**PNEUMOTHORAX OR NORMAL CLASSIFICATION USING CNN**

**ABSTRACT:**

A pneumothorax (noo-moe-THOR-aks) is a collapsed lung. A pneumothorax occurs when air leaks into the space between your lung and chest wall. This air pushes on the outside of your lung and makes it collapse. A pneumothorax can be a complete lung collapse or a collapse of only a portion of the lung. A collapsed lung occurs when air enters the pleural space, the area between the chest wall and the lung. Air in the pleural space can build up and press against the lung, causing it to collapse partially or fully. Also called a deflated lung or pneumothorax, a collapsed lung needs immediate medical care. The proposed system based on CNN using images to classifying, pneumothorax or Normal in this system using CNN model. It is predicted that the success of the obtained results will increase if the CNN method is supported by adding extra feature extraction methods and classify successfully pneumothorax. We have demonstrated the efficacy and potential of using deep CNN to images.

**Keywords:** pneumothorax, deep learning, TensorFlow, Keras, CNN

**EXISTING SYSTEM:**

Pneumothorax is potentially a life-threatening disease that requires urgent diagnosis and treatment. The chest Xray is the diagnostic modality of choice when pneumothorax is suspected. The computer-aided diagnosis of pneumothorax has received a dramatic boost in the last few years due to deep learning advances and the first public pneumothorax diagnosis competition with 15257 chest X-rays manually annotated by a team of 19 radiologists. This paper describes one of the top frameworks that participated in the competition. The framework investigates the benefits of combining the Unet convolutional neural network with various backbones, namely ResNet34, SE-ResNext50, SEResNext101, and DenseNet121. The paper presents a step-by-step instruction for the framework application, including data augmentation, and different pre- and post-processing steps. The performance of the framework was of 0.8574 measured in terms of the Dice coefficient

**Drawbacks:**

* It has focused segmentation ,It has not focused on identifying classification method
* There is no any comparison in different architecture.
* There is no any deployment in web frame

**PREPARING THE DATASET:**

This dataset contains approximately 992 train and 225 test image records which were then classified into 2 classes:

* NORMAL
* PNEUMOTHORAX

**PROPOSED SYSTEM:**

* We are proposing recognition framework based on the structured two dimensional Convolutional Neural Networks (CNNs) type of AlexNet to classify the Pneumothorax and improve the accuracy of workflow.
* The proposed method for this project is to train a Deep Learning algorithm capable of classifying pneumothorax images and data preprocessing and visualizing the image then feature extracting to build AlexNet CNN using Pneumothorax image dataset. we classify it such as Pneumothorax or Normal this system using CNN model.
* It is predicted that the success of the obtained results will increase if the CNN method is supported by adding extra feature extraction methods and classify successfully Pneumothorax. We have demonstrated the efficacy and potential of using deep CNN to images.

**Advantages:**

* The large amount of chest x-ray data can be train on artificial neural network.
* It is best model for deep learning technique to easily classifying Pneumothorax

**LITERATURE SURVEY**

**General**

A literature review is a body of text that aims to review the critical points of current knowledge on and/or methodological approaches to a particular topic. It is secondary sources and discuss published information in a particular subject area and sometimes information in a particular subject area within a certain time period. Its ultimate goal is to bring the reader up to date with current literature on a topic and forms the basis for another goal, such as future research that may be needed in the area and precedes a research proposal and may be just a simple summary of sources. Usually, it has an organizational pattern and combines both summary and synthesis.

A summary is a recap of important information about the source, but a synthesis is a re-organization, reshuffling of information. It might give a new interpretation of old material or combine new with old interpretations or it might trace the intellectual progression of the field, including major debates. Depending on the situation, the literature review may evaluate the sources and advise the reader on the most pertinent or relevant of them.

**Review of Literature Survey**

**Title** : Automatic Diagnosis of Pneumothorax from Chest Radiographs: A Systematic Literature Review

**Author**: Tahira Iqbal , Arslan Shaukat , Usman Akram

**Year** : 2015

Among various medical imaging tools, chest radiographs are the most important and widely used diagnostic tool for detection of thoracic pathologies. Researches are being carried out in order to propose robust automatic diagnostic tool for detection of pathologies from chest radiographs. Artificial Intelligence techniques especially deep learning methodologies have been found to be giving promising results in automating the field of medicine. Lot of research has been done for automatic and fast detection of pneumothorax from chest radiographs while proposing several frameworks based on artificial intelligence and machine learning techniques. This study summarizes the existing literature for the automatic detection of pneumothorax from chest x-rays along with describing the available chest radiographs datasets. The comparative analysis of the literature is also provided in terms of goodness and limitations of the existing literature along with highlighting the research gaps which need to be further explored. The paper provides a brief overview of the present work for pneumothorax detection for helping the researchers in selection of optimal approach for future research.

# **Title** : Effective Pneumothorax Detection for Chest X-Ray Images Using Local Binary Pattern and Support Vector Machine

**Author**: [Yuan-Hao Chan](https://pubmed.ncbi.nlm.nih.gov/?term=Chan+YH&cauthor_id=29849996), [Yong-Zhi Zeng](https://pubmed.ncbi.nlm.nih.gov/?term=Zeng+YZ&cauthor_id=29849996), [Hsien-Chu Wu](https://pubmed.ncbi.nlm.nih.gov/?term=Wu+HC&cauthor_id=29849996), [Ming-Chi Wu](https://pubmed.ncbi.nlm.nih.gov/?term=Wu+MC&cauthor_id=29849996), [Hung-Min Sun](https://pubmed.ncbi.nlm.nih.gov/?term=Sun+HM&cauthor_id=29849996)

**Year** : 2018

Automatic image segmentation and feature analysis can assist doctors in the treatment and diagnosis of diseases more accurately. Automatic medical image segmentation is difficult due to the varying image quality among equipment. In this paper, the automatic method employed image multiscale intensity texture analysis and segmentation to solve this problem. In this paper, firstly, SVM is applied to identify common pneumothorax. Features are extracted from lung images with the LBP (local binary pattern). Then, classification of pneumothorax is determined by SVM. Secondly, the proposed automatic pneumothorax detection method is based on multiscale intensity texture segmentation by removing the background and noises in chest images for segmenting abnormal lung regions. The segmentation of abnormal regions is used for texture transformed from computing multiple overlapping blocks. The rib boundaries are identified with Sobel edge detection. Finally, in obtaining a complete disease region, the rib boundary is filled up and located between the abnormal regions.

# **Title** : Deep Learning for Pneumothorax Detection and Localization in Chest Radiographs

**Author**: [André Gooßen](https://arxiv.org/search/eess?searchtype=author&query=Goo%C3%9Fen%2C+A), [Hrishikesh Deshpande](https://arxiv.org/search/eess?searchtype=author&query=Deshpande%2C+H), [Tim Harder](https://arxiv.org/search/eess?searchtype=author&query=Harder%2C+T), [Evan Schwab](https://arxiv.org/search/eess?searchtype=author&query=Schwab%2C+E), [Ivo Baltruschat](https://arxiv.org/search/eess?searchtype=author&query=Baltruschat%2C+I), [Thusitha Mabotuwana](https://arxiv.org/search/eess?searchtype=author&query=Mabotuwana%2C+T), [Nathan Cross](https://arxiv.org/search/eess?searchtype=author&query=Cross%2C+N), [Axel Saalbach](https://arxiv.org/search/eess?searchtype=author&query=Saalbach%2C+A)

**Year** : 2019

Pneumothorax is a critical condition that requires timely communication and immediate action. In order to prevent significant morbidity or patient death, early detection is crucial. For the task of pneumothorax detection, we study the characteristics of three different deep learning techniques: (i) convolutional neural networks, (ii) multiple-instance learning, and (iii) fully convolutional networks. We perform a five-fold cross-validation on a dataset consisting of 1003 chest X-ray images. ROC analysis yields AUCs of 0.96, 0.93, and 0.92 for the three methods, respectively. We review the classification and localization performance of these approaches as well as an ensemble of the three aforementioned techniques.

# **Title** : Crowdsourcing pneumothorax annotations using machine learning annotations on the NIH chest X-ray dataset

**Author**: [Ross W Filice](https://pubmed.ncbi.nlm.nih.gov/?term=Filice+RW&cauthor_id=31768897), [Anouk Stein](https://pubmed.ncbi.nlm.nih.gov/?term=Stein+A&cauthor_id=31768897), [Carol C Wu](https://pubmed.ncbi.nlm.nih.gov/?term=Wu+CC&cauthor_id=31768897), [Veronica A Arteaga](https://pubmed.ncbi.nlm.nih.gov/?term=Arteaga+VA&cauthor_id=31768897), [Stephen Borstelmann](https://pubmed.ncbi.nlm.nih.gov/?term=Borstelmann+S&cauthor_id=31768897), [Ramya Gaddikeri](https://pubmed.ncbi.nlm.nih.gov/?term=Gaddikeri+R&cauthor_id=31768897), [Maya Galperin-Aizenberg](https://pubmed.ncbi.nlm.nih.gov/?term=Galperin-Aizenberg+M&cauthor_id=31768897), [Ritu R Gill](https://pubmed.ncbi.nlm.nih.gov/?term=Gill+RR&cauthor_id=31768897), [Myrna C Godoy](https://pubmed.ncbi.nlm.nih.gov/?term=Godoy+MC&cauthor_id=31768897), [Stephen B Hobbs](https://pubmed.ncbi.nlm.nih.gov/?term=Hobbs+SB&cauthor_id=31768897), [Jean Jeudy](https://pubmed.ncbi.nlm.nih.gov/?term=Jeudy+J&cauthor_id=31768897), [Paras C Lakhani](https://pubmed.ncbi.nlm.nih.gov/?term=Lakhani+PC&cauthor_id=31768897), [Archana Laroia](https://pubmed.ncbi.nlm.nih.gov/?term=Laroia+A&cauthor_id=31768897), [Sundeep M Nayak](https://pubmed.ncbi.nlm.nih.gov/?term=Nayak+SM&cauthor_id=31768897), [Maansi R Parekh](https://pubmed.ncbi.nlm.nih.gov/?term=Parekh+MR&cauthor_id=31768897), [Prasanth Prasanna](https://pubmed.ncbi.nlm.nih.gov/?term=Prasanna+P&cauthor_id=31768897), [Palmi Shah](https://pubmed.ncbi.nlm.nih.gov/?term=Shah+P&cauthor_id=31768897), [Dharshan Vummidi](https://pubmed.ncbi.nlm.nih.gov/?term=Vummidi+D&cauthor_id=31768897), [Kavitha Yaddanapudi](https://pubmed.ncbi.nlm.nih.gov/?term=Yaddanapudi+K&cauthor_id=31768897), [George Shih](https://pubmed.ncbi.nlm.nih.gov/?term=Shih+G&cauthor_id=31768897)

**Year** : 2019

Pneumothorax is a potentially life-threatening condition that requires prompt recognition and often urgent intervention. In the ICU setting, large numbers of chest radiographs are performed and must be interpreted on a daily basis which may delay diagnosis of this entity. Development of artificial intelligence (AI) techniques to detect pneumothorax could help expedite detection as well as localize and potentially quantify pneumothorax. Open image analysis competitions are useful in advancing state-of-the art AI algorithms but generally require large expert annotated datasets. We have annotated and adjudicated a large dataset of chest radiographs to be made public with the goal of sparking innovation in this space. Because of the cumbersome and time-consuming nature of image labeling, we explored the value of using AI models to generate annotations for review. Utilization of this machine learning annotation (MLA) technique appeared to expedite our annotation process with relatively high sensitivity at the expense of specificity. Further research is required to confirm and better characterize the value of MLAs. Our adjudicated dataset is now available for public consumption in the form of a challenge.

# **Title** : Deep learning for chest X-ray analysis: A survey

**Author**: [ErdiÇallı](https://www.sciencedirect.com/science/article/pii/S1361841521001717" \l "!) , [EcemSogancioglu,](https://www.sciencedirect.com/science/article/pii/S1361841521001717#!)  [Bramvan Ginneken](https://www.sciencedirect.com/science/article/pii/S1361841521001717#!),[Kicky G.van Leeuwen](https://www.sciencedirect.com/science/article/pii/S1361841521001717#!), [KeelinMurphy](https://www.sciencedirect.com/science/article/pii/S1361841521001717#!)

**Year** : 2021

Recent advances in deep learning have led to a promising performance in many [medical image analysis](https://www.sciencedirect.com/topics/medicine-and-dentistry/image-analysis-medical-imaging) tasks. As the most commonly performed radiological exam, chest radiographs are a particularly important modality for which a variety of applications have been researched. The release of multiple, large, publicly available chest X-ray datasets in recent years has encouraged research interest and boosted the number of publications. In this paper, we review all studies using deep learning on chest radiographs published before March 2021, categorizing works by task: image-level prediction (classification and regression), segmentation, localization, image generation and domain adaptation. Detailed descriptions of all publicly available datasets are included and commercial systems in the field are described. A comprehensive discussion of the current state of the art is provided, including caveats on the use of public datasets, the requirements of clinically useful systems and gaps in the current literature.

**SYSTEM STUDY**

**Project Goals:**

* Load the data
* Loading the given dataset
* Import required libraries packages

# Pre-process the data

* Reshape, data augmentations

# Define model

* Sequential or Functional
* Number of layers to be used, Number of nodes to be used in the model, Evaluation metrics

# Compile the model

* Define loss function, optimizer, weights and bias

# Fit the model

* Train data, Test data, epoch, Batchsize.

#### **Objectives:**

The goal is to develop a deep learning model for Pneumothorax, normal

image classification by convolutional neural network algorithm for potentially classifying the results in the form of best accuracy by comparing the CNN architectures.

**Scope:**

The scope of this paper is to implement and investigate how different CNN methods impact the prediction. The scope in our research is to split this classes into Pneumothorax and normal images. This can be classified into 2 classes. Besides, extracting hand crafted features from raw images after different processing is one of the best scopes we have worked in this research. The objective of using this method is to get some features which are specifically responsible for the Pneumothorax disease prediction.

**FEASIBILITY STUDY**

**Splitting the dataset:**

The data use is usually split into training data and test data. The training set contains a known output and the model learns on this data in order to be generalized to other data later on. It has the test dataset (or subset) in order to test our models and it will do this using Tensor flow library in Python using the Keras method.

**Construction of a Detecting Model:**

## Deep learning needs data gathering have lot of past image data’s. Training and testing this model working and predicting correctly.

Data Gathering

CNN Algorithm

Train model

Prediction

Test model

Fig: Steps of dataflow diagram

**Data Flow Diagram:**

Image Details

Test dataset

Preprocessing

Prediction of Pneumothorax

CNN Algorithm

Training dataset

Fig: Process of dataflow diagram

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. A DFD is often used as a preliminary step to create an overview of the system without going into great detail, which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kind of information will be input to and output from the system, how the data will advance through the system, and where the data will be stored. It does not show information about process timing or whether processes will operate in sequence or in parallel, unlike a traditional structured flowchart which focuses on control flow, or a UML activity workflow diagram, which presents both control and data flows as a unified model. Data flow diagrams are also known as bubble charts. DFD is a designing tool used in the top down approach to Systems Design. Symbols and Notations Used in DFDs Using any convention’s DFD rules or guidelines, the symbols depict the four components of data flow diagrams.

External entity: an outside system that sends or receives data, communicating with the system being diagrammed. They are the sources and destinations of information entering or leaving the system. They might be an outside organization or person, a computer system or a business system. They are also known as terminators, sources and sinks or actors. They are typically drawn on the edges of the diagram.

Process: any process that changes the data, producing an output. It might perform computations, or sort data based on logic, or direct the data flow based on business rules.

Data store: files or repositories that hold information for later use, such as a database table or a membership form.

Data flow: the route that data takes between the external entities, processes and data stores. It portrays the interface between the other components and is shown with arrows, typically labeled with a short data name, like “Billing details.”

DFD levels and layers A data flow diagram can dive into progressively more detail by using levels and layers, zeroing in on a particular piece. DFD levels are numbered 0, 1 or 2, and occasionally go to even Level 3 or beyond. The necessary level of detail depends on the scope of what you are trying to accomplish. DFD Level 0 is also called a Context Diagram. It’s a basic overview of the whole system or process being analyzed or modeled. It’s designed to be an at-a-glance view, showing the system as a single high-level process, with its relationship to external entities. It should be easily understood by a wide audience, including stakeholders, business analysts, data analysts and developers.

Level 0:

Image

Disease Classification

Level 1:

Pneumothorax disease prediction

Feature extraction

Test image

Training dataset

Level 2:

Testing dataset

Training dataset

Level 3:

pneumothorax features

Pneumothorax disease

**LIST OF MODULES**

1. Manual Net

2. AlexNet

3. LeNet

4. Deploy

**PROJECT REQUIREMENTS**

**General:**

Requirements are the basic constrains that are required to develop a system. Requirements are collected while designing the system. The following are the requirements that are to be discussed.

1. Functional requirements

2. Non-Functional requirements

3. Environment requirements

A. Hardware requirements

B. software requirements

**Functional requirements:**

The software requirements specification is a technical specification of requirements for the software product. It is the first step in the requirements analysis process. It lists requirements of a particular software system. The following details to follow the special libraries like tensorflow, keras, matplotlib.

**Non-Functional Requirements:**

Process of functional steps,

1. Problem define
2. Preparing data
3. Evaluating algorithm
4. Improving results
5. Prediction the result

**Environment Requirements:**

**1.** **Software Requirements**:

Operating System : Windows / Linux

Simulation Tool : Anaconda with Jupyter Notebook

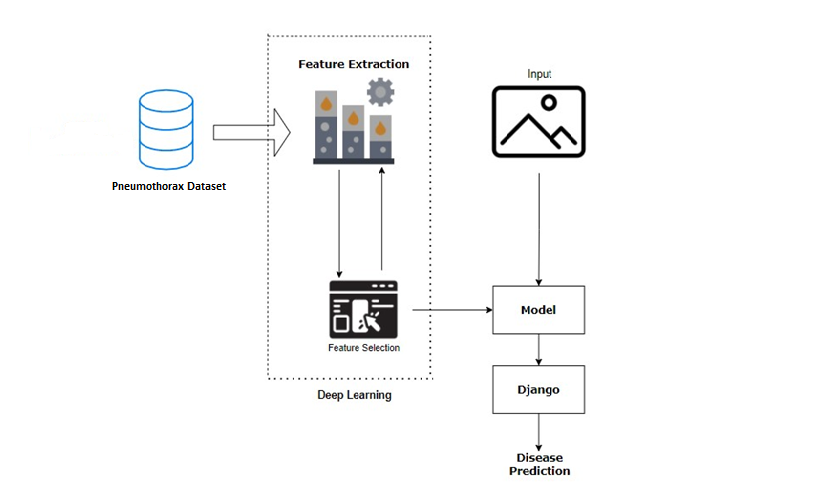
**2.** **Hardware requirements**:

Processor : Pentium IV/III

Hard disk : minimum 80 GB

RAM : minimum 2 GB

**SYSTEM ARCHITECTURE:**



**WORK FLOW DIAGRAM:**

Source images

Testing Dataset

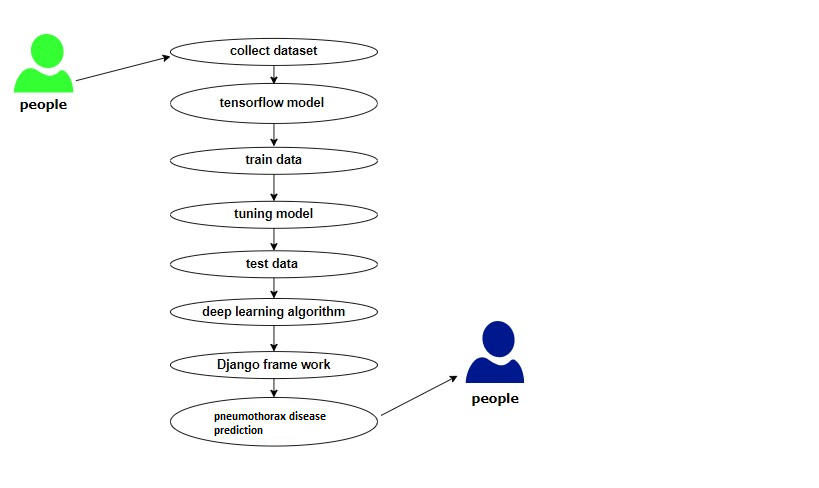
Training Dataset

CNN algorithm

Prediction of Pneumothorax disease

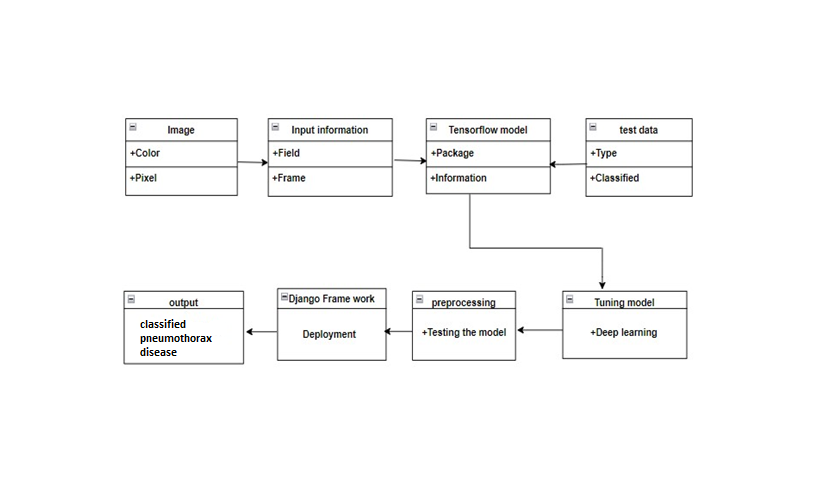
Workflow Diagram

**USECASE DIAGRAM:**



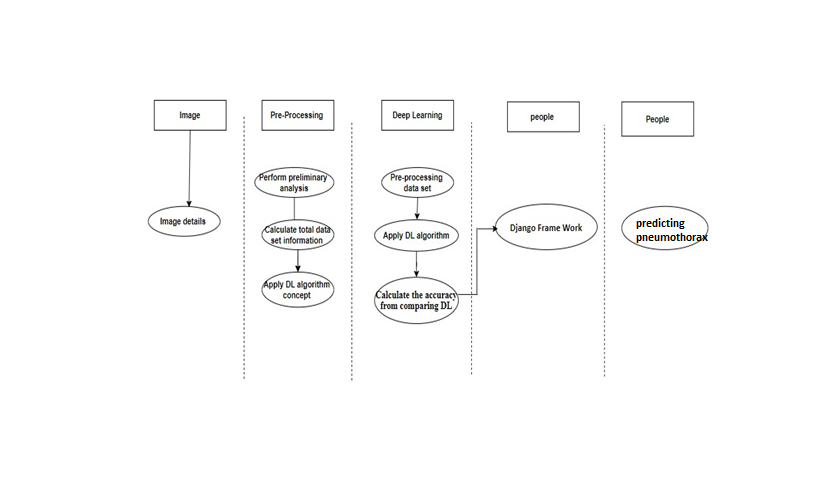
Use case diagrams are considered for high level requirement analysis of a system. So when the requirements of a system are analyzed the functionalities are captured in use cases. So, it can say that uses cases are nothing but the system functionalities written in an organized manner.

**CLASS DIAGRAM:**



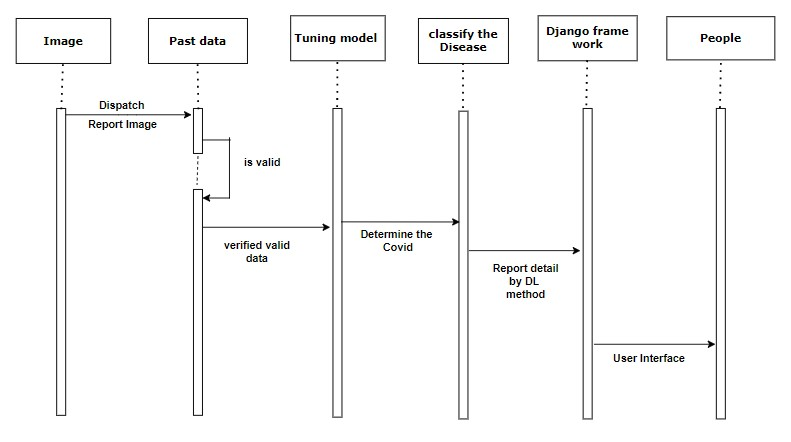
Class diagram is basically a graphical representation of the static view of the system and represents different aspects of the application. So a collection of class diagrams represent the whole system. The name of the class diagram should be meaningful to describe the aspect of the system. Each element and their relationships should be identified in advance Responsibility (attributes and methods) of each class should be clearly identified for each class minimum number of properties should be specified and because, unnecessary properties will make the diagram complicated. Use notes whenever required to describe some aspect of the diagram and at the end of the drawing it should be understandable to the developer/coder. Finally, before making the final version, the diagram should be drawn on plain paper and rework as many times as possible to make it correct.

**ACTIVITY DIAGRAM**:



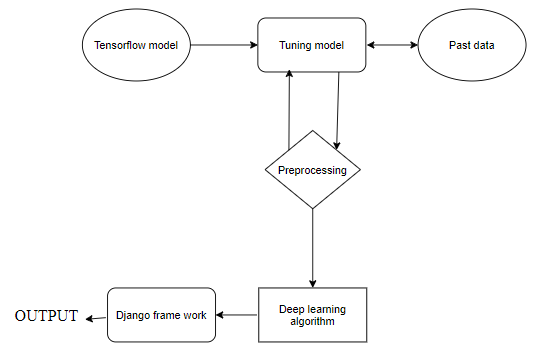
Activity is a particular operation of the system. Activity diagrams are not only used for visualizing dynamic nature of a system but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in activity diagram is the message part. It does not show any message flow from one activity to another. Activity diagram is some time considered as the flow chart. Although the diagrams looks like a flow chart but it is not. It shows different flow like parallel, branched, concurrent and single.

**SEQUENCE DIAGRAM:**



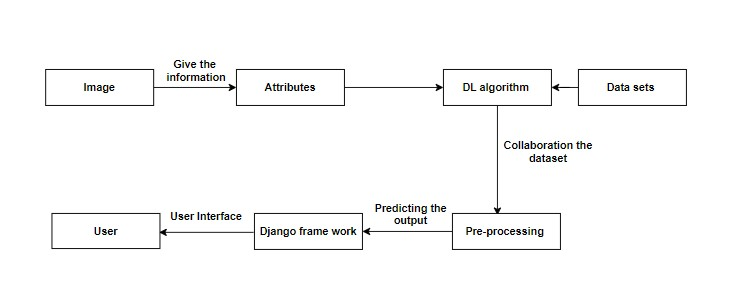
Sequence diagrams model the flow of logic within your system in a visual manner, enabling you both to document and validate your logic, and are commonly used for both analysis and design purposes. Sequence diagrams are the most popular UML artifact for dynamic modelling, which focuses on identifying the behaviour within your system. Other dynamic modelling techniques include [activity diagramming](http://agilemodeling.com/artifacts/activityDiagram.htm), [communication diagramming](http://agilemodeling.com/artifacts/communicationDiagram.htm), [timing diagramming](http://agilemodeling.com/artifacts/timingDiagram.htm), and [interaction overview diagramming](http://agilemodeling.com/artifacts/interactionOverviewDiagram.htm). Sequence diagrams, along with [class diagrams](http://agilemodeling.com/artifacts/classDiagram.htm) and [physical data models](http://agiledata.org/essays/dataModeling101.html) are in my opinion the most important design-level models for modern business application development.

**ER DIAGRAM:**



An entity relationship diagram (ERD), also known as an entity relationship model, is a graphical representation of an information system that depicts the relationships among people, objects, places, concepts or events within that system. An ERD is a [data modeling](https://searchdatamanagement.techtarget.com/definition/data-modeling) technique that can help define business processes and be used as the foundation for a [relational database](https://searchdatamanagement.techtarget.com/definition/relational-database). Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. After a relational database is rolled out, an ERD can still serve as a referral point, should any debugging or business process re-engineering be needed later.

**COLLABORATION DIAGRAM:**



A collaboration diagram show the objects and relationships involved in an interaction, and the sequence of messages exchanged among the objects during the interaction. The collaboration diagram can be a decomposition of a class, class diagram, or part of a class diagram.it can be the decomposition of a use case, use case diagram, or part of a use case diagram. The collaboration diagram shows messages being sent between classes and object (instances). A diagram is created for each system operation that relates to the current development cycle (iteration).

**MODULE DESCRIPTION:**

**IMPORT THE GIVEN IMAGE FROM DATASET:**

We have to import our data set using keras preprocessing image data generator function also we create size, rescale, range, zoom range, horizontal flip. Then we import our image dataset from folder through the data generator function. Here we set train, test, and validation also we set target size, batch size and class-mode from this function we have to train using our own created network by adding layers of CNN.

**TO TRAIN THE MODULE BY GIVEN IMAGE DATASET:**

To train our dataset using classifier and fit generator function also we make training steps per epoch’s then total number of epochs, validation data and validation steps using this data we can train our dataset.

**WORKING PROCESS OF LAYERS IN CNN MODEL:**

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics. The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. Their network consists of four layers with 1,024 input units, 256 units in the first hidden layer, eight units in the second hidden layer, and two output units.

Input Layer:

Input layer in CNN contain image data. Image data is represented by three dimensional matrixes. It needs to reshape it into a single column. Suppose you have image of dimension 28 x 28 =784, it need to convert it into 784 x 1 before feeding into input.

## **Convo Layer:**

Convo layer is sometimes called feature extractor layer because features of the image are get extracted within this layer. First of all, a part of image is connected to Convo layer to perform convolution operation as we saw earlier and calculating the dot product between receptive field (it is a local region of the input image that has the same size as that of filter) and the filter. Result of the operation is single integer of the output volume. Then the filter over the next receptive field of the same input image by a Stride and do the same operation again. It will repeat the same process again and again until it goes through the whole image. The output will be the input for the next layer.

**Pooling Layer:**

Pooling layer is used to reduce the spatial volume of input image after convolution. It is used between two convolution layers. If it applies FC after Convo layer without applying pooling or max pooling, then it will be computationally expensive. So, the max pooling is only way to reduce the spatial volume of input image. It has applied max pooling in single depth slice with Stride of 2. It can observe the 4 x 4 dimension input is reducing to 2 x 2 dimensions.

## **Fully Connected Layer (FC):**

Fully connected layer involves weights, biases, and neurons. It connects neurons in one layer to neurons in another layer. It is used to classify images between different categories by training.

## **Softmax / Logistic Layer:**

Softmax or Logistic layer is the last layer of CNN. It resides at the end of FC layer. Logistic is used for binary classification and softmax is for multi-classification.

## **Output Layer:**

Output layer contains the label which is in the form of one-hot encoded. Now you have a good understanding of CNN.

**CLASSIFICATION IDENTIFICATION:**

We give input image using keras preprocessing package. That input Image converted into array value using pillow and image to array function package. We have already classified disease of pneumothorax in our dataset. It classifies what are the pneumothorax disease. Then we have to predict our pneumothorax diseases using predict function.

Given dataset

Pneumothorax disease Classification

Feature Extractions

CNN Model

Input image

The pneumothorax disease recognition method is based on a two-channel architecture that is able to recognize disease of pneumothorax. The pneumothorax disease parts are cropped and extracted and then used as the input into the inception layer of the CNN. The Training phase involves the feature extraction and classification using convolution neural network.

**Libraries Required:**

* **tensorflow**: Just to use the tensor board to compare the loss and adam curve our result data or obtained log.
* **keras**: To pre-process the image dataset.
* **matplotlib**: To display the result of our predictive outcome.
* **os**: To access the file system to read the image from the train and test directory from our machines.

**TensorFlow:**

TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow.

TensorFlow is a software library or framework, designed by the Google team to implement machine learning and deep learning concepts in the easiest manner. It combines the computational algebra of optimization techniques for easy calculation of many mathematical expressions.

Let us now consider the following important features of TensorFlow −

* It includes a feature of that defines, optimizes and calculates mathematical expressions easily with the help of multi-dimensional arrays called tensors.
* It includes a programming support of deep neural networks and machine learning techniques.
* It includes a high scalable feature of computation with various data sets.
* TensorFlow uses GPU computing, automating management. It also includes a unique feature of optimization of same memory and the data used.

**Keras:**

Keras runs on top of open source machine libraries like TensorFlow, Theano or Cognitive Toolkit (CNTK). Theano is a python library used for fast numerical computation tasks. TensorFlow is the most famous symbolic math library used for creating neural networks and deep learning models. TensorFlow is very flexible and the primary benefit is distributed computing. CNTK is deep learning framework developed by Microsoft. It uses libraries such as Python, C#, C++ or standalone machine learning toolkits. Theano and TensorFlow are very powerful libraries but difficult to understand for creating neural networks.

Keras is based on minimal structure that provides a clean and easy way to create deep learning models based on TensorFlow or Theano. Keras is designed to quickly define deep learning models. Well, Keras is an optimal choice for deep learning applications.

## **Features**

Keras leverages various optimization techniques to make high level neural network API easier and more performant. It supports the following features −

* Consistent, simple and extensible API.
* Minimal structure - easy to achieve the result without any frills.
* It supports multiple platforms and backends.
* It is user friendly framework which runs on both CPU and GPU.
* Highly scalability of computation.

**Matplotlib:**

Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays. Matplotlib is written in Python and makes use of NumPy, the numerical mathematics extension of Python. It provides an object-oriented API that helps in embedding plots in applications using Python GUI toolkits such as PyQt, WxPythonotTkinter. It can be used in Python and IPython shells, Jupyter notebook and web application servers also.

Matplotlib has a procedural interface named the Pylab, which is designed to resemble MATLAB, a proprietary programming language developed by MathWorks. Matplotlib along with NumPy can be considered as the open source equivalent of MATLAB.

**os:**

The OS module in Python comes with various functions that enables developers to interact with the Operating system that they are currently working on. In this article we’ll be learning mainly to create and delete a directory/folder, rename a directory and even basics of file handling.

Python OS module provides the facility to establish the interaction between the user and the operating system. It offers many useful OS functions that are used to perform OS-based tasks and get related information about operating system.

The OS comes under Python's standard utility modules. This module offers a portable way of using operating system dependent functionality.

**Conclusion:**

It focused how image from given dataset (trained dataset) and past data set used to predict the pattern of pneumothorax disease using CNN model. This brings some of the following insights about pneumothorax disease prediction. The major benefit of the CNN classification framework is the ability to classify images automatically. The pneumothorax diseases mainly contribute to misshape and often can’t be remedied because the patients are diagnosed too late with the diseases. In this study, we have discussed the overview of methodologies for detecting the abnormalities in pneumothorax images which includes collection of pneumothorax image data set, preprocessing techniques, feature extraction techniques and classification schemes.

**Future Work:**

* Medical department wants to automate the detecting of pneumothorax disease from eligibility process (real time).
* To optimize the work to implement in Artificial Intelligence environment.