FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG
FACULTY OF ENGINEERING

Seminar Automatic Question Answering Using IBM Watson

Project:

Medical Assistant(Chest X-ray Image Classifier)

By:
Johar Kanti Sarker
Arindam Chakraborty

Organization:

- Project Objective
- Training of Classifier
- Creation of Chat bot
- Create Web Infrastructure
- Create Web based User Interface
- Fuse the classifier and Chat bot in User Interface

Project Objective :

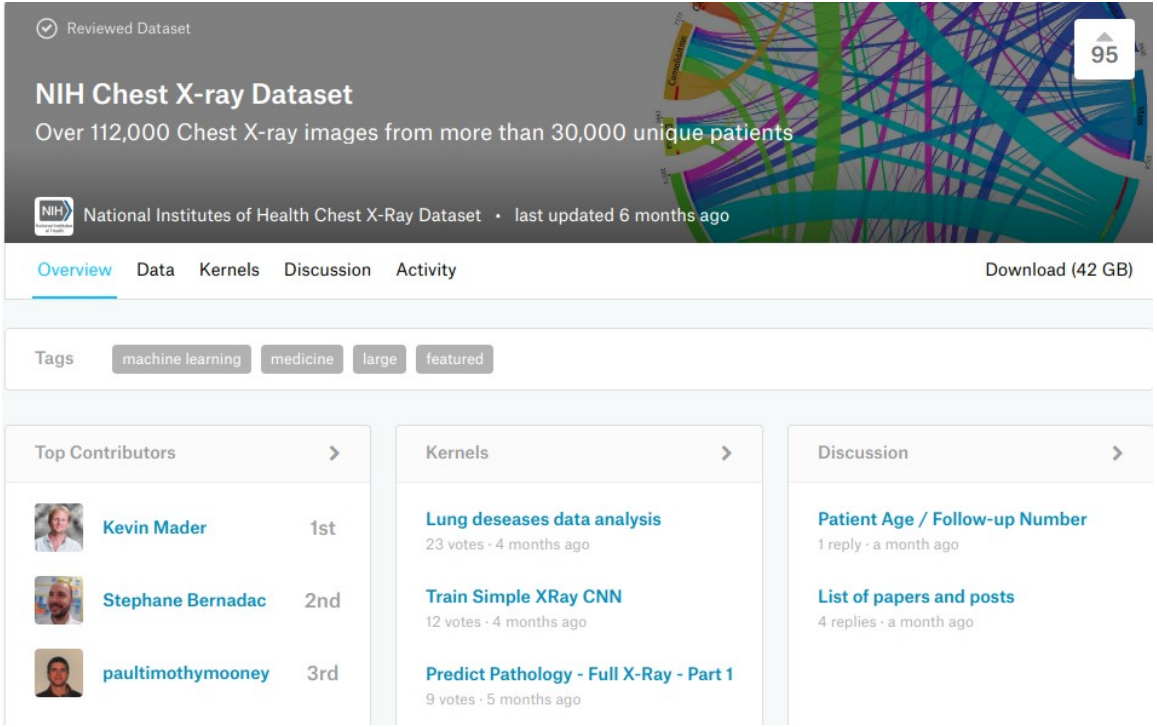
- ✓ Use Chest X-ray Image
- ✓ Detect diseases
- ✓ Integrate multiple classifier kernel with a web framework
- ✓ Provide a web based user interface for easy and mobile access
- ✓ Use as an assistant tool for Doctors

Training of Classifier

Training of Classifier

- ✓ Data set selection
- ✓ Choosing Training Data
- ✓ Train Classifier

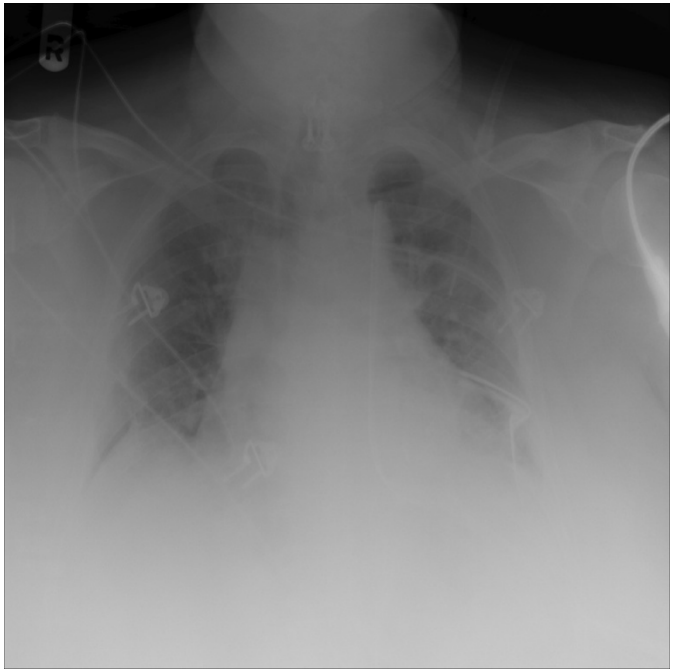
Dataset selection:



Source: https://www.kaggle.com/nih-chest-xrays/data/version/1

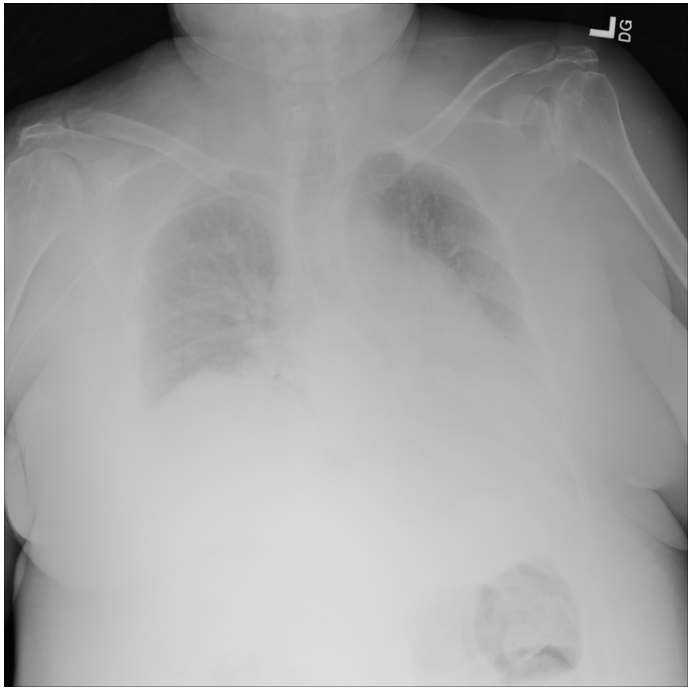
Choosing Training Data 1/5:

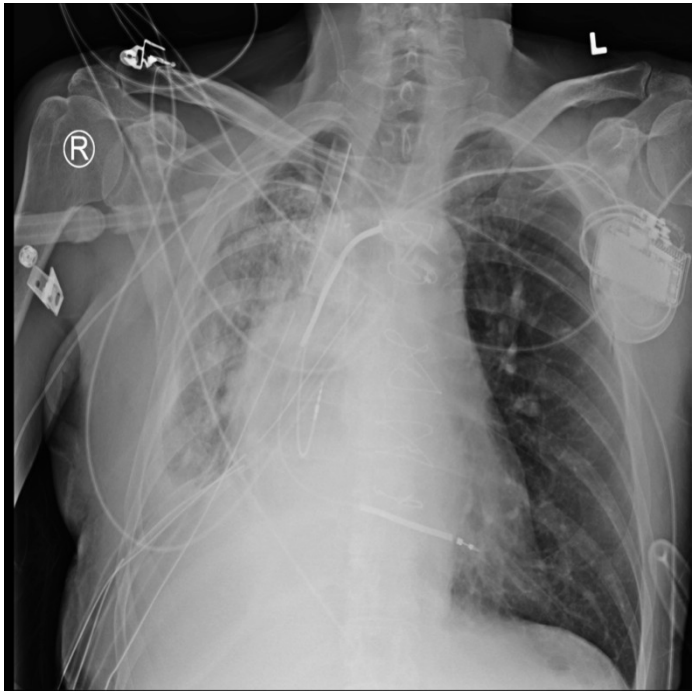
With huge data, comes huge Problem



Choosing Training Data 2/5:

With huge data, comes huge problem

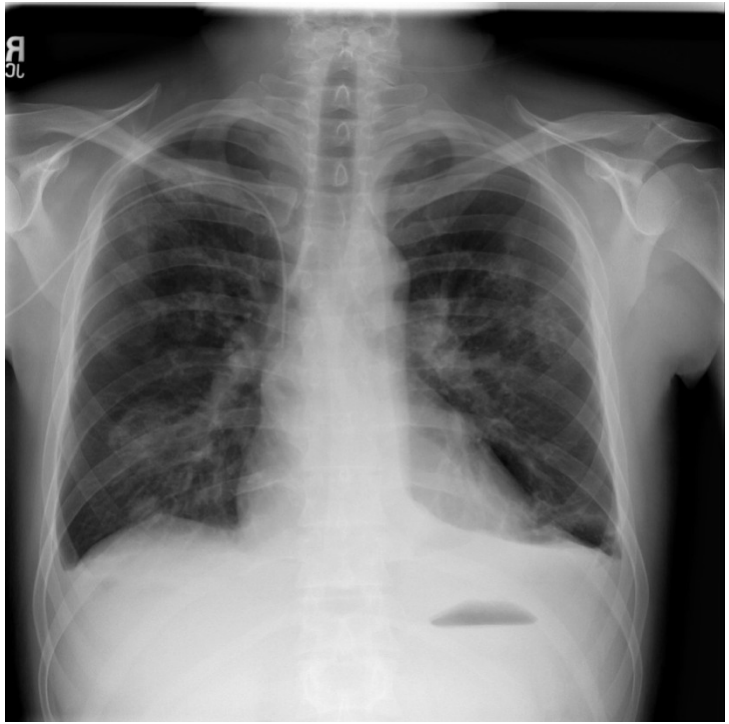




Choosing Training Data 3/5:

Hand picked images

- with better contrast
- Images having visually better feature



Choosing Training Data 4/5:

- 112,000 images
- 14 classes

Selected diseases:

Pneumonia	Lung Nodule	Pleural Effusion	Lung Hernia
Common	Common	Common	Rare

Classes:

Pneumonia

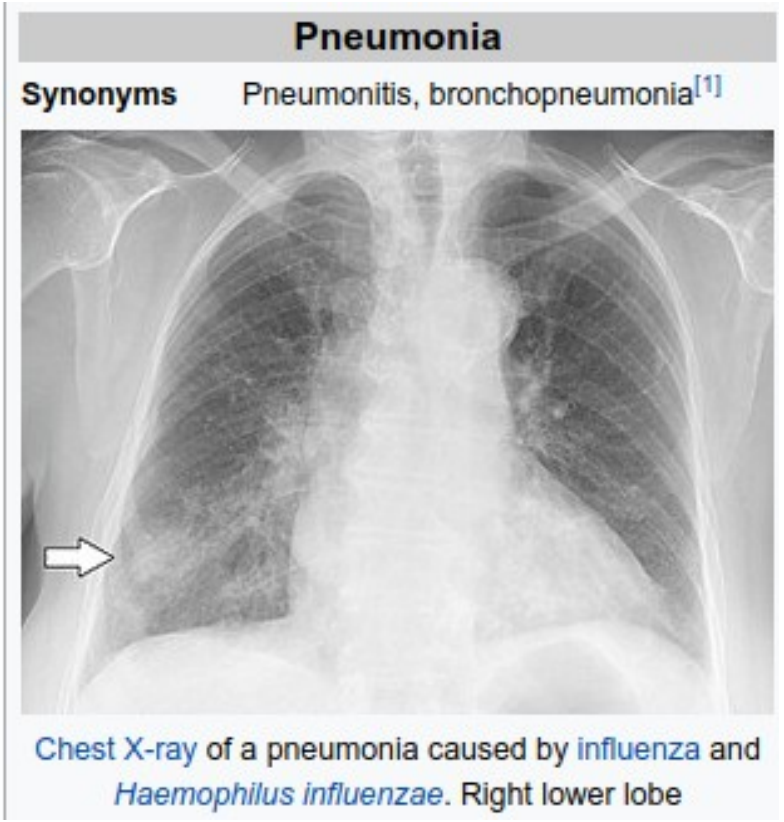
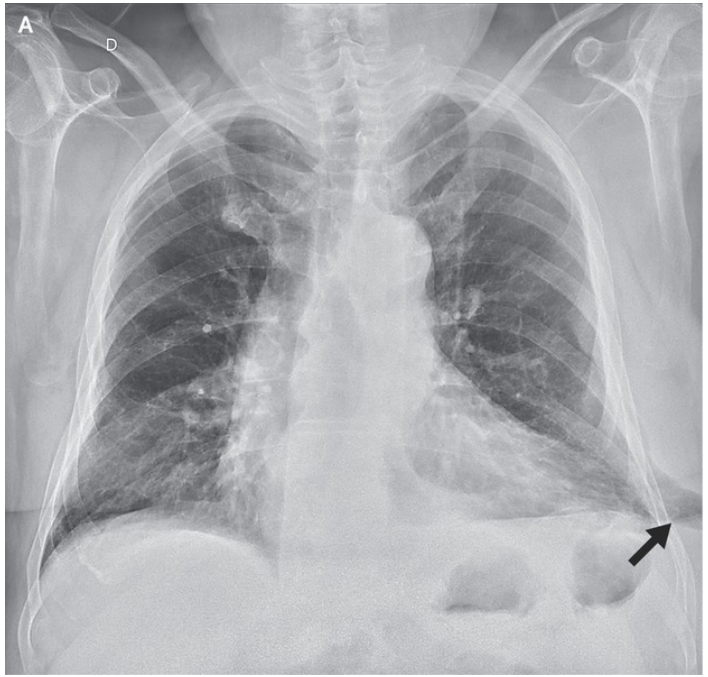


Image Source: Wikipedia

Classes:

Lung Hernia



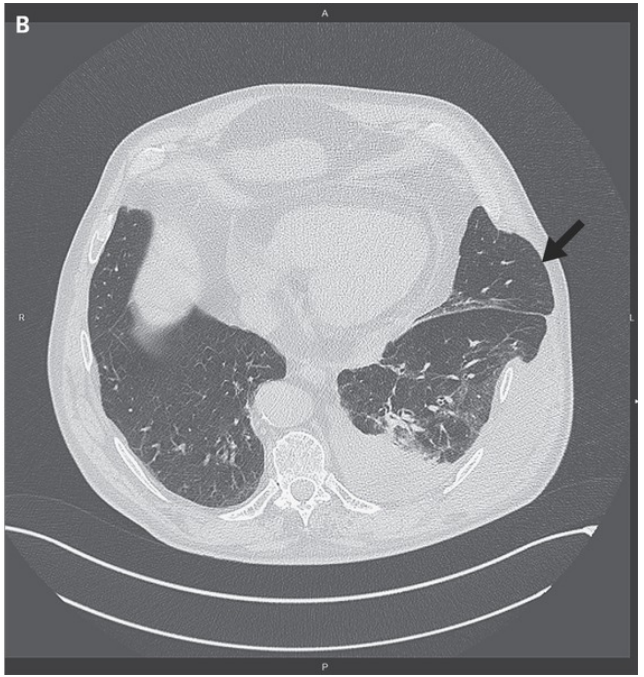


Image Source:
https://www.urmc.rochester.edu/

Classes:

Lung Nodule

Image Source: https://www.med-ed.virginia.edu/

Classes:

Pleural Effusion →

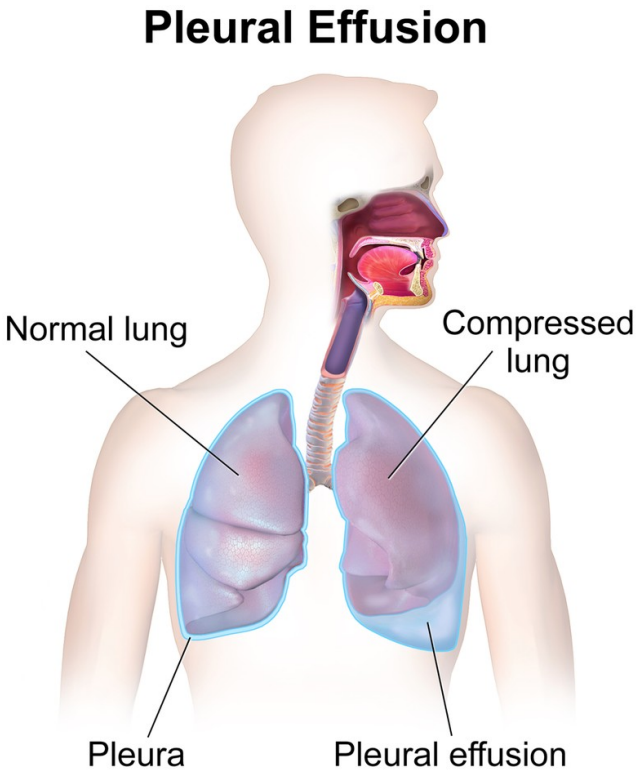


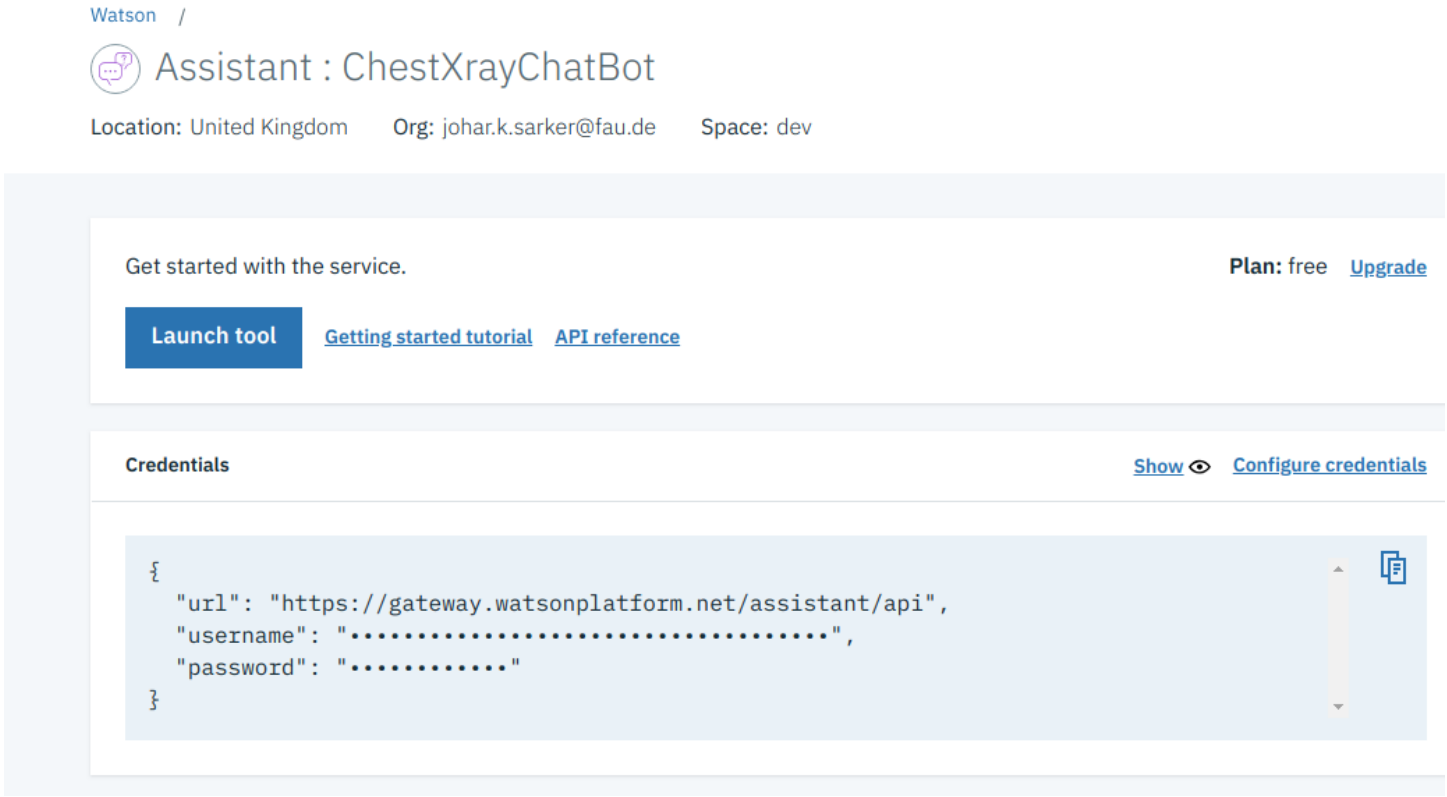
Image Source: Wikipedia

Choosing Training Data 5/5:

Pneumonia	271
Nodule	200
Effusion	200
Hernia	200
Total=	871

Creation of Chat bot

Creation of Chatbot:



Creation of Chat bot:

For each of the classes we have created :

- Intent, Entity, dialogue
- Each entity has values:
- definition,causes, symptoms, diagnosis and treatment
- Dialogue nodes contains dialogues for each of the classes along with greetings node

Creation of Chat bot:

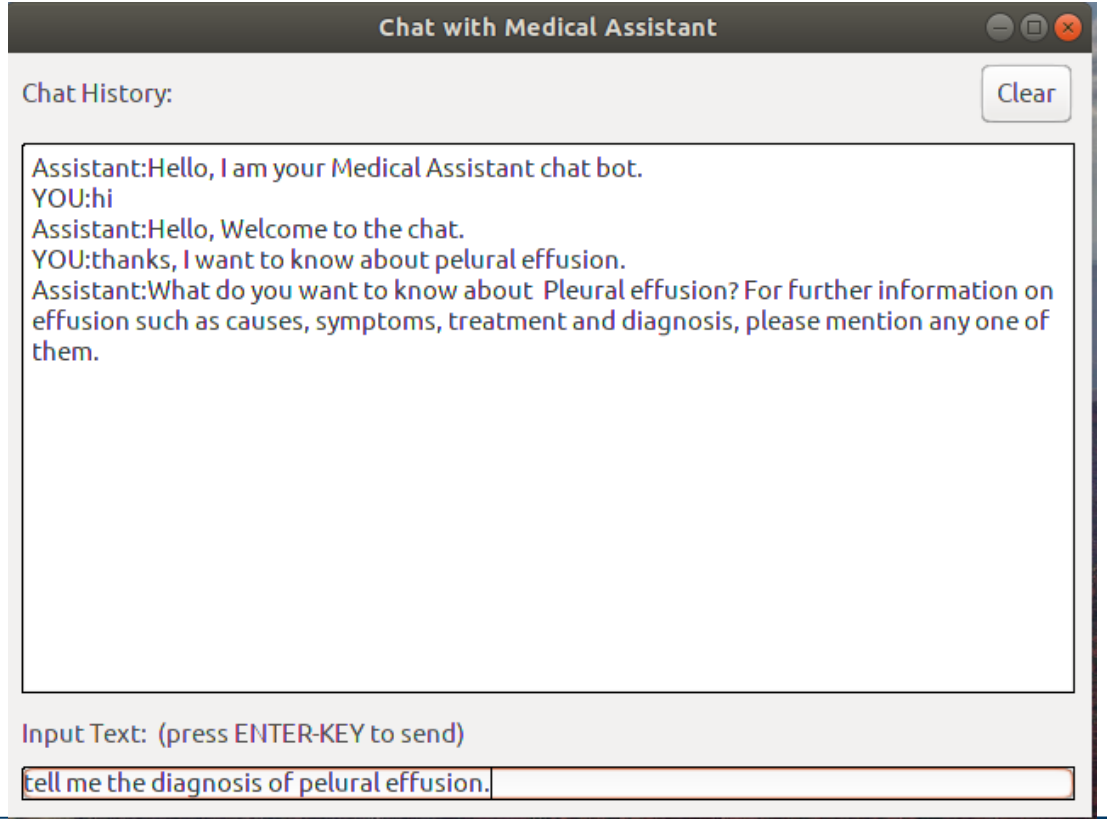
- The chat bot interface is developed in Java
- We created a .jar file from the class and then executed anywhere we needed such as inside python or Java script

WatsonChat.java

```
34
35 @SuppressWarnings("serial")
36 public class WatsonChat extends JFrame {
37
38     //---------------------------------------------
39     private static final String WATSON_CONVERSATION_USERNAME = "62ee3d3a-b3f8-4a01-b85b-97c0358b1dc7";
40     private static final String WATSON_CONVERSATION_PASSWORD = "za8u4Xld2Cqb";
41     private static final String WATSON_CONVERSATION_WORKSPACE_ID = "24e99336-59ef-4602-9cb8-c11fcbb1ae87";
42     //---------------------------------------------
43
44     private static final String WCS_CLIENT_ID = "user_id_johar";
45
46     private static final int WIDTH_PX =  640;
47     private static final int HEIGHT_PX = 480;
48
49     private final WcsClient mWatson = new WcsClient(
50             WATSON_CONVERSATION_USERNAME,
51             WATSON_CONVERSATION_PASSWORD,
52             WATSON_CONVERSATION_WORKSPACE_ID);
53
54     private final ExecutorService mExecutor = Executors.newSingleThreadExecutor();
55     private final StringBuilder mSb = new StringBuilder();
56
57     private final JTextArea mTextArea = new JTextArea();
58     private final JTextField mTextBox = new JTextField(""); //client text box
```

Creation of Chat bot:

- Chat bot in action
- We also used 3rd party Chatlio service for web integration



Creating Web Application Infrastructure

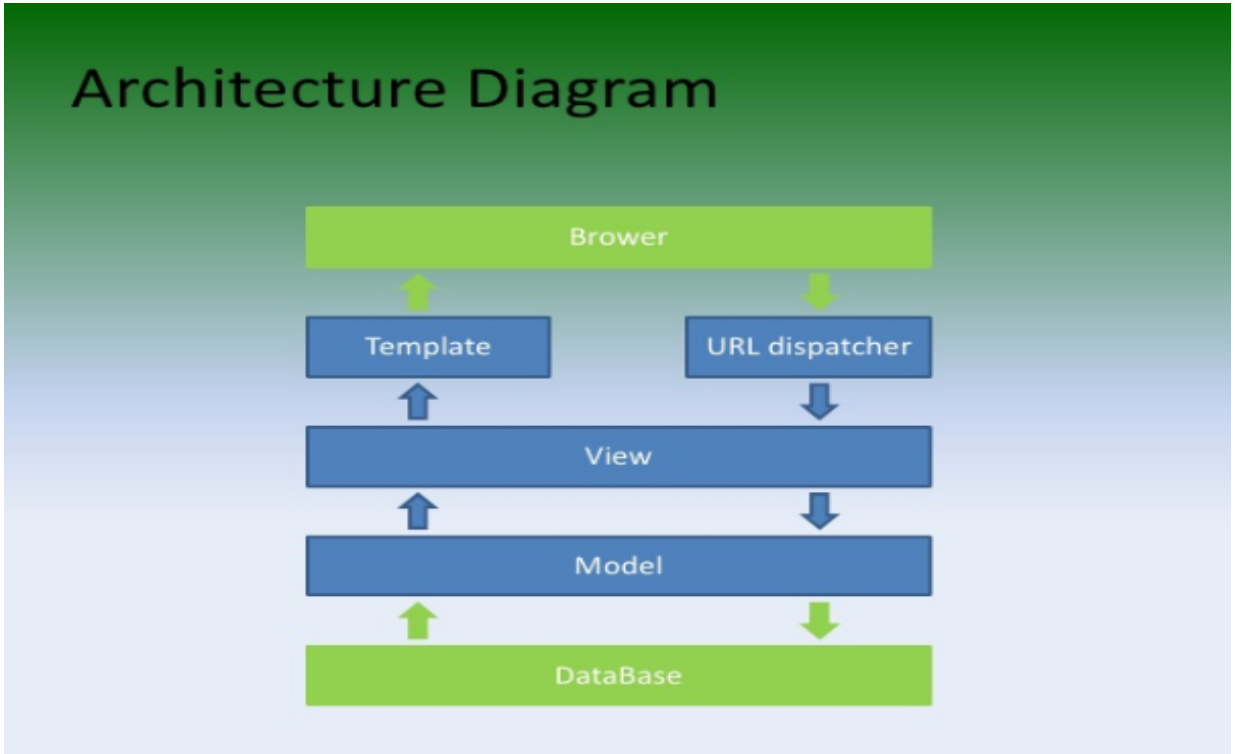
For creating Web Application Infrastructure

- Web Framework (Django)
- Architecture Pattern (MTV)
- configure for classifier integration
- configure for server communication
- Web User Interface for user interaction

Django Web Framework

- A high-level Python Web framework
- Can be focused separately on:
  - business logic and presentation layer
  - Focus on automation
- Widely supported with many deployment options

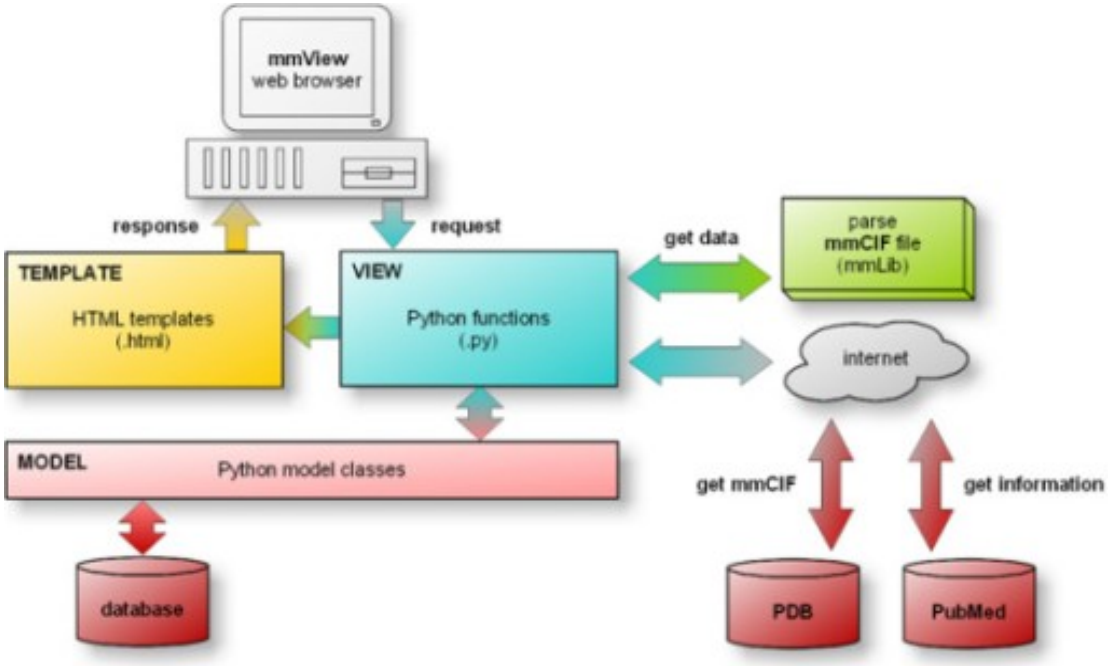
Django Web Framework



MTV Architecture

- MTV stands for Model Template View
- Django Components
  - Model – Django ORM
  - Templates – Django Template Engine (HTML)
  - Views – Python function

MTV Architecture



Deploying Classifier kernel

- Put the classifier packages in the same folder with static folder
- Multiple classifier packages can be deployed
- Configure and Map static and Media directory in settings.py
- Create model files and folder to store image data
- Create interface file(*_upd.py) to interact with the classifier
- Create view files(tfExec.py) to handle request and response
- Map the view resource with the url pattern in urls.py

Web User Interface

- Create static pages using HTML
- HTML, CSS and other static resources in static folder
- The view generates the dynamic components
- User uploads the image through the UI
- Provide facilities for image cropping of selected image to discard unnecessary data for more accurate classification
- Map url pattern with static html as template view in urls.py

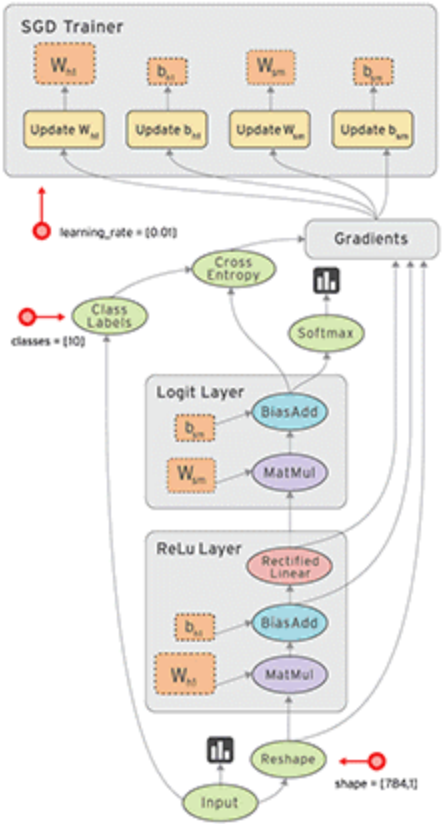
Technologies used

- ✓ Python – For view and core logic
- ✓ HTML – For static view
- ✓ Jquery – For presentation layer logic
- ✓ Python Image Library(PIL) – For preprocessing of image
- ✓ Tensorflow library – For preparing image tensors for classifiers
- ✓ Used a plug-in ‘Chatlio’ for js based chatbot integration

Classifiers : MobileNet Vs Inception

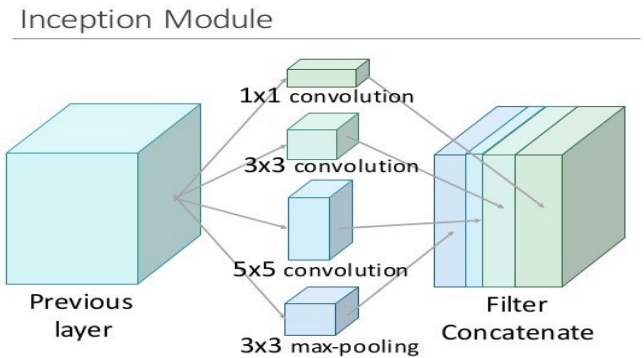
TensorFlow:

- an open source software library for numerical computation using dataflow graphs.
- Nodes in the graph represents mathematical operations, while graph edges represent multi-dimensional data arrays (aka tensors) communicated between them.
- The flexible architecture allows to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API



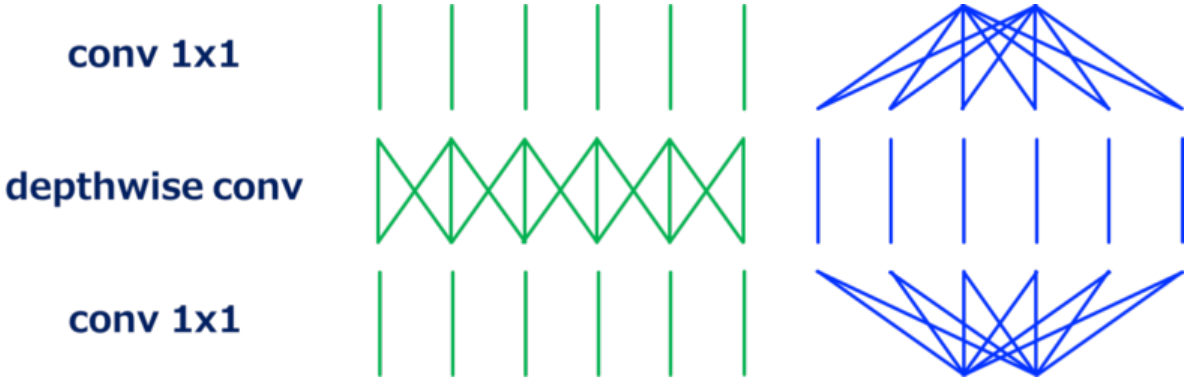
Classifiers : Inception

- A deep convolutional neural network architecture
- Hallmark of this architecture is the improved utilization of the computing resources inside the network
- This was achieved by a carefully crafted design that allows for increasing the depth and width of the network while keeping the computational budget constant.



Classifiers : Mobilenet

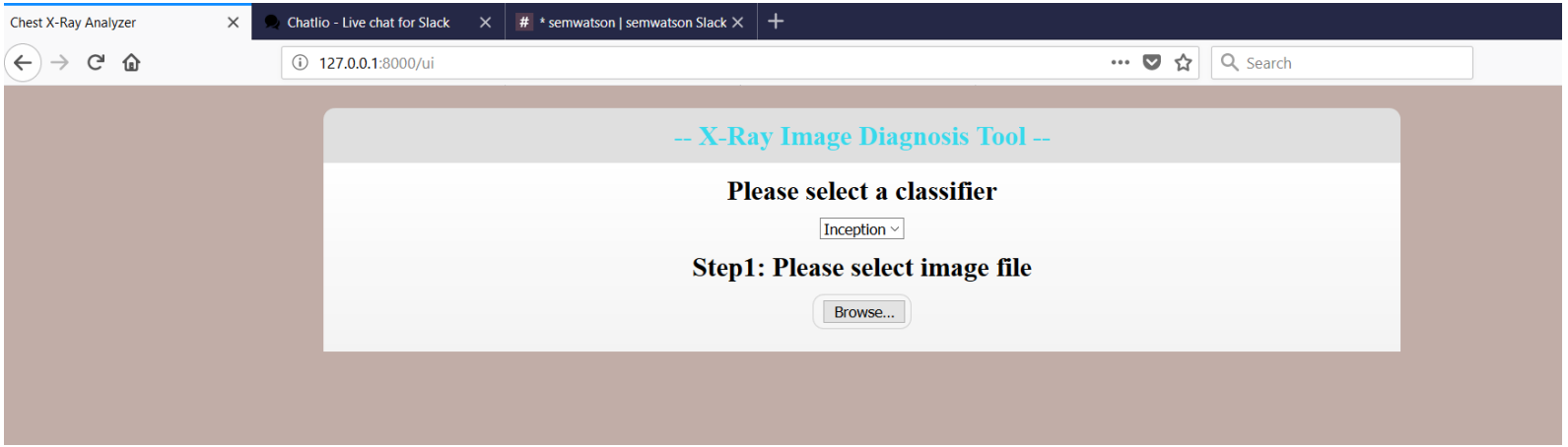
- MobileNet V2 is a family of neural network architecture
- Efficient on- device image classification and related tasks
- Originally published by Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen: "Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation", 2018.



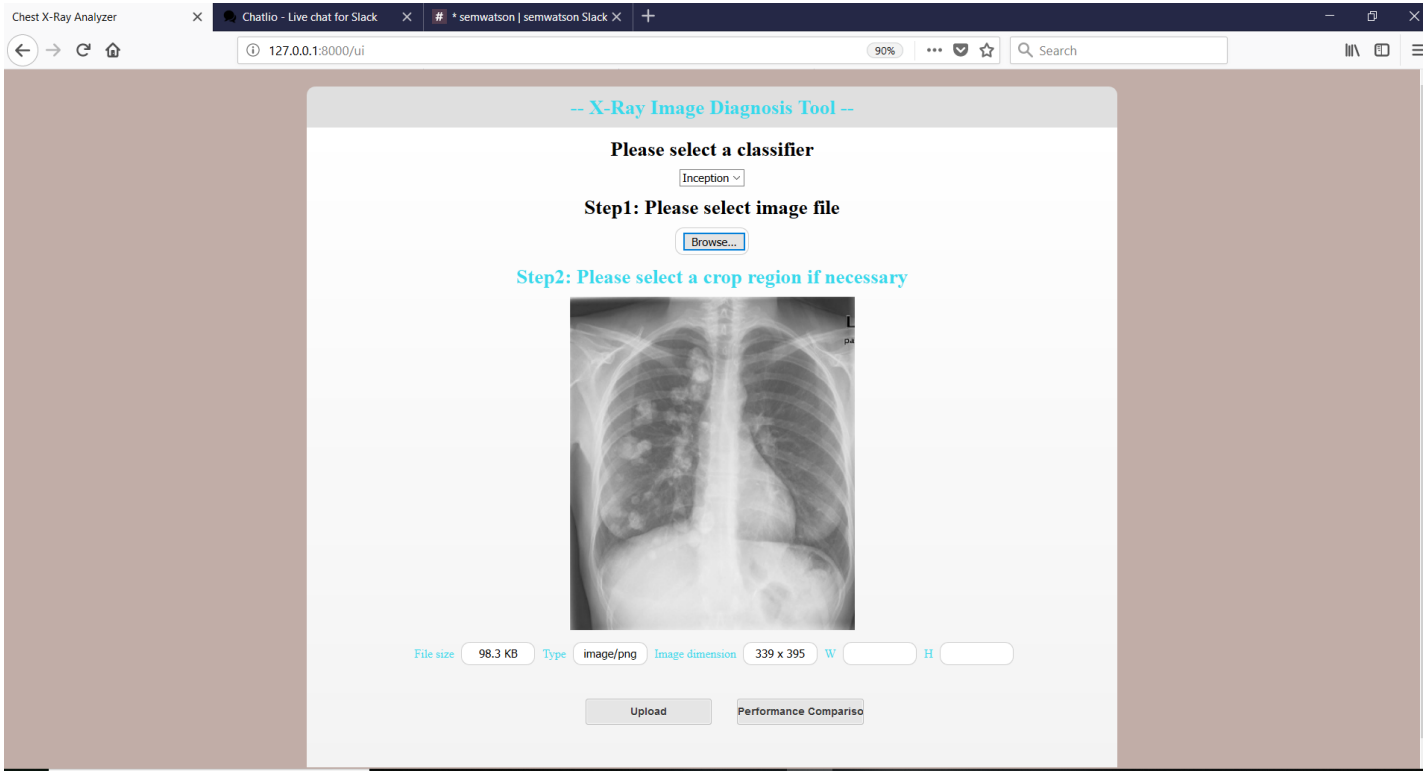
Which Model : Inception vs Mobilenet

- Performance of MobileNet model relative to Inception V3 is usually lower since MobileNet is optimized for speed whereas Inception aims for correctness.
- MobileNet has a much lower model size compared to Inception model
- MobileNet training accuracy can sometimes be better in case of reduced training sets
- MobileNet has shorter prediction time compared to Inception

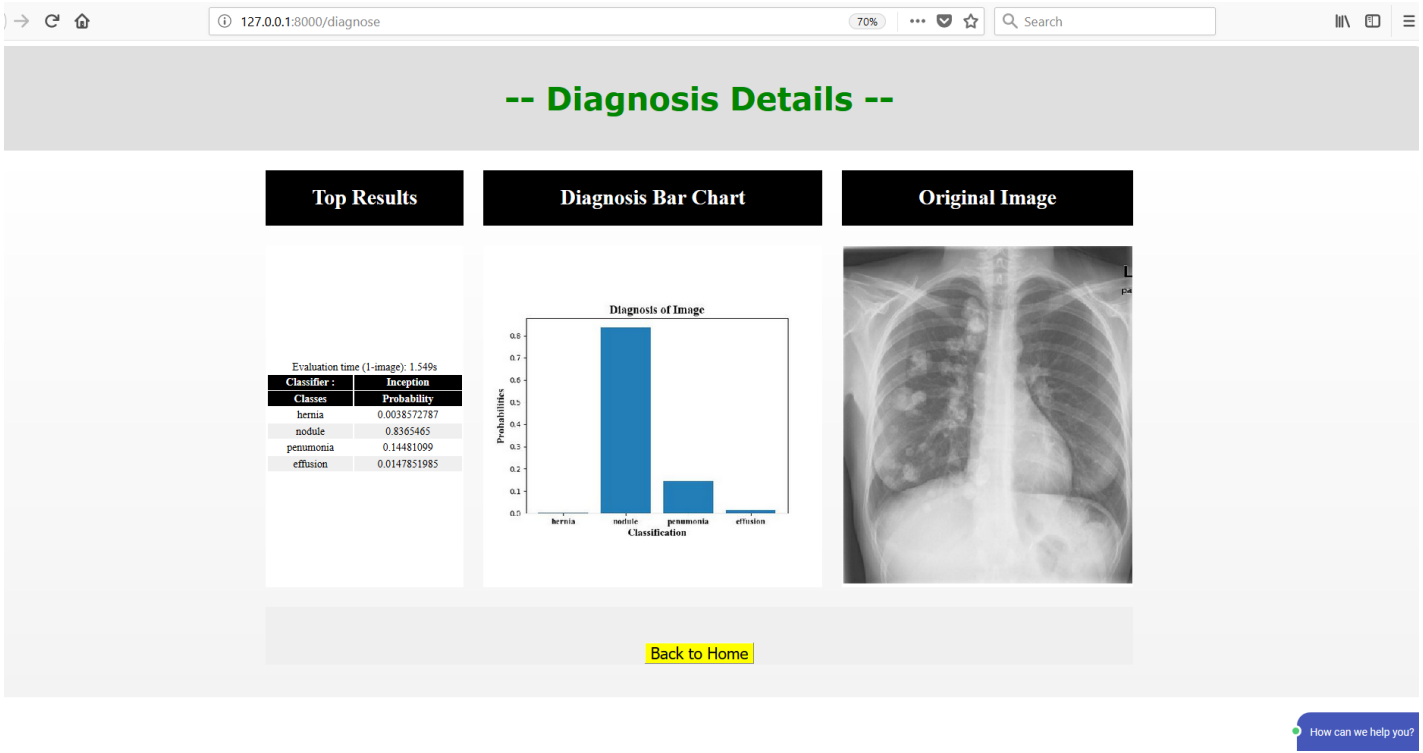
Web User Interface: select file



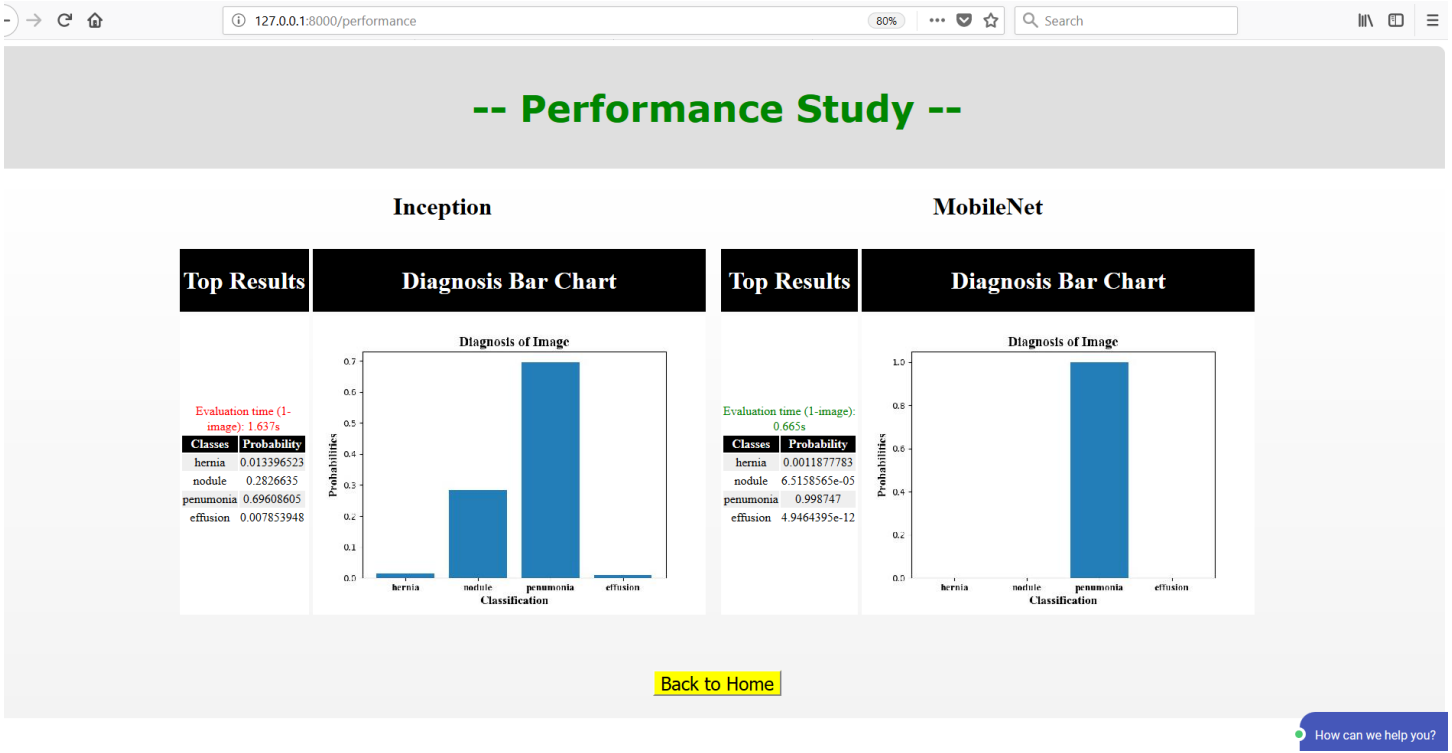
Web User Interface: crop & upload for classification



Web User Interface: using single classifier



Web User Interface: performance comparison



Future Work:

- Add more classes
- Add image pre-processing facility
- Improve GUI and Chat-bot
- Better classification model
- Add heat map

Thank You :)

Questions ?