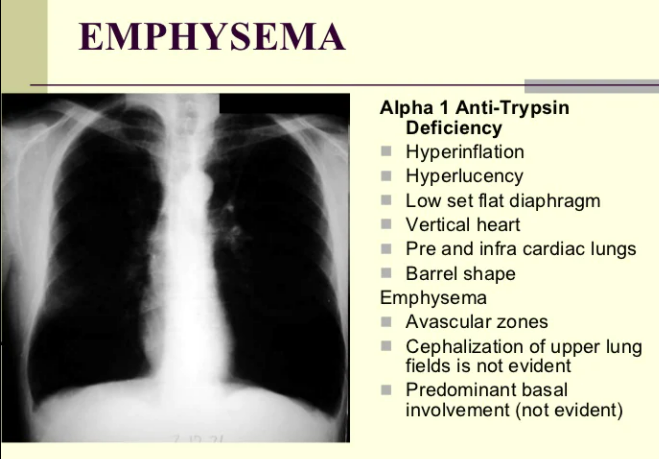












**DEVELOPING OUR MACHINE LEARNING MODEL:**

**PRE-PROCESSING**

1) **Importing dataset**

The data set ‘Data\_Entry\_2017.csv' is file containing Class labels and patient data for the entire dataset. The data set is uploaded.

2) **Label Encoding**

In machine learning, we usually deal with datasets that contain multiple labels in one or more than one column. These labels can be in the form of words or numbers. To make the data understandable or in human-readable form, the training data is often labelled in words.

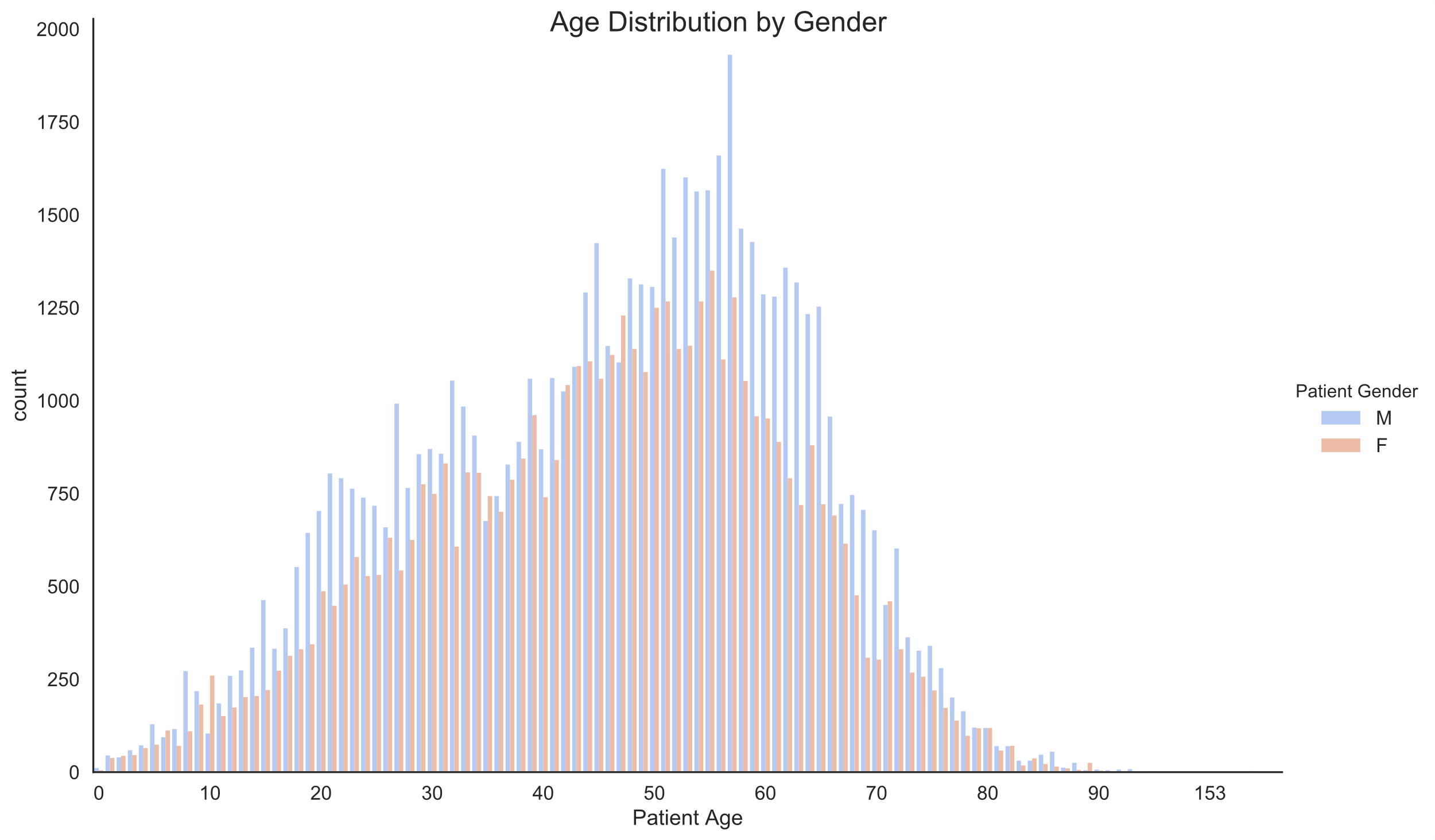
Label Encoding refers to converting the labels into a numeric form so as to convert them into the machine-readable form. Machine learning algorithms can then decide in a better way how those labels must be operated. It is an important pre-processing step for the structured dataset in supervised learning.

In our dataset there are lots of class labels. We know though that these are combinations of 15 labels(diseases) from Kaggle's data dictionary. Each label is put in its own column. The 15 class labels are made as new row names and the presence of each 15 classes in any of the column is marked by 1 else with 0.

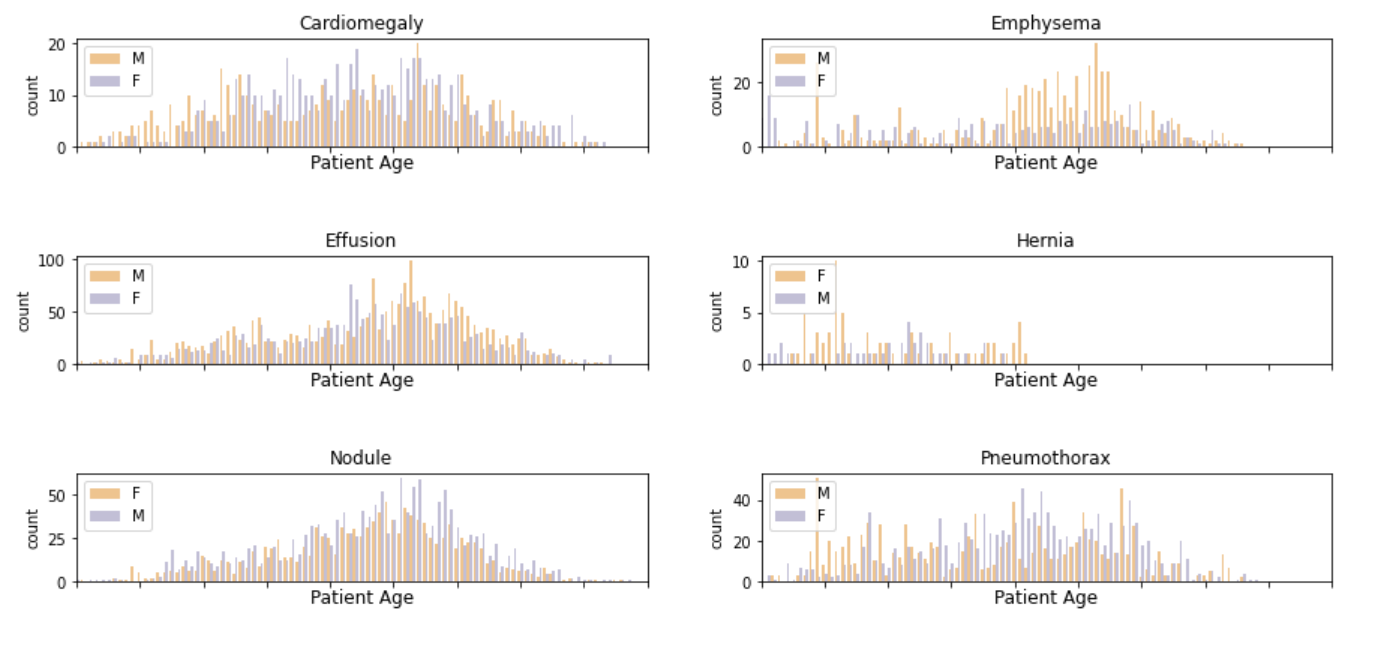
3)**Visualization of Dataset**

To extract the required information from the different visuals we create, it is quintessential that we use the correct representation based on the type of data and the questions that we are trying to understand.

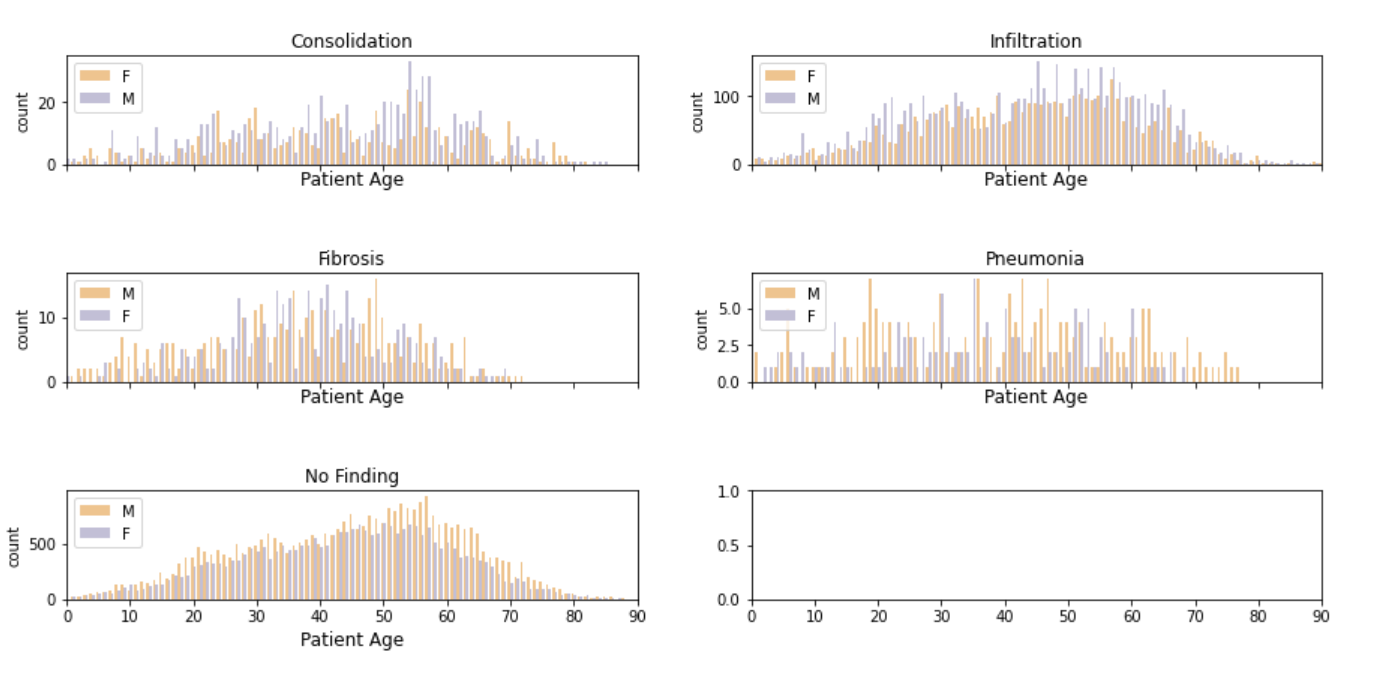
The distribution of male and female by their age and number is visualized by a plot



A distribution of patient age and their count for each 15 different cardiac pathology is visualized







4)**Image Preprocessing**

We have done the following Image pre-processing for our sample dataset using opencv and numpy

* 1. Grayscaling
  2. Resizing
  3. Normalizing

Grayscaling is the process of converting an image from other color spaces e.g. RGB, CMYK, HSV, etc. to shades of gray. It varies between complete black and complete white.

The following are the reasons why we grayscaled the images

Dimension reduction: For example, In RGB images there are three color channels and has three dimensions while grayscale images are single-dimensional.

Reduces model complexity: Consider training neural article on RGB images of 10x10x3 pixel. The input layer will have 300 input nodes. On the other hand, the same neural network will need only 100 input nodes for grayscale images.

The dimension of the image is resized from 256 x 256 to 128 x 128

Data normalization is an important step which ensures that each input parameter (pixel, in this case) has a similar data distribution. This makes convergence faster while training the network. Data normalization is done by subtracting the mean from each pixel and then dividing the result by the standard deviation. The distribution of such data would resemble a Gaussian curve centred at zero. For image inputs we need the pixel numbers to be positive, so we might choose to scale the normalized data in the range [0,1] or [0, 255].

**CNN MODEL**

CNN is combination of Convolutional Layers and Neural Network.

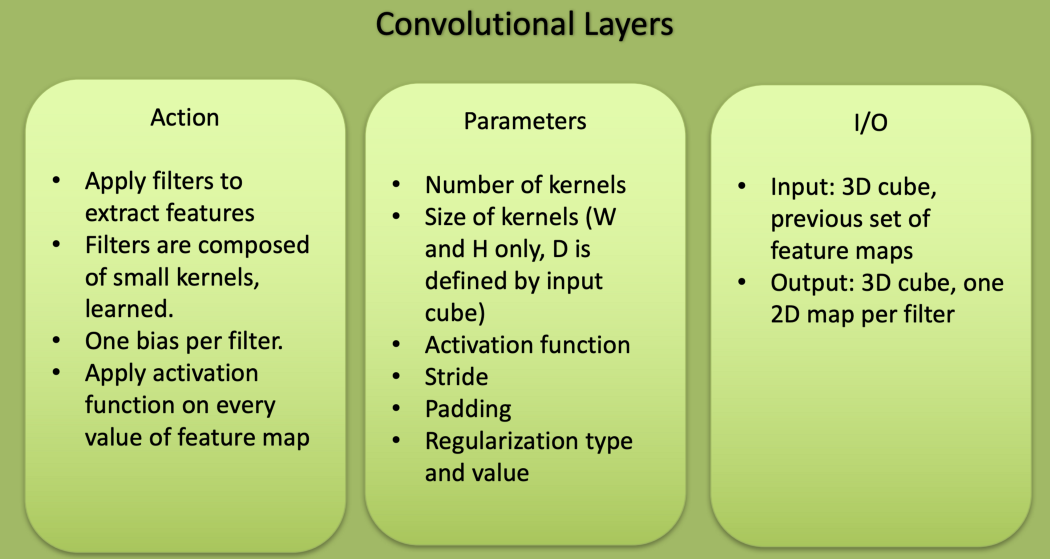
Basically any *Neural Network*which is used for image processing, consist of following layers -Input layer, Convolutional Layer, Pooling Layer, Dense Layer.

Each of these layers has different parameters that can be optimized and performs a different task on the input data.

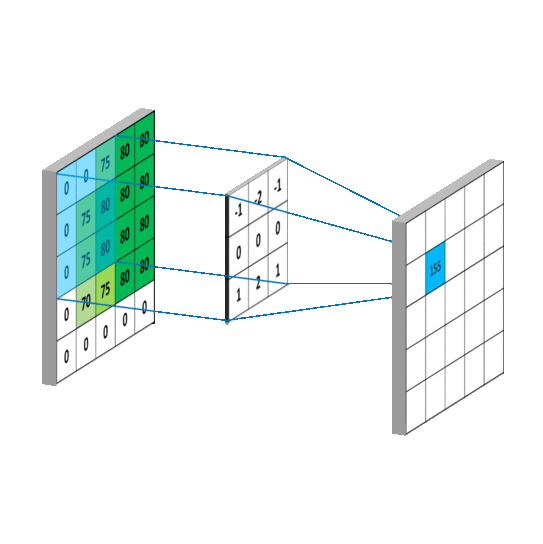
* Each CNN layer learns filters of increasing complexity.
* The first layers learn basic feature detection filters: edges, corners, etc
* The middle layers learn filters that detect parts of objects. For faces, they might learn to respond to eyes, noses, etc
* The last layers have higher representations: they learn to recognize full objects, in different shapes and positions.

**CONVOLUTIONAL LAYER.**

Convolutional layers are the layers where filters are applied to the original image, or to other feature maps in a deep CNN. This is where most of the user-specified parameters are in the network. The most important parameters are the number of kernels and the size of the kernels.



Convolution is nothing but a filter which is applied on image *to extract feature*from it. We will use such different convolutions to extract different features like edges, high-lighted patterns from the image.



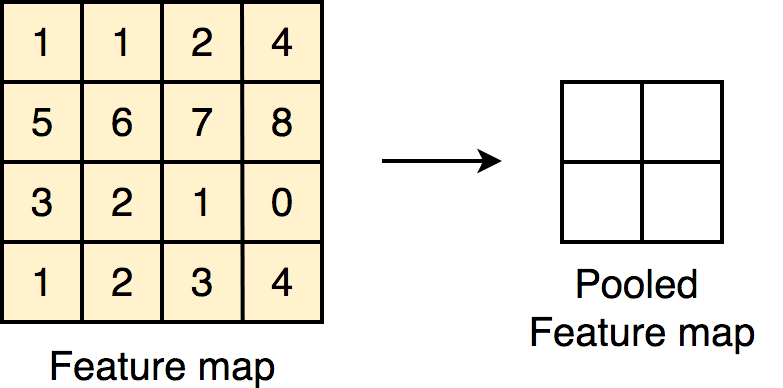
What this convolution does is, it creates a filter of some size (default size is 3X3 ). After creating filter, it starts performing element-wise multiplication starting from top left corner of image. Element-wise multiplication means multiplying elements with same index. These computed values are summed up to obtain a pixel value and it is stored in the new matrix. The size of matrix decreases as we keep on applying filters on the obtained matrix.When we say that size of convolutional layer is 32. It means that 32 randomly generated filters will be applied to the image, which outputs 32 feature matrices for that image. These feature matrices are passed to next layer as input.

**POOL LAYERS:**

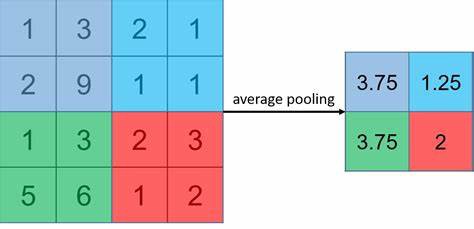
After applying convolutions, there is another concept as **pooling**. Pooling is used to reduce the size of image. There are two types of pooling :

1. **Max Pooling:**It is nothing but selecting maximum value from the matrix of specified size(default size is 2 X 2). This method is helpful to extract features with high importance or which are high-lighted in the image.

High-lighted feature is part of image having high pixel values.

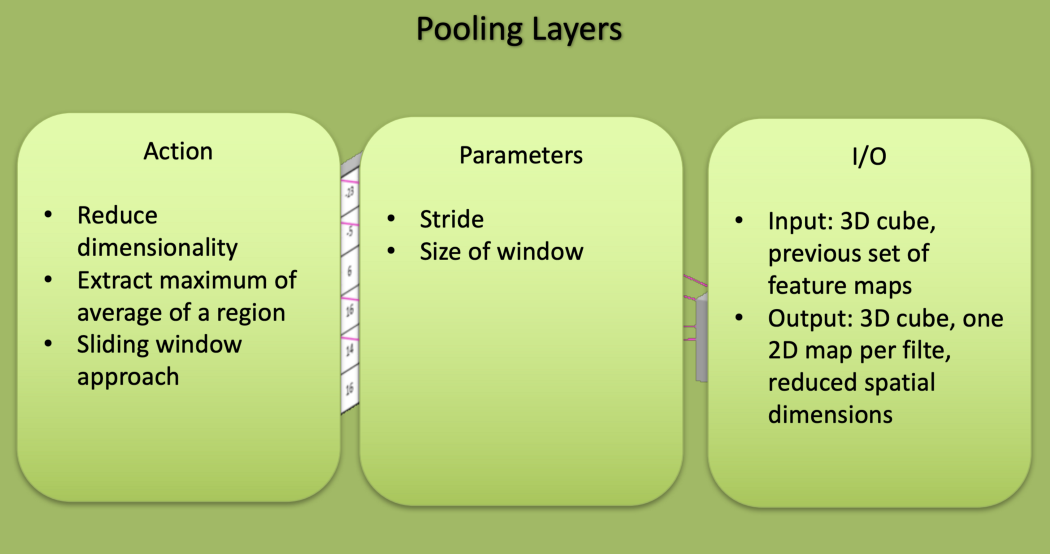


2. **Average Pooling**: Unlike Max-pooling, Average pooling takes average of all the pixel values of the matrix( default size is 2 X 2) of pooling layer.



In above example, image size is 4 X 4 and pooling size is 2 X 2. Starting from top left pixels. It will calculate average of 2 X 2 chunk matrices. For 1st 2 X 2 chunk, output value is calculated as (1+3+2+9)/4 = 15 / 4 =***3.75***. In similar way, all other values will be calculated. In most of the cases, max pooling is used because its performance is much better than average pooling.

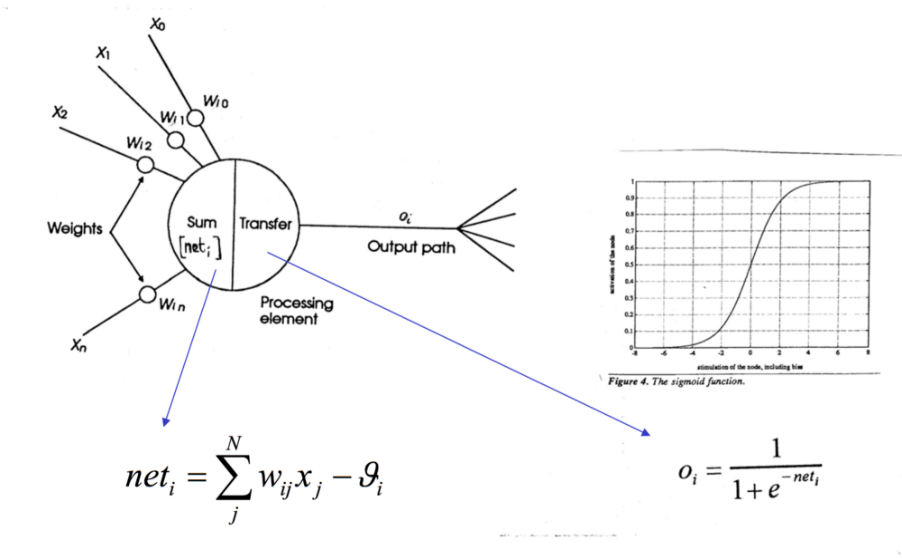
Pooling layers are similar to convolutional layers, but they perform a specific function such as max pooling, which takes the maximum value in a certain filter region, or average pooling, which takes the average value in a filter region. These are typically used to reduce the dimensionality of the network.



In any Neural Network, first layer will be input layer and last will be the output layer. Input layer contains all the inputs, here images is inputs. These images are given as input to the first convolutional layer. The output of 1st layer will be given as input to the 2nd layer, so on & so forth. This process will continue till the last layer.

**DENSE LAYER:** While defining Neural Network, first convolutional layer requires the shape of image that is passed to it as input. After passing the image, through all convolutional layers and pooling layers, output will be passed to dense layer. We can not pass output of convolutional layer directly to the dense layer because output of convolutional layer is in multi-dimensional shape and **dense layer requires input in single-dimensional** **shape**i.e. 1-D array. So we will use Flatten() method in between convolutional and dense layer. Flatten() method converts multi-dimensional matrix to single dimensional matrix. In Neural Network, non-linear function is used as activation function.

Dense Layer is simple layer of neurons in which each neuron receives input from all the neurons of previous layer, thus called as *dense*. Dense Layer is used to classify image based on output from convolutional layers.



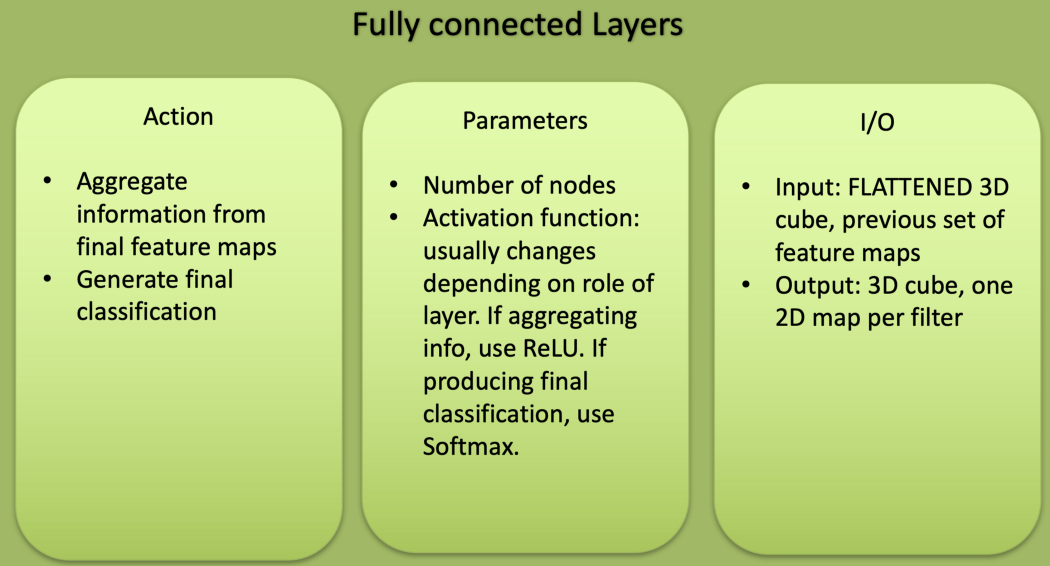
Working of single neuron. A layer contains multiple number of such neurons.

Each Layer in the *Neural Network* contains neurons, which compute the **weighted average of its input** and this weighted average is passed through a non-linear function, called as an “***activation function”***. Result of this activation function is treated as output of that neuron. In similar way, the process is carried out for all neurons of all layers.

The output of the last layer will be considered as output for that image. In same way, we will pass all the images as to convolutional layer and then to the *Neural Network,* which will produce corresponding outputs for those images. **Parameters** (params) are the *weights and biases* that will be used for *computation* in all neurons of the CNN. When we train any model on some number of images, it will determine some specific values for all parameters(i.e. weights and biases), which are used to process the image and predict the output for that image.

**FULLY CONNECTED LAYERS:**

Fully connected layers are placed before the classification output of a CNN and are used to flatten the results before classification. This is similar to the output layer of an MLP.Finally, we need the **logits layer**, which will take the output of the fully connected layer and then produce the raw prediction values.



**DROPOUT LAYER:**

Usually, when all the features are connected to the FC layer, it can cause overfitting in the training dataset. Overfitting occurs when a particular model works so well on the training data causing a negative impact in the model’s performance when used on a new data.

To overcome this problem, a dropout layer is utilised wherein a few neurons are dropped from the neural network during training process resulting in reduced size of the model. On passing a dropout of 0.3, 30% of the nodes are dropped out randomly from the neural network.

**SPLITTING OF TRAINING AND TEST DATASET:**

The train-test split is a technique for evaluating the performance of a machine learning algorithm.

The procedure involves taking a dataset and dividing it into two subsets. The first subset is used to fit the model and is referred to as the training dataset. The second subset is not used to train the model; instead, the input element of the dataset is provided to the model, then predictions are made and compared to the expected values. This second dataset is referred to as the test dataset.

* **Train Dataset**: Used to fit the machine learning model.
* **Test Dataset**: Used to evaluate the fit machine learning model.

The objective is to estimate the performance of the machine learning model on new data: data not used to train the model.

The procedure has one main configuration parameter, which is the size of the train and test sets. This is most commonly expressed as a percentage between 0 and 1 for either the train or test datasets. For example, a training set with the size of 0.67 (67 percent) means that the remainder percentage 0.33 (33 percent) is assigned to the test set.

For our model the training and test sets are split with the size of 0.8 (80 percent) assigned to training set ,means that the remainder percentage 0.2 (20 percent) is assigned to the test set.

**EVALUATION OF MODEL:**

The developed CNN model with 10 layers is trained by input images which are divided into about 35 epoches. After training of each epoch ,the train loss is computed and displayed. The Training loss graph is plotted. The evaluation of model is done by plotting ROC curves and confusion matrices.

An **ROC curve** (**receiver operating characteristic curve**) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:

* True Positive Rate
* False Positive Rate

**True Positive Rate** (**TPR**) is a synonym for recall and is therefore defined as follows:

TPR=TPTP+FN

**False Positive Rate** (**FPR**) is defined as follows:

FPR=FPFP+TN

An ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives. The following figure shows a typical ROC curve.

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.

**True Positive (TP)**

* The predicted value matches the actual value
* The actual value was positive and the model predicted a positive value

**True Negative (TN)**

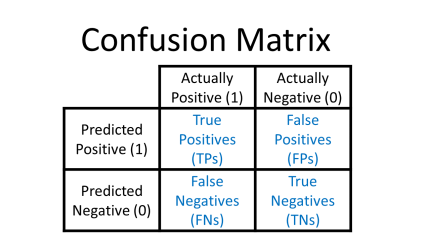
* The predicted value matches the actual value
* The actual value was negative and the model predicted a negative value

**False Positive (FP) – Type 1 error**

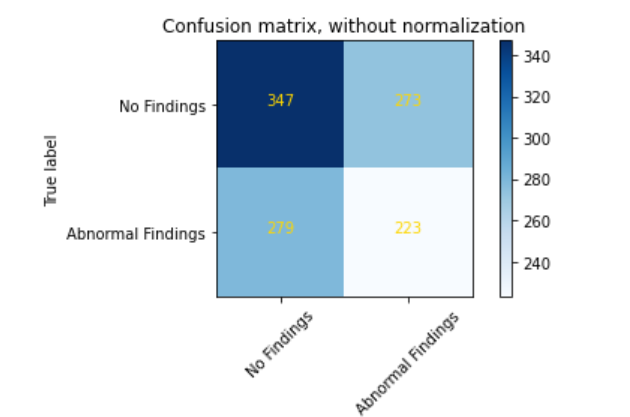
* The predicted value was falsely predicted
* The actual value was negative but the model predicted a positive value
* Also known as the **Type 1 error**

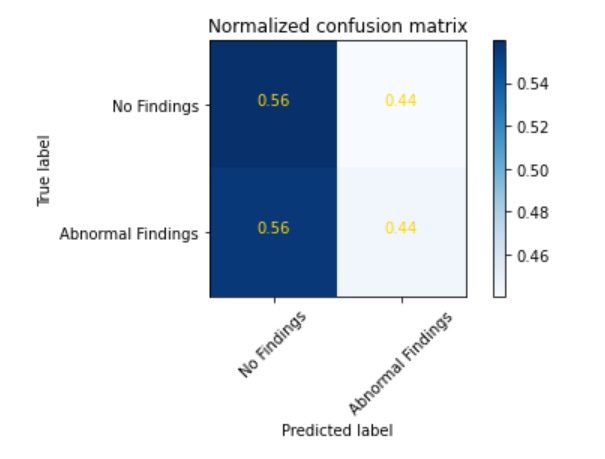
**False Negative (FN) – Type 2 error**

* The predicted value was falsely predicted
* The actual value was positive but the model predicted a negative value
* Also known as the **Type 2 error**



**CONFUSION MATRIX:**

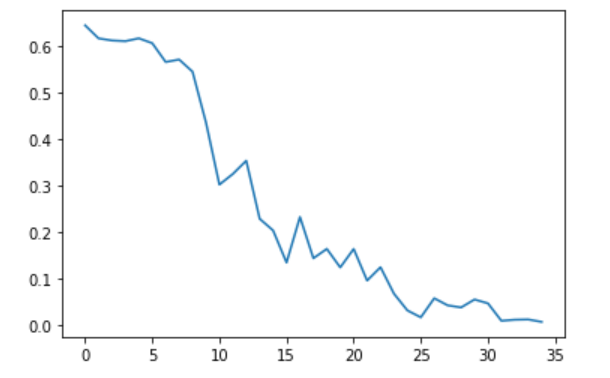




**ROC CURVE:**

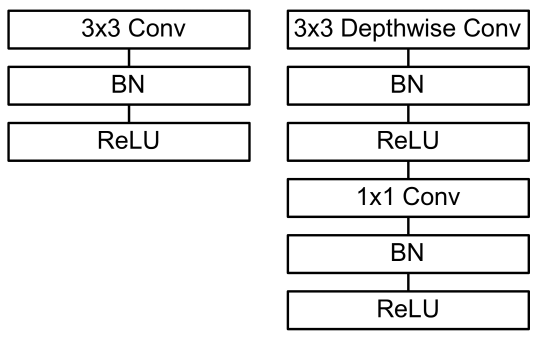


**Training Error Plot:**



It uses depthwise separable convolutions which basically means it performs a single convolution on each colour channel rather than combining all three and flattening it. This has the effect of filtering the input channels. Or as the authors of the paper explain clearly: “ For MobileNets the depthwise convolution applies a single filter to each input channel. The pointwise convolution then applies a 1×1 convolution to combine the outputs the depthwise convolution. A standard convolution both filters and combines inputs into a new set of outputs in one step. The depthwise separable convolution splits this into two layers, a separate layer for filtering and a separate layer for combining. This factorization has the effect of drastically reducing computation and model size. ”

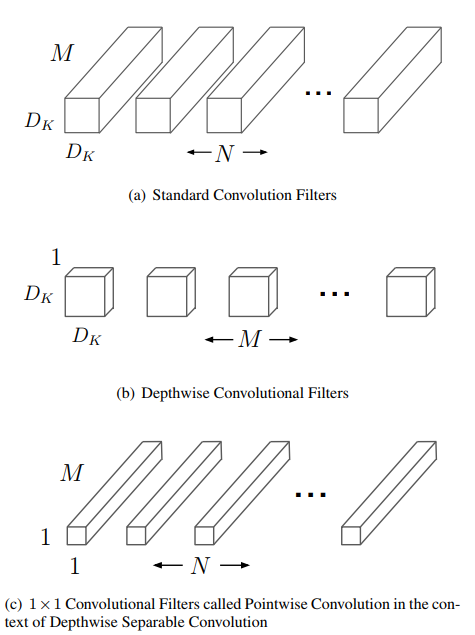
The MobileNet structure is built on depthwise separable convolutions as mentioned in the previous section except for the first layer which is a full convolution. By defining the network in such simple terms we are able to easily explore network topologies to find a good network. All layers are followed by a batchnorm  and ReLU nonlinearity with the exception of the final fully connected layer which has no nonlinearity and feeds into a softmax layer for classification.The figure below contrasts a layer with regular convolutions, batchnorm and ReLU nonlinearity to the factorized layer with depthwise convolution, 1×1 pointwise convolution as well as batchnorm and ReLU after each convolutional layer.



Down sampling is handled with strided convolution in the depthwise convolutions as well as in the first layer. A final average pooling reduces the spatial resolution to 1 before the fully connected layer. Counting depthwise and pointwise convolutions as separate layers, MobileNet has 28 layers.

It is not enough to simply define networks in terms of a small number of Mult-Adds. It is also important to make sure these operations can be efficiently implementable. For instance unstructured sparse matrix operations are not typically faster than dense matrix operations until a very high level of sparsity. Our model structure puts nearly all of the computation into dense 1×1 convolutions. This can be implemented with highly optimized general matrix multiply (GEMM) functions. Often convolutions are implemented by a GEMM but require an initial reordering in memory called im2col in order to map it to a GEMM. For instance, this approach is used in the popular Caffe package . 1×1 convolutions do not require this reordering in memory and can be implemented directly with GEMM which is one of the most optimized numerical linear algebra algorithms. MobileNet spends 95% of it’s computation time in 1×1 convolutions which also has 75% of the parameters as can be seen in Table [2](https://www.arxiv-vanity.com/papers/1704.04861/#S3.T2). Nearly all of the additional parameters are in the fully connected layer.

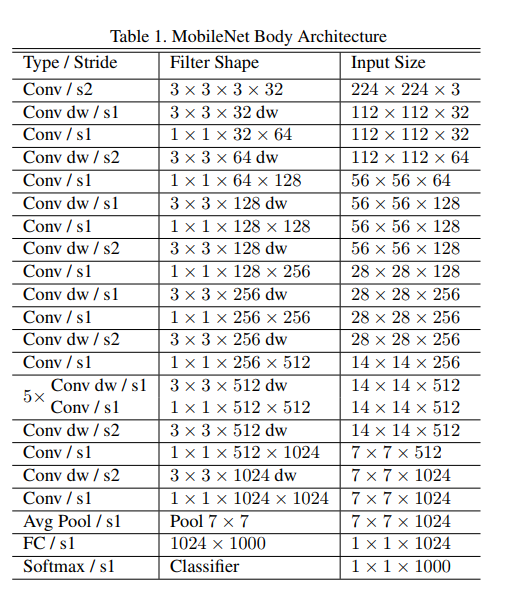
MobileNet models were trained in TensorFlow using RMSprop  with asynchronous gradient descent similar to Inception V3 . However, contrary to training large models we use less regularization and data augmentation techniques because small models have less trouble with overfitting. When training MobileNets we do not use side heads or label smoothing and additionally reduce the amount image of distortions by limiting the size of small crops that are used in large Inception training . Additionally, we found that it was important to put very little or no weight decay (l2 regularization) on the depthwise filters since their are so few parameters in them. For the ImageNet benchmarks in the next section all models were trained with same training parameters regardless of the size of the model.



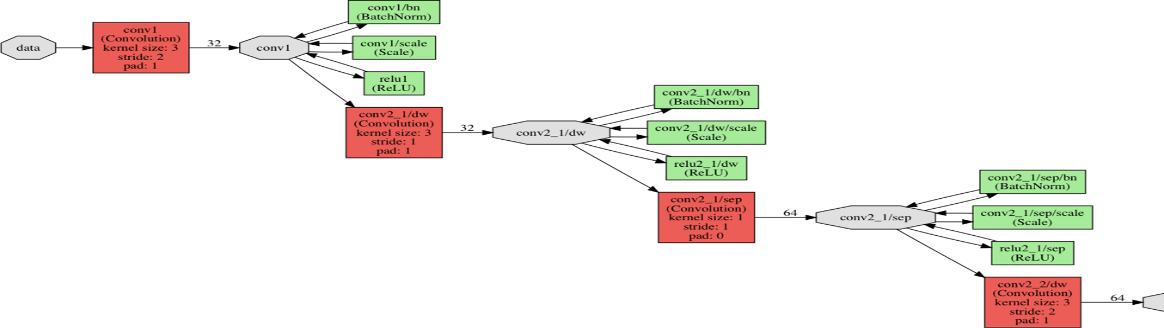
Difference between pointwise and depth wise convolutions

So the overall architecture of the Mobilenet is as follows, having 30 layers with

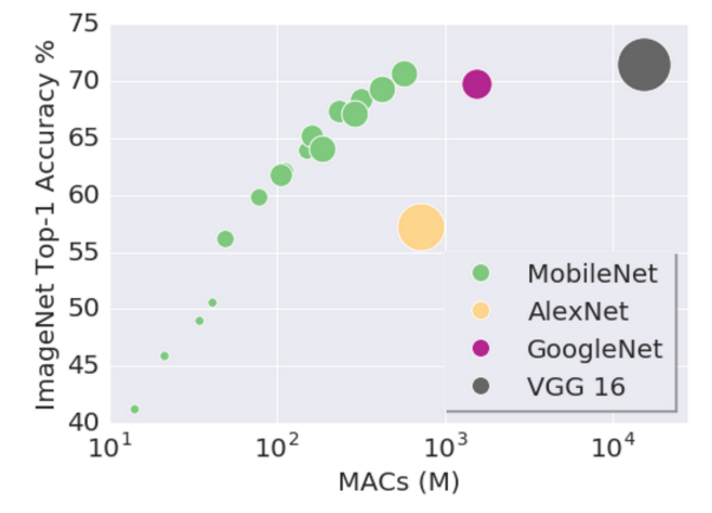
1. convolutional layer with stride 2
2. depthwise layer
3. pointwise layer that doubles the number of channels
4. depthwise layer with stride 2
5. pointwise layer that doubles the number of channels etc.



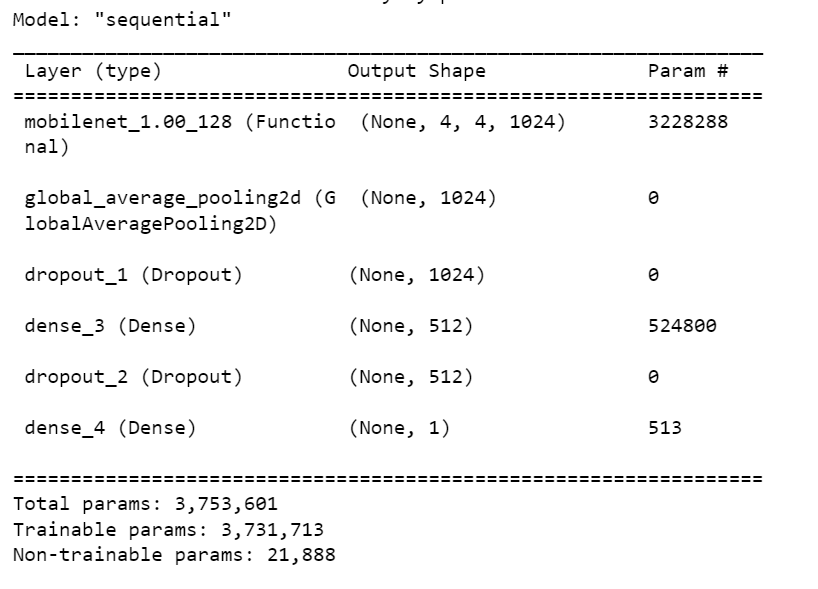
Mobile net full architecture:



It is also very low maintenance thus performing quite well with high speed. There are also many flavours of pre-trained models with the size of the network in memory and on disk being proportional to the number of parameters being used. The speed and power consumption of the network is proportional to the number of MACs (Multiply-Accumulates) which is a measure of the number of fused Multiplication and Addition operations.



A Mobile Net model is developed with 6 layers and trained for both 128 x 128 Grayscaled image and also for the original 256 x 256 Image. The summary of our developed model is shown below.



The model is trained by splitting the training dataset into 35 epoches for the Grayscaled images. The model is trained by splitting the training dataset into 40 epoches for the Original images. The Evaluation of the model is done by plotting ROC curve and Confusion matrices.

The activation function used for training our model is sigmoid function. When the activation function for a neuron is a sigmoid function it is a guarantee that the output of this unit will always be between 0 and 1. Also, as the sigmoid is a non-linear function, the output of this unit would be a non-linear function of the weighted sum of inputs.

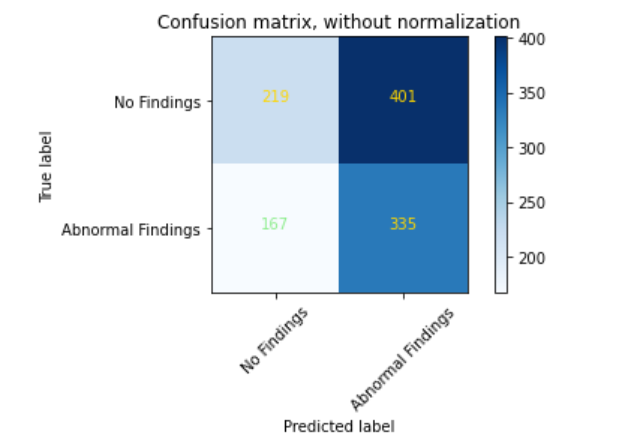
The optimizer used for minimizing the loss is ADAM optimizer. Adam is a replacement optimization algorithm for stochastic gradient descent for training deep learning models.

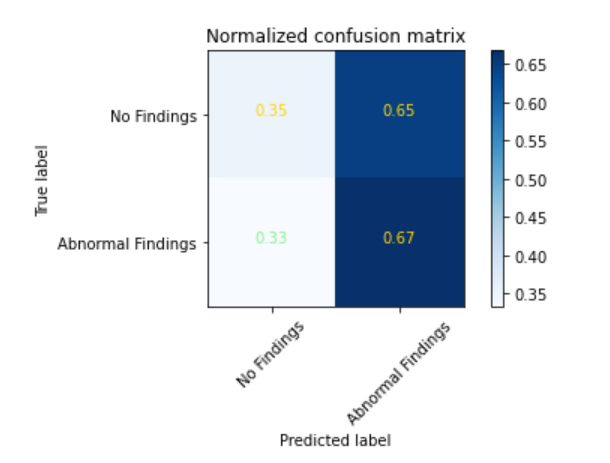
Adam combines the best properties of the AdaGrad and RMSProp algorithms to provide an optimization algorithm that can handle sparse gradients on noisy problems.

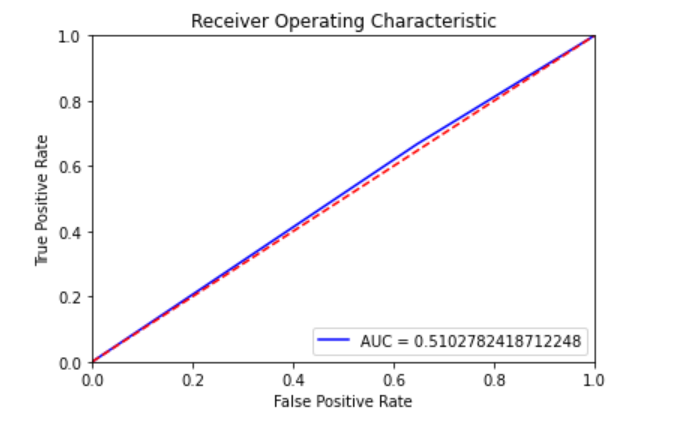
The loss function used in our model is Binary cross entropy. It compares each of the predicted probabilities to actual class output which can be either 0 or 1. It then calculates the score that penalizes the probabilities based on the distance from the expected value. That means how close or far from the actual value.

The metrices used for evaluating are binary accuracy and MAE(Mean Absolute error)

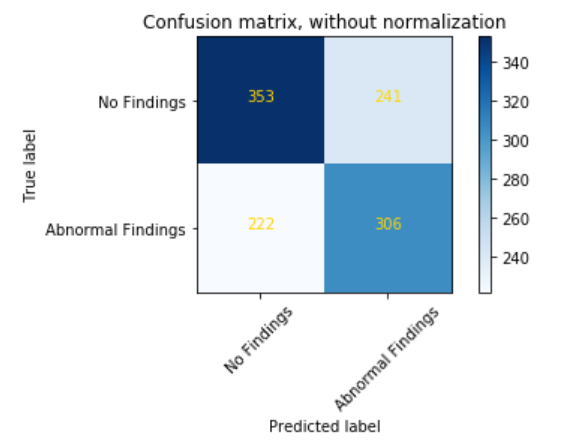
FOR GRAYSCALED TRAINED IMAGES:

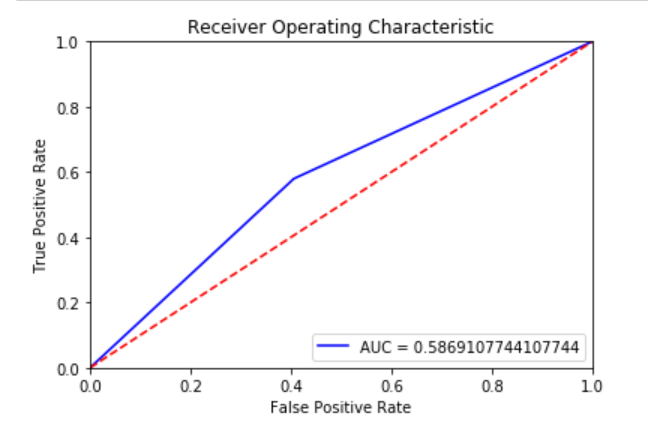
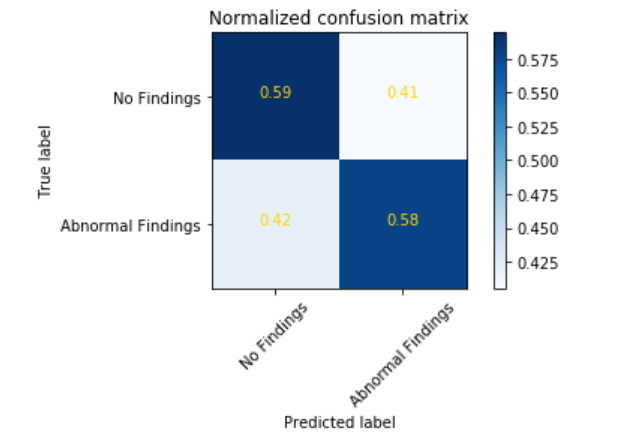






FOR ORIGINAL TRAINED IMAGES:





**InceptionV3:**

Inception v3 mainly focuses on burning less computational power by modifying the previous Inception architectures.

The Inception V3 is a deep learning model based on Convolutional Neural Networks, which is used for image classification. The inception V3 is a superior version of the basic model Inception V1 which was introduced as GoogLeNet in 2014. As the name suggests it was developed by a team at Google.

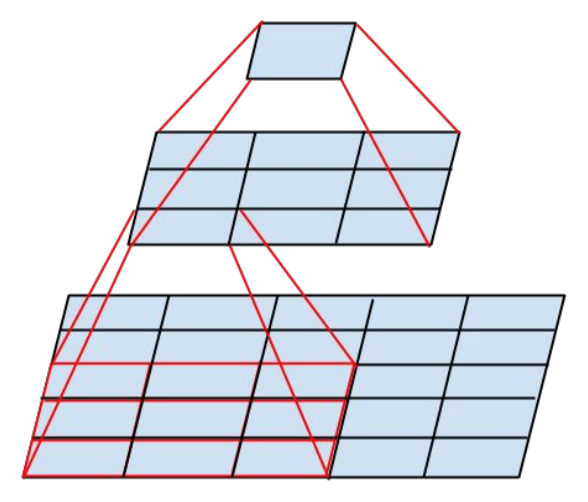
In comparison to VGGNet, Inception Networks (GoogLeNet/Inception v1) have proved to be more computationally efficient, both in terms of the number of parameters generated by the network and the economical cost incurred (memory and other resources). If any changes are to be made to an Inception Network, care needs to be taken to make sure that the computational advantages aren’t lost. Thus, the adaptation of an Inception network for different use cases turns out to be a problem due to the uncertainty of the new network’s efficiency. In an Inception v3 model, several techniques for optimizing the network have been put suggested to loosen the constraints for easier model adaptation. The techniques include factorized convolutions, regularization, dimension reduction, and parallelized computations.

Inception v3 Architecture

The architecture of an Inception v3 network is progressively built, step-by-step, as explained below:

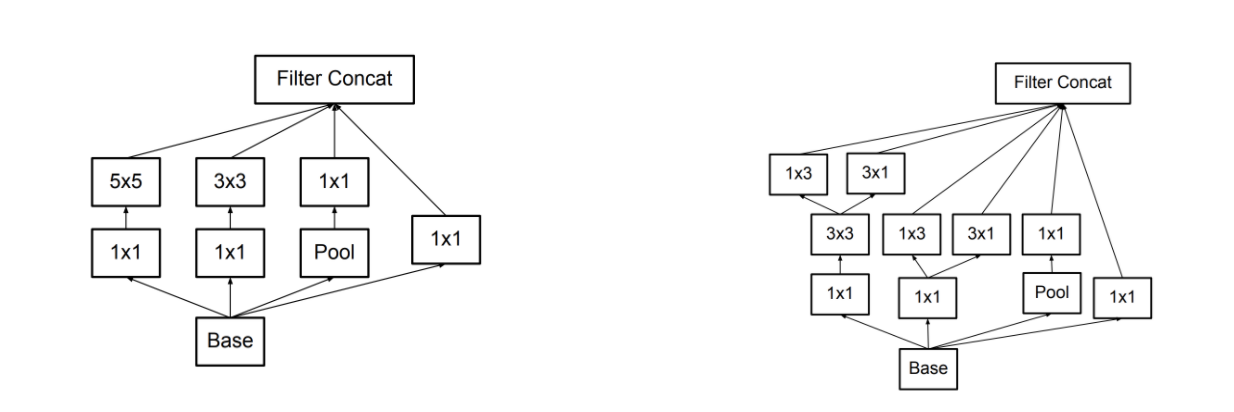
1. Factorized Convolutions: this helps to reduce the computational efficiency as it reduces the number of parameters involved in a network. It also keeps a check on the network efficiency.

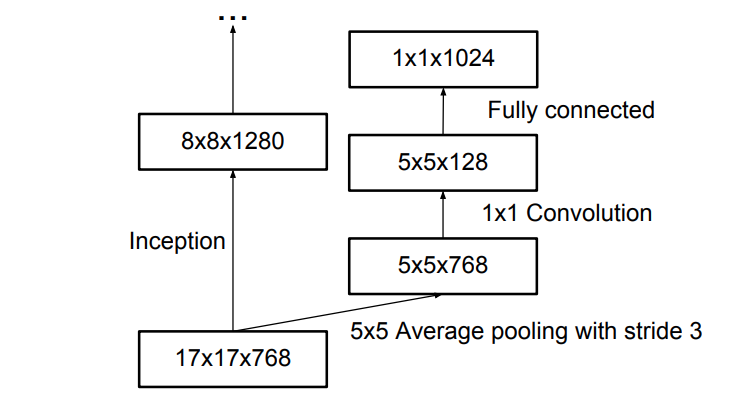
2. Smaller convolutions: replacing bigger convolutions with smaller convolutions definitely leads to faster training. Say a 5 × 5 filter has 25 parameters; two 3 × 3 filters replacing a 5 × 5 convolution has only 18 (3\*3 + 3\*3) parameters instead.



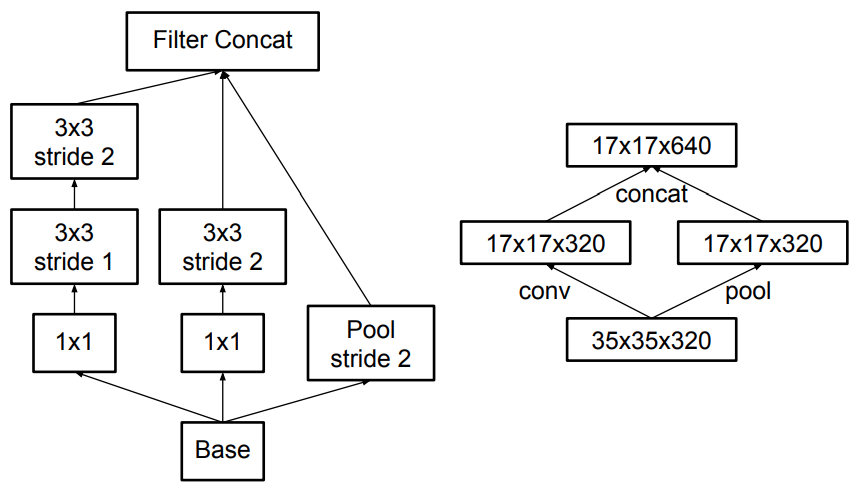
In the middle we see a 3x3 convolution, and below a fully-connected layer. Since both 3x3 convolutions can share weights among themselves, the number of computations can be reduced.

3. Asymmetric convolutions: A 3 × 3 convolution could be replaced by a 1 × 3 convolution followed by a 3 × 1 convolution. If a 3 × 3 convolution is replaced by a 2 × 2 convolution, the number of parameters would be slightly higher than the asymmetric convolution proposed.

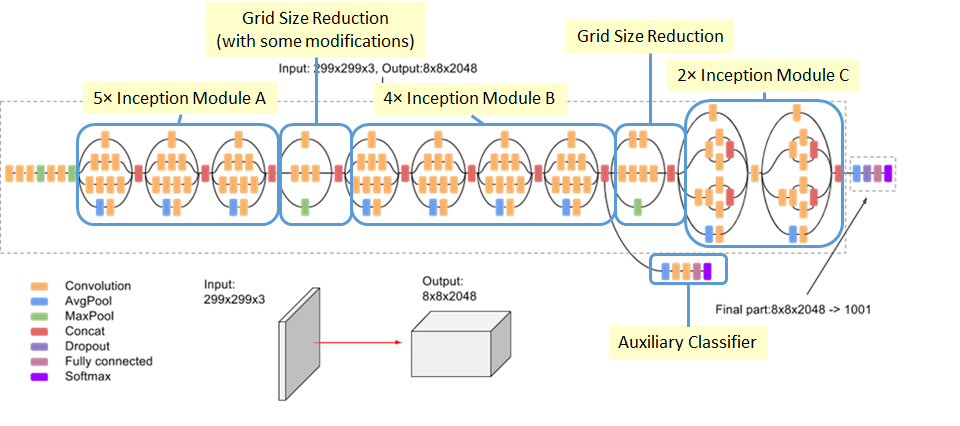


4. Auxiliary classifier: an auxiliary classifier is a small CNN inserted between layers during training, and the loss incurred is added to the main network loss. In GoogLeNet auxiliary classifiers were used for a deeper network, whereas in Inception v3 an auxiliary classifier acts as a regularizer.

5. Grid size reduction: Grid size reduction is usually done by pooling operations. However, to combat the bottlenecks of computational cost, a more efficient technique is proposed:



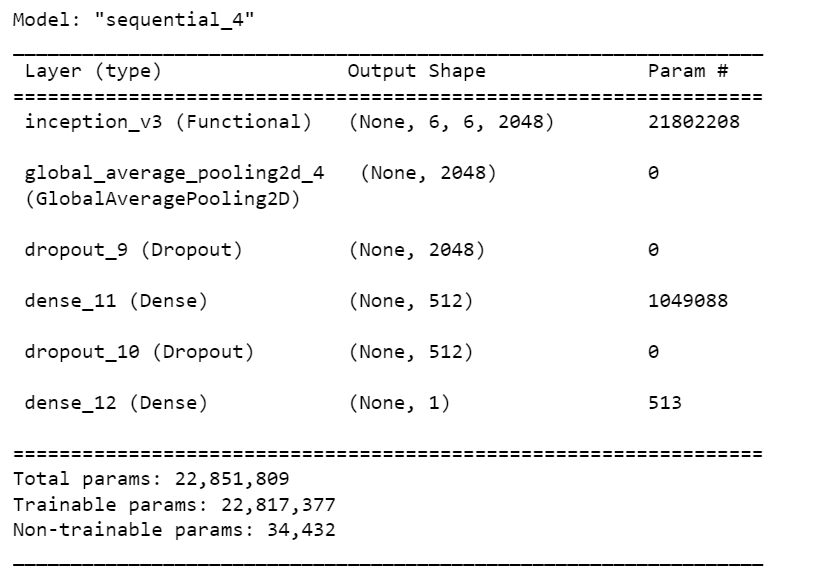
All the above concepts are consolidated into the final architecture.



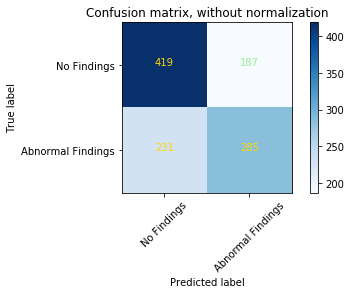
A Inception V3 model is developed with 6 layers and trained for both 256 x 256 Grayscaled image and also for the original 256 x 256 Image. The summary of our developed model is shown below.

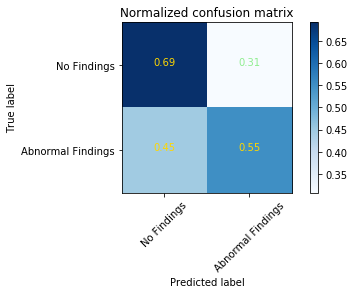
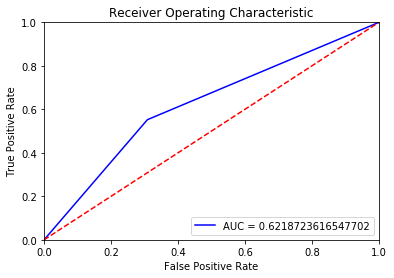
The model is trained by splitting the training dataset into 40 epoches for the Grayscaled images. The model is trained by splitting the training dataset into 40 epoches for the Original images. The Evaluation of the model is done by plotting ROC curve and Confusion matrices.

The hyperparameters used in training of this model is same as that of MobileNet model



**FOR GRAYSCALED TRAINED IMAGES:**





**FOR ORIGINAL TRAINED IMAGES:**

