**PNEUMONIA DETECTION USING CHEST X RAYS USING DEEP LEARNING**

A Mini Project report submitted to   
**JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD**In partial fulfilment of the requirements for the award of the degree of  **BACHELOR OF TECHNOLOGY**In **COMPUTER SCIENCE AND ENGINEERING**

Submitted By

**KANDAGATLA SOWMYA 18UK1A05K3**

**KOTRA VINAY REDDY 18UK1A05E9**

**MOHAMMED AZHARUDDIN 18UK1A05J1**

**SAMA NAIMISHA 18UK1A05M1**

Under the guidance of **Mr. SHIVA SAI PRASAD**   
Assistant Professor  
  
**A picture containing text

Description automatically generated**  
**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
VAAGDEVI ENGINEERING COLLEGE**  
Affiliated to JNTUH, HYDERABAD  
BOLLIKUNTA, WARANGAL (T.S) – 506005

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**VAAGDEVI ENGINEERING COLLEGE**

**WARANGAL**

**A picture containing text

Description automatically generated**

**CERTIFICATE**

This is to certify that the Mini project report entitled " **PNEUMONIA DETECTION USING CHEST X RAYS USING DEEP LEARNING**" is being submitted by **K. SOWMAYA (18UK1A05K3), K. VINAY (18UK1A05E9), MD. AZHARUDDIN (18UK1A05J1), S. NAIMISHA (18UK1A05M1)** in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2021- 2022

**Project Guide HOD**

**MR. SHIVA SAI PRASAD Dr. R. NAVEEN KUMAR**

**External**

**ACKNOWLEDGEMENT**

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr. P. Prasad Rao**, Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this mini project in the institute.

We extend our heartfelt thanks to **Dr. R. Naveen Kumar**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the mini project.

We express heartfelt thanks to the Mini Project Coordinator**, Dr. J. Srikanth**, Assistant Professor, Department of CSE for his constant support and giving necessary guidance for completion of this mini project.

We express heartfelt thanks to the guide**, Mr. Shiva Sai Prasad** Assistant Professor**,** Department of CSE for his constant support and giving necessary guidance for completion of this mini project.

Finally, we express our sincere thanks and gratitude to our family members, friends for their encouragement and outpouring their knowledge and experiencing throughout thesis.

**Abstract:**

Pneumonia is a fatal disease that majorly affects the elderly and may sometimes prove to be life threatening. Early diagnosis of Pneumonia gains a paramount importance for saving many human lives. This paper aims at the detection and classification of patients affected by Pneumonia based on their chest X-rays. A convolutional neural network is employed from scratch to make the above diagnosis and yield highly accurate results. Deep Learning models automate the process and ensure speedy, adroit, and adept results when provided with X-rays of patients. The classification occurs after the image is fed through a series of convolutional and max pooling layers that are activated by using the ReLU activation function that is subsequently fed into the neurons present in the dense layers and finally, the output neuron is activated by the sigmoidal function. The accuracy increases as the model trains and decreases the loss simultaneously. Overfitting is prevented by implementing data augmentation before fitting the model. Thus, efficient, and cogent results are obtained by the proposed deep learning models to classify the chest X-rays for the detection of pneumonia.

**CONTENTS**

**LIST OF CHAPTERS**

**1. INTRODUCTION……………………………………………………...9-11**

1.1 Motivation……………………………………………………………………...9

1.2 Problem definition……………………………………………………………..10

1.3 Objective of Project……………………………………………………………10

1.4 Limitations of Project………………………………………………………….11

**2. LITERATURE SURVEY……………………………………………...12-15**

2. I Introduction……………………………………………………………………12

2.2 Existing System………………………………………………………………..12

2.3 Proposed System………………………………………………………………13

2.3.1 proposed architecture……………………………………………..13

2.3.2 proposed solution…………………………………………………14

**3. ANALYSIS……………………………………………………………..16-19**

3.1 Introduction…………………………………………………………………….16

3.2 Software Requirement Specifications………………………………………….16

3.2.1 Software requirement ………………………………………16

3.2.1 Hardware requirement……………………………………....17

3.3 Content diagram of Project ……………………………………………………17

3.4 Algorithms and Flowcharts…………………………………………………….18

3.5 Conclusion……………………………………………………………………..19

**4. DESIGN………………………………………………………………...20-25**

* 1. Introduction……………………………………………………………………20
  2. UML diagram………………………………………………………………….23

4.2.1 use case diagram…………………………………………………………23

* 1. Project architecture…………………………………………………………….24
  2. Model Design………………………………………………………………….25

4.4.1 Project Flow…………………………………………………………….25

**5.CODE SNIPPETS……………………………………………………..26-34**

5.1 Model Code…………………………………………………………………….26

5.2 base.html………………………………………………………………………..29

5.3 CSS……………………………………………………………………………..31

5.4 JS……………………………………………………………………………….32

5.5 APP.py…………………………………………………………………………33

5.6 Conclusion……………………………………………………………………..34

**6. IMPLEMENTATION………………………………………………..35-41**

6.1 Introduction…………………………………………………………………..35

6.2 Explanation of Key functions………………………………………………...35

**7. APPLICATION……………………………………………………….42**

**8. ADVANTAGES……………………………………………………….42**

**9. FUTURE SCOPE……………………………………………………..42**

**10. CONCLUSION………………………………………………………42**

**11. BIBILIOGRAPHY…………………………………………………..43**

**LIST OF FIGURES** **PAGENO.**

Figure 1: Classification of Pneumonia 10

Figure 2: CNN Model 13

Figure 3: The main function code of the CNN model. 14

Figure 4: DenseNet 15

Figure 5: Fully Connected layer 17

Figure 6: Flow chart 18

Figure 7: Algorithm 18

Figure 8: Flow chart 19

Figure 9: Max Pooling 21

Figure 10: DenseNet 22

Figure 11: Fully Connected layer 22

Figure 12: Architecture of CNN 23

Figure 13: Use case diagram 23

Figure 14: Flow chart 24

Figure 15: Architecture 24

Figure 16: .ipynb describing libraries importing and images preprocessing 26

Figure 17: .ipynb initialize the model and adding convolution layer 26

Figure 18: .ipynb for adding flatten layer and hidden layer 27

Figure 19: .ipynb for adding output layer and Model training 27

Figure 20: .ipynb for saving the Model 28

Figure 21: .ipynb for Prediction 28

Figure 22: .ipynb for Prediction 29

Figure 23: base.html for describing the webpage styling 29

Figure 24: base.html describing the body of webpage 30

Figure 25: base.html describing the body of webpage 30

Figure 26: CSS code for styling the webpage 31

Figure 27: CSS code for styling the webpage 31

Figure 28: JS code 32

Figure 29: JS code 32

Figure 30: Python code 33

Figure 31: Python code 33

Figure 32: Output predicting that person is safe 34

Figure 33: Output predicting that person is affected by pneumonia 34

Figure 34: Sigmoid function 35

Figure 35: Rectified linear unit 36

Figure 36: Libraries 37

Figure 37: Image preprocessing 38

Figure 38: Model initialize and adding convolution layer 38

Figure 39: key functions 39

Figure 40: Model training 40

Figure 41: Model evaluation 41

Figure 42: Model saving 41

1. **INTRODUCTION**
   1. **MOTIVATION**

A Deep Learning based model used for the prediction whether a person is suffering from pneumonia or not. The project is absed upon the Standard **Convolutional Based Neural Network** Architectural Implementation. It incorporates the use of CNN layers with Hyper-parameters tuning. The motivation behind the project is to effectively classify the reports of chest x-rays to classify into pneumonia or non-pneumonia cases.

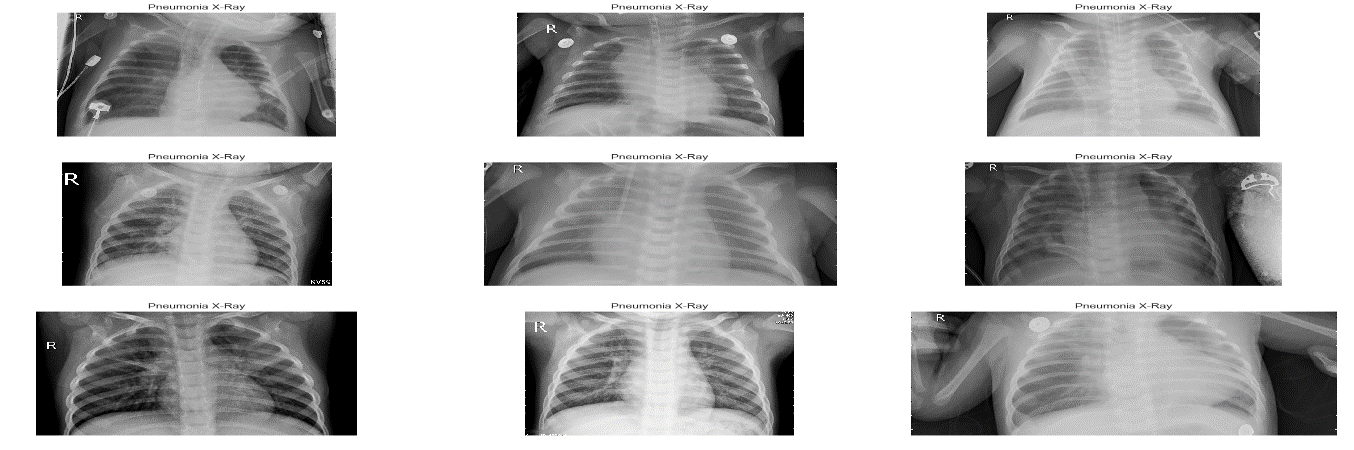
Pneumonia affects a large number of individuals, especially children, mostly in developing and underdeveloped countries characterized by risk factors such as overcrowding, poor hygienic conditions, and malnutrition, coupled with the unavailability of appropriate medical facilities. Early diagnosis of pneumonia is crucial to cure the disease completely. Examination of X-ray scans is the most common means of diagnosis, but it depends on the interpretative ability of the radiologist and frequently is not agreed upon by the radiologists. Thus, an automatic CAD system with generalizing capability is required to diagnose the disease. To the best of our knowledge, most previous methods in the literature focused on developing a single CNN model for the classification of pneumonia cases, and the use of the ensemble learning paradigm in this classification task has not been explored. However, the ensemble learning model incorporates the discriminative information from all the constituent base learners, allowing it to make superior predictions, and thus was implemented in this study. To handle the low amount of available biomedical data, transfer learning models were used as base learners, the decision scores of which were ensembled.

# **WHAT IS PNEUMONIA?**

Pneumonia is an infection that inflames the air sacs in one or both lungs. The air sacs may fill with fluid or pus (purulent material), causing cough with phlegm or pus, fever, chills, and difficulty breathing. A variety of organisms, including bacteria, viruses and fungi, can cause pneumonia. Chest X-ray, blood tests, and culture of the sputum may help confirm the diagnosis. The disease may be classified by where it was acquired, such as community- or hospital-acquired or healthcare-associated pneumonia.

Pneumonia is a form of an acute respiratory infection that affects the lungs. The lungs are made up of small sacs called alveoli, which fill with air when a healthy person breathes. When an individual has pneumonia, the alveoli are filled with pus and fluid, which makes breathing painful and limits oxygen intake.

Pneumonia is the single largest infectious cause of death in children worldwide. Pneumonia killed 808 694 children under the age of 5 in 2017, accounting for 15% of all deaths of children under five years old. Pneumonia affects children and families everywhere but is most prevalent in South Asia and sub-Saharan Africa.



**FIGURE 1: Classification of Pneumonia**

* 1. **PROBLEM DEFINATION**

Our Problem Statement deals with "Deep Learning Techniques for Pneumonia Detection using Python". It states/defines that predicting whether the person is having Pneumonia or not using deep learning techniques by convolution neural networks algorithm. It demonstrates how deep learning technology can be used for the detection of Pneumonia through images using Python. This statement clearly explains about whether a person is affected by Pneumonia or not.

A Convolution Neural Network can take in an input image, assign importance (leanable weights and biases) to various aspects/objects in the image and be able to differentiate one from other i.e., PNEUMONIA AND NORMAL images

* 1. **OBJECTIVE OF THE PROJECT**

The recent progress in deep learning has contributed to unprecedented improvements in computer vision. Convolution neural network (CNN) is one of the most recognized deep-learning algorithms for its wide application in image classification, segmentation, and detection. Therefore, in this literature, CNN is proposed to perform Pneumonia Detection.

The main objective of this project is to help the doctors to predict the pneumonia disease more accurately using a deep learning model. The objective is not only to help the doctors but also to the patients to precisely predict pneumonia.

* 1. **Drawbacks of previous problem, solution to this problem**

1. Diagnosis manifests as an area(s) of increased opacity on a chest radiograph (CXR) which needs to be reviewed by highly trained specialists.
2. Although very common and curable, accurately diagnosing pneumonia is difficult
3. Challenges faced in diagnosis using CXR are-

* Lung conditions: fluid overload (pulmonary edema), bleeding, volume loss (atelectasis/collapse), lung cancer, or post-radiation.
* Outside of the lungs, fluid in the pleural space (pleural effusion) also appears as increased opacity on CXR.
* Positioning of the patient and depth of inspiration High workload on radiologist.

Thus, there is a requirement for a faster method to predict pneumonia using a CXR.

So, this project will be very useful to automatically locate lung opacities on chest radiographs and predict if the patient is suffering from pneumonia through a user friendly web app.

User may upload the images of chest x ray and simply submit them. The model will accurately predict pneumonia and return the result in 5 seconds The suitability of this deep learning model for clinical use has not been validated! The neural network achieved accuracy exceeding 90% on the relevant test set of more than 5000 x-rays. Please note that the predictive accuracy of the model, depends on the quality and quantity of the training data. Therefore, the model has limitations and cannot be used for definite clinical predictions! Instead, it can be used for research purposes and for comparison with another similar model.

1. **LITERATURE SURVEY**
   1. **Introduction**

Pneumonia is a common disease caused by different microbial species such as bacteria, virus, and fungi. The word “Pneumonia” comes from the Greek word “Pneumon” which translates to the lungs. Thus, the word pneumonia is associated with lung disease. In medical terms, pneumonia is a disease that causes inflammation of either one or both lung parenchyma [[1](https://link.springer.com/article/10.1007/s12559-020-09787-5#ref-CR1)]. However, other causes of pneumonia include food aspiration and exposure to chemicals. Based on infection, pneumonia occurs as a result of inflammation caused by pathogens which lead the lung’s alveoli to fill up with fluid or pus and thereby leading to decrease of carbon dioxide (CO2) and oxygen (O2) exchange between blood and the lungs, making it hard for infected persons to breathe. Some of the symptoms of pneumonia include the shortness of breath, fever, cough, chest pain, etc. Moreover, the people at risk of pneumonia are elderly people (above 65 years), children (below the age of 5 years), and people with other complications such as HIV/AIDS, diabetes, chronic respiratory diseases, cardiovascular diseases, cancer, hepatic disease, etc. [[2](https://link.springer.com/article/10.1007/s12559-020-09787-5#ref-CR2),[3](https://link.springer.com/article/10.1007/s12559-020-09787-5#ref-CR3),[4](https://link.springer.com/article/10.1007/s12559-020-09787-5#ref-CR4),[5](https://link.springer.com/article/10.1007/s12559-020-09787-5#ref-CR5)]. Table [1](https://link.springer.com/article/10.1007/s12559-020-09787-5#Tab1) presents a classification of pathogens that causes pneumonia.

* 1. **Existing System**

Pneumonia is an infection that inflames the air sacs in one or both lungs. The air sacs may fill with fluid or pus (purulent material), causing cough with phlegm or pus, fever, chills, and difficulty breathing. A variety of organisms, including bacteria, viruses and fungi, can cause pneumonia.

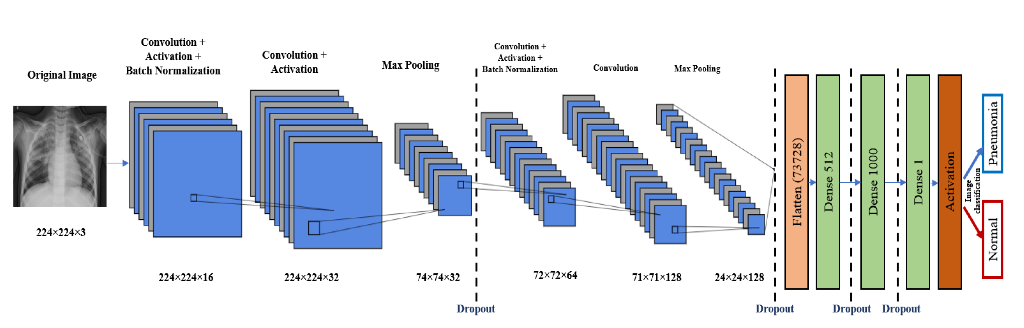
Pneumonia is an inflammatory condition of the lung affecting primarily the small air sacs known as alveoli.Symptoms typically include some combination of productive or dry cough, chest pain, fever and difficulty breathing. The severity of the condition is variable. Pneumonia is usually caused by infection with viruses or bacteria and less commonly by other microorganisms, certain medications or conditions such as autoimmune diseases.Risk factors include cystic fibrosis, chronic obstructive pulmonary disease (COPD), asthma, diabetes, heart failure, a history of smoking, a poor ability to cough such as following a stroke and a weak immune system. Diagnosis is often based on symptoms and physical examination. Chest X-ray, blood tests, and culture of the sputum may help confirm the diagnosis. The disease may be classified by where it was acquired, such as community- or hospital-acquired or healthcare-associated pneumonia.

Due to the intrinsic difficulties associated with an image, noise, and lack of appreciation by the eye, instruments have been prepared to make and improve image processing. Nowadays, Artificial Intelligence (AI), Machine Learning (ML) are the quickest rising areas of healthcare industry. AI and ML are found in the research arena that deals with and improves technological systems to resolve complex tasks through reducing necessity of human intelligence. Pneumonia detection can be effectively treated through its early detection. Thus, the availability of proper screening methods is important for detecting the initial symptom of pneumonia.

Deep learning (DL) which is part of machine learning family depended on artificial neural networks. DL architectures, such as DNN (deep neural networks), RNN (recurrent neural networks), DBN (deep belief networks), and CNN, are generally applied to the areas likecomputer vision, audio recognition, speech recognition, social network filtering, natural language processing, machine translation, drug design, bioinformatics, medical image analysis, materials scrutiny, Histopathological diagnosis, and board game programs . These new technologies, in particular DL algorithms, can be applied to improve the diagnostic accuracy and efficiency of cancer detection.

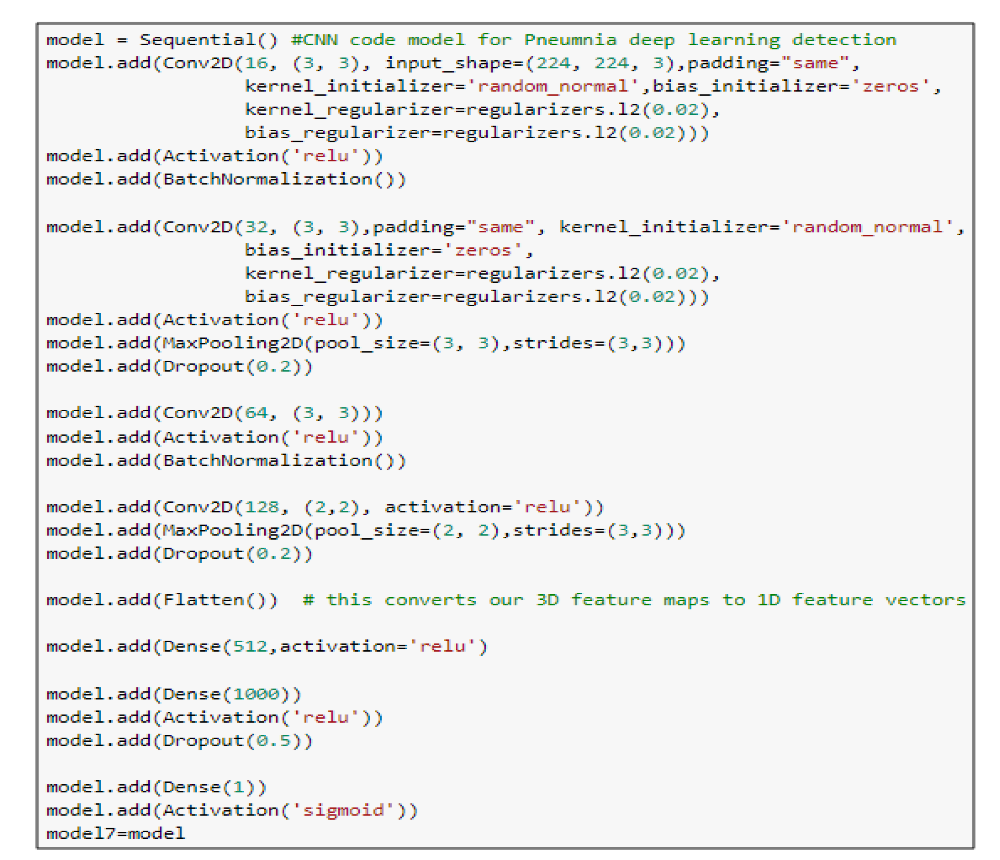
* 1. **Proposed System**
     1. **Proposed architecture**
* **CNN Model**

Deep neural networks with convolutional neural networks (CNNs) are employed to identify the pneumonia diagnosis of chest X-rays as a feature extraction and classification method. The proposed CNN model is demonstrated in figure 2. It consists of input, feature extraction, and classification layers.



**FIGURE 2: CNN Model**

The input layer has a 224×224×3 chest image. The feature extraction part consists of four CNN blocks. Each one of these blocks has mainly a convolution layer, a batch normalization layer, and a ReLU layer. It may have maximum pooling and a dropout layer, as shown in [**Figure 2**](https://www.mdpi.com/2075-4418/10/9/649/htm#fig_body_display_diagnostics-10-00649-f002). The output of the feature extraction part is then passed to the flattened layer to change the data shape to a one-dimensional data vector, which is the correctly used format for the classification dense layer. The dense layer is the regular, deeply connected neural network layer. It is the most common and frequently used layer, where every input is connected to every output. Here, we use three dense layers and four dropout layers. The final output is produced from a dense layer with sigmoid activation function that classifies the output image to Pneumonia (represented in the figure by blue arrow) or normal (represented by red arrow). The proposed CNN and the main function of the code is shown in [**Figure 3**](https://www.mdpi.com/2075-4418/10/9/649/htm#fig_body_display_diagnostics-10-00649-f003). The total number of model parameters is 38,320,049: the trainable parameters amount to 38,319,889, and the non-trainable parameters only amount to 160.



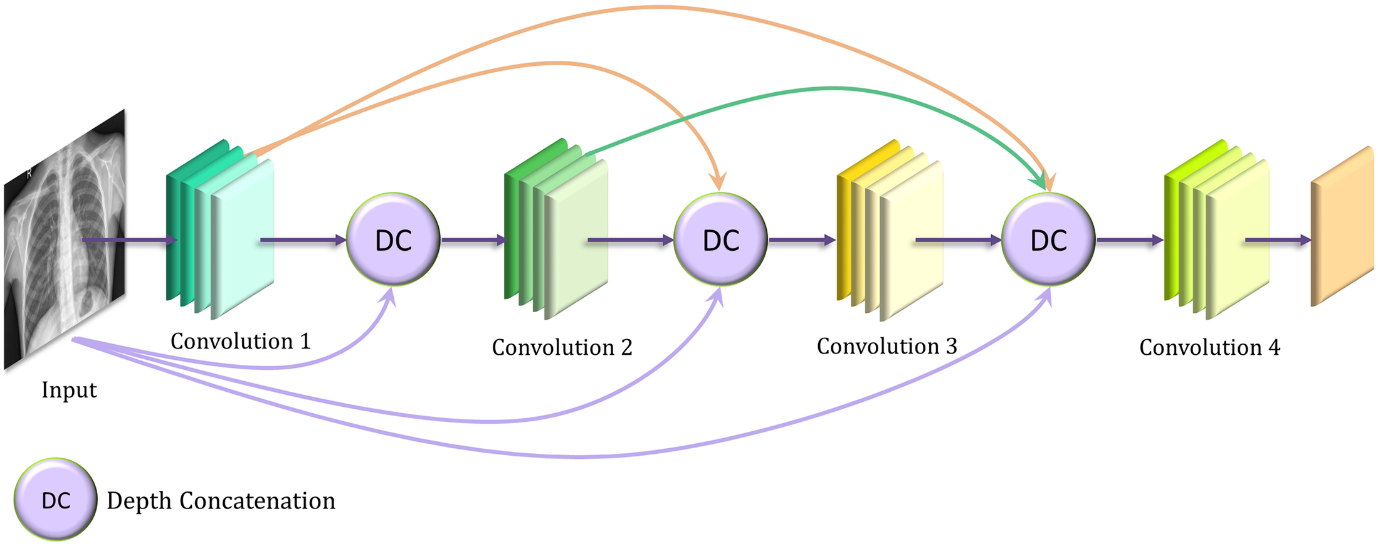
**FIGURE 3:** The main function code of the CNN model.

* + 1. **Proposed solution**

In this study, we designed an ensemble framework of classifier DenseNet-121 [[36](https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0256630#pone.0256630.ref036)], using a weighted average ensemble scheme wherein the weights allocated to the classifiers are generated using a novel scheme, as explained in detail in the following sections.

* **DenseNet-121**

The DenseNet architectures proposed by Huang et al. provide a rich feature representation while being computationally efficient. The primary reason is that, in each layer of the DenseNet model, the feature maps in the current layer are concatenated with those from all the preceding layers, as shown in figure 4. Because fewer channels are accommodated in the convolutional layers, the number of trainable parameters is diminished, and thus, the model is computationally efficient. Furthermore, the concatenation of the feature maps from the previous layers with the current layer enhances the feature representation.



**FIGURE 4: DenseNet**

1. **ANALYSIS**
   1. **INTRODUCTION:**

While selecting the algorithm that gives an accurate prediction we gone through lot of algorithms which gives the results abruptly accurate and from them we selected only one algorithm for the prediction problem that is convolution neural networks, it assumes that the presence of a particular feature detector in a image data generator class detects the image and transforms the image in the batch by a series of random translations and these translations are based on arguments and replaces original batch with new data. And also performs data augmentation. That's how the prediction work great with the neural networks. The peculiarity of this problem is collecting the data of Pneumonia details real time and working with the prediction at the same time, so we developed a user interface for the people who'll have the symptoms of for the type Pneumonia prediction Accuracy is defined as the ratio of the number of samples correctly classified by the classifier to the total number of samples for a given test data set. The formula is as follows

Accuracy=TP+ TN/TP+ TN+FT+FN

At first, we got like lot of worst accuracies because we tried lot of trained images by giving a greater number of epochs for the best accurate algorithm, finally after all of that we tried the best suitable algorithm which gives the prediction accurately is convolution neural networks. And developed it to use as a real time prediction problem for the pneumonia detection.

* 1. **SOFTWARE REQUIREMENT SPECIFICATION:**

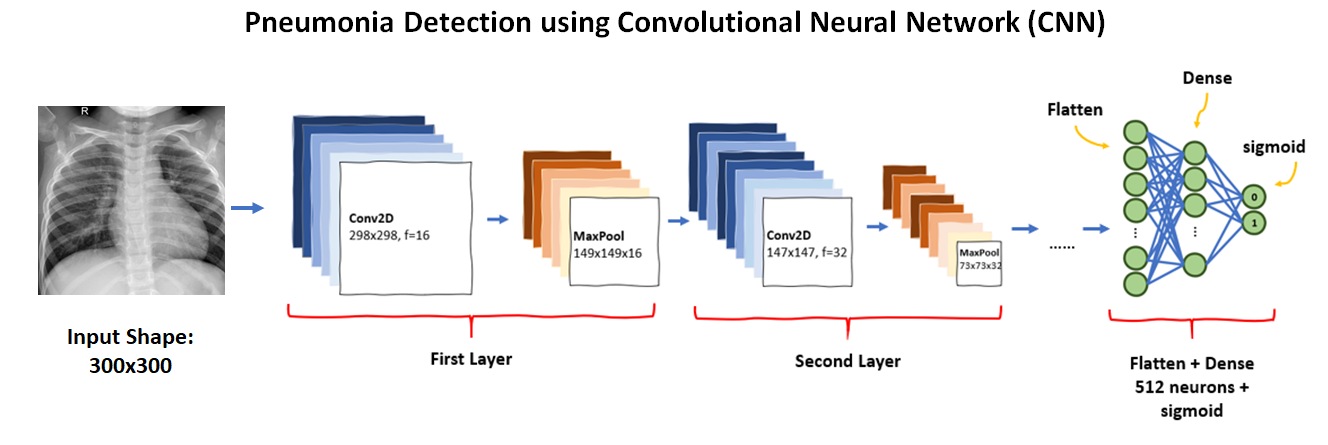
The software requirements specification document lists sufficient and necessary requirements for the project development. To derive the requirements, the developer needs to have clear and thorough understanding of the products under development. This is achieved through detailed and continuous communication with the project team and customer throughout the software development process.

* + 1. **Software Requirement:**
* Jupyter Notebook Environment
* Visual studio code
* Python 3.9
* IBM Watson Studio
* Spyder lde
* Deep Learning Algorithms
* Python (pandas, NumPy, Keras, TensorFlow)
* HTML
* Flask

We developed this garbage classification by using the Python language which is an interpreted and high-level programming language and using the Deep Leaming algorithms. For coding we used the Jupiter Notebook environment of the Anaconda distributions and the Spyder, it is an integrated scientific programming in the python language. For creating a user interface for the prediction, we used flask. It is a micro web framework written in Python and uses WSGI for web development. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions, and a scripting language to create a webpage is HTML by creating the templates to use in the functions of the Flask and HTML.

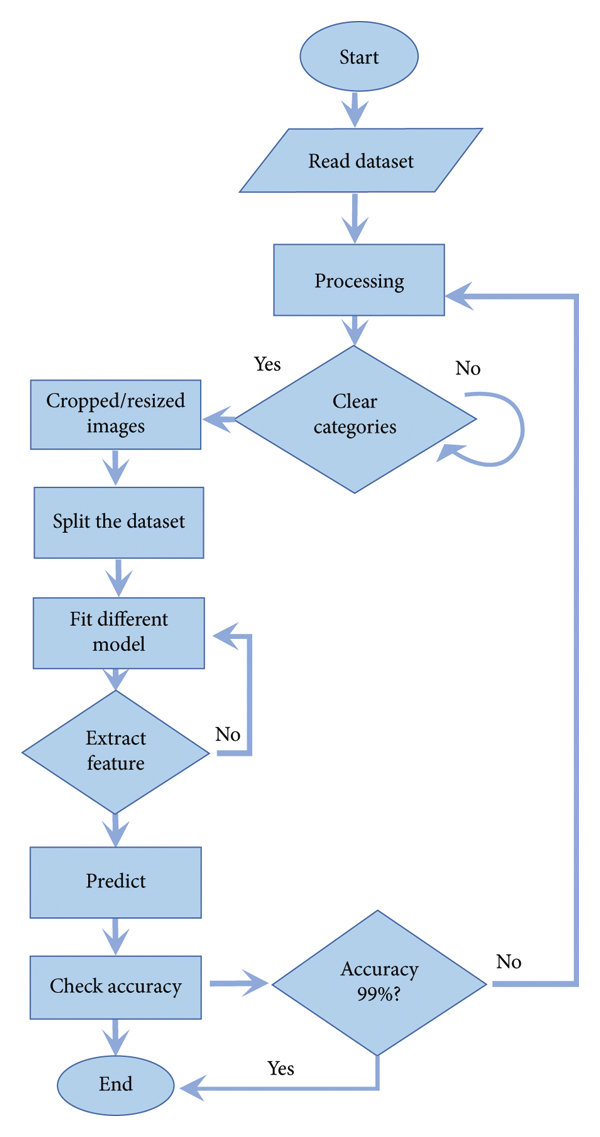
* + 1. **Hardware Requirement:**
       - System: 64-bit windows 10
       - RAM:4GB
       - Processor:2.3GHz

* 1. **CONTENT DIAGRAM OF PROJECT:**

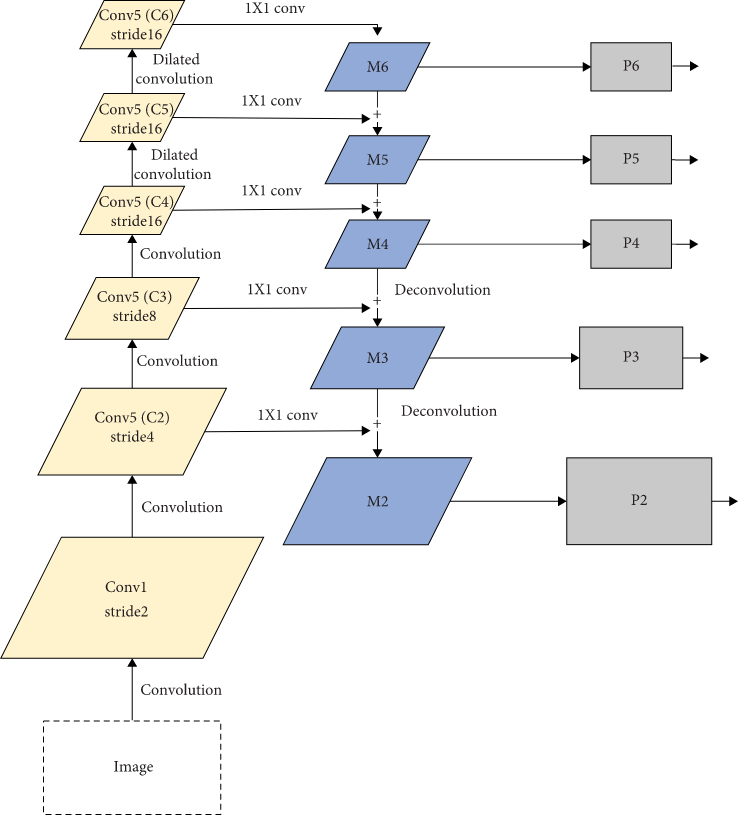


**FIGURE 5: Fully Connected layer**

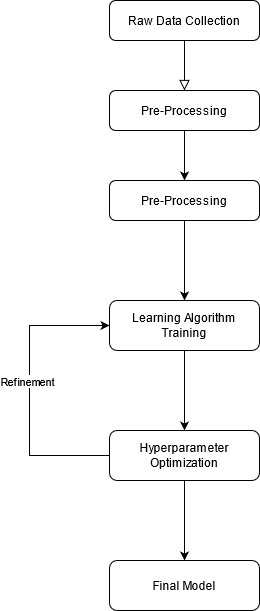
* 1. **ALGORITHMS AND FLOWCHARTS:**



**FIGURE 6: Flow chart**

****

**FIGURE 7: Algorithm**



**Figure 8: Flow chart**

In this paper, Deep Learning Algorithm called Convolution Neural Networks is used for better prediction. Built an algorithm to automatically identify the type of the garbage through an image. The convolution neural network is used for an efficient and accurate processing of the image processing.

* 1. **CONCLUSION:**

Making use of the prior computer technology we have successfully found an algorithm to be trained in-order to identify the image and developed a process to classify and sort the images in to pneumonia and normal chest x-rays and says whether the person is affected.

1. **DESIGN**
   1. **INTRODUCTION:**

* **CNN Architecture:**

Let's go step by step and analyze each layer in the Convolutional Neural Network

* **Input**

A Matrix of pixel values in the shape of [WIDTH, HEIGHT, CHANNELS]. Let's assume that our input is [32x32x3]

* **Convolution layer:**

The purpose of this layer is to receive a feature map. Usually, we start with low number of filters for low-level feature detection. The deeper we go into the CNN, the more filters we use to detect high-level features. Feature detection is based on 'scanning' the input with the filter of a given size and applying matrix computations in order to derive a feature map.

In a CNN, the input is a Tensor with a shape: (number of inputs) x (input height) x (input width) x (input channels). After passing through a convolutional layer, the image becomes abstracted to a feature map, also called an activation map, with shape: (number of inputs) x (feature map height) x (feature map width) x (feature map channels).

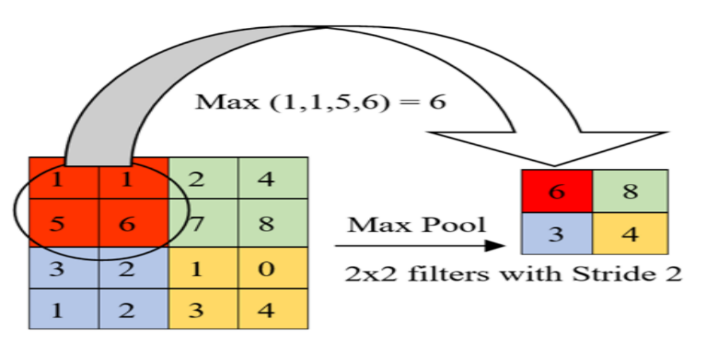
Convolutional layers convolve the input and pass its result to the next layer. This is similar to the response of a neuron in the visual cortex to a specific stimulus. Each convolutional neuron processes data only for its receptive field. Although fully connected feed forward networks can be used to learn features and classify data, this architecture is generally impractical for larger inputs such as high resolution images. It would require a very high number of neurons, even in a shallow architecture, due to the large input size of images, where each pixel is a relevant input feature. For instance, a fully connected layer for a (small) image of size 100 x 100 has 10,000 weights for *each* neuron in the second layer. Instead, convolution reduces the number of free parameters, allowing the network to be deeper. For example, regardless of image size, using a 5 x 5 tiling region, each with the same shared weights, requires only 25 learnable parameters. Using regularized weights over fewer parameters avoids the vanishing gradients and exploding gradients problems seen during backpropagation in traditional neural networks. Furthermore, convolutional neural networks are ideal for data with a grid-like topology (such as images) as spatial relations between separate features are taken into account during convolution and/or pooling.

* **Pooling Layer:**

The goal of this layer is to provide spatial variance, which simply means that thesystemwillbecapableofrecognizinganobjectevenwhenitsappearancevariesinsome way. Pooling layer will perform a down sampling operation along the spatial dimensions (width, height), resulting in output such as [16xl 6x12] for pooling size= (2, 2)

Convolutional networks may include local and/or global pooling layers along with traditional convolutional layers. Pooling layers reduce the dimensions of data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer. Local pooling combines small clusters, tiling sizes such as 2 x 2 are commonly used. Global pooling acts on all the neurons of the feature map. There are two common types of pooling in popular use: max and average. *Max pooling* uses the maximum value of each local cluster of neurons in the feature map, while *average pooling* takes the average value.

This layer cut down the values further to half of its original value by choosing only max values from the kernel matrix. The sample illustration is the 4x4 matrix pixel values of an input image degraded into 2x2 filters.



**FIGURE 9: Max Pooling**

* **DenseNet:**

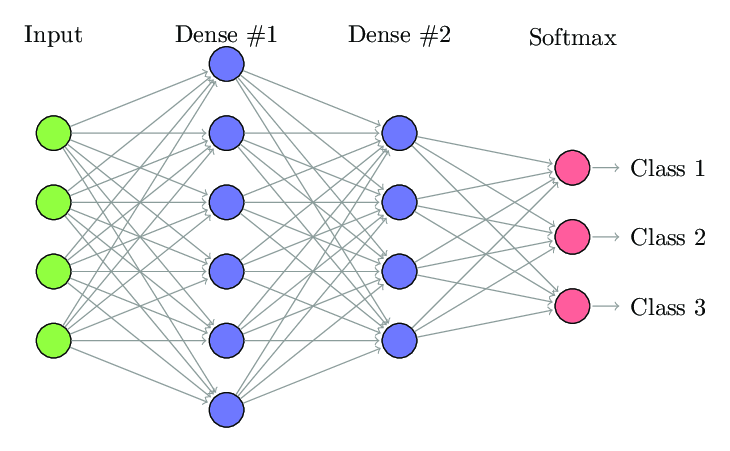
Studies have shown that the connections between layers close to the input and those close to the output help in increasing the performance of convolutional networks. This idea was implemented in residual networks [[63](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7345724/#B63-diagnostics-10-00417)] and also in dense convolutional networks (DenseNet). DenseNet requires lesser parameters compared to conventional convolutional networks as it does not require relearning excessive feature maps because of its dense connectivity pattern. The network is divided into dense blocks, with the dimensions of the feature maps remaining constant within a block and the number of filters changing between them. DenseNet has several advantages such as the number of parameters is significantly reduced, the features are reused, and the vanishing gradient is mitigated. Figure depicts the pre-trained DenseNet architecture used in the paper.

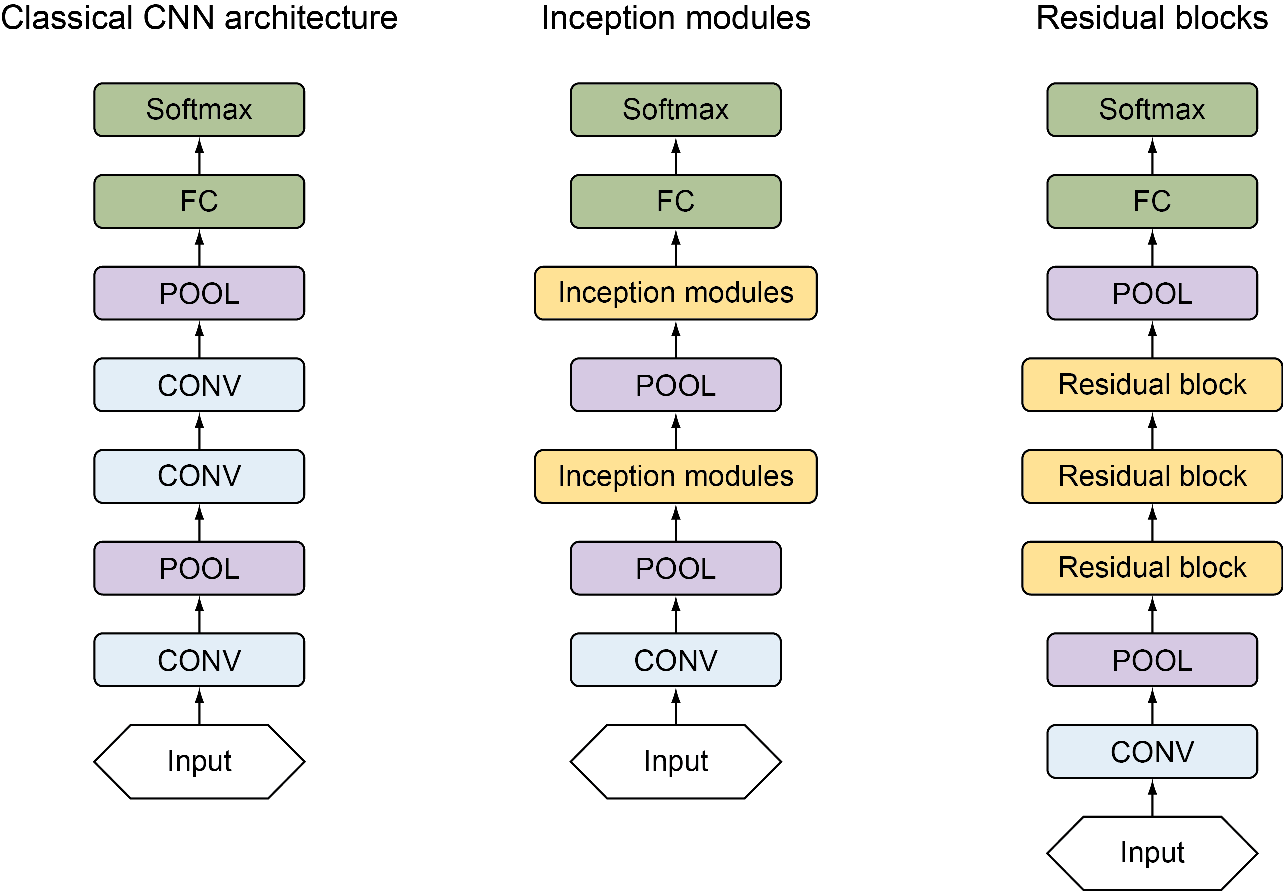


**FIGURE 10: Dense Net**

* **Fully connected layer:**

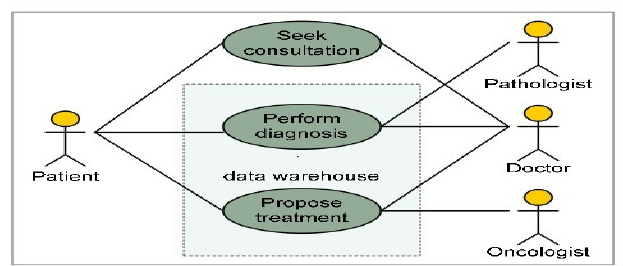
In a fully connected layer, we flatten and connect the output of the last convolution layer every node of the current layer with the other nodes of the next layer. Neurons in a fully connected layer have full connections to all activations in the previous layer, as seen in regular Neural Networks and work in a similar way.

**FIGURE 11: Fully Connected layer**

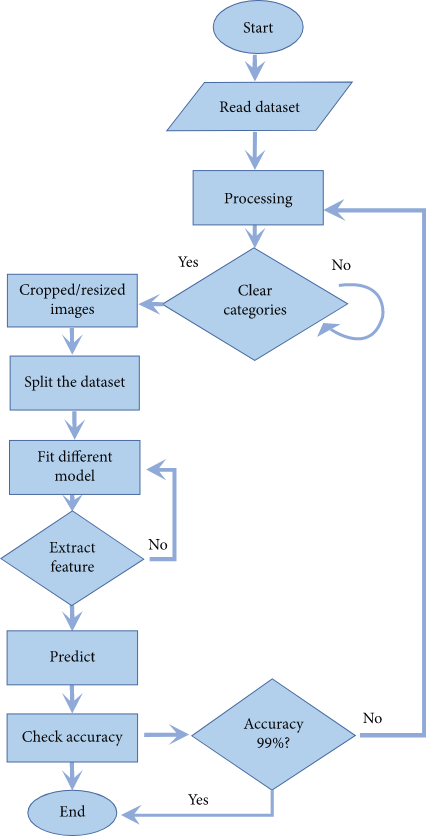


**FIGURE 12: Architecture of CNN**

* 1. **UML Diagram**
     1. **USE CASE DIAGRAM:**

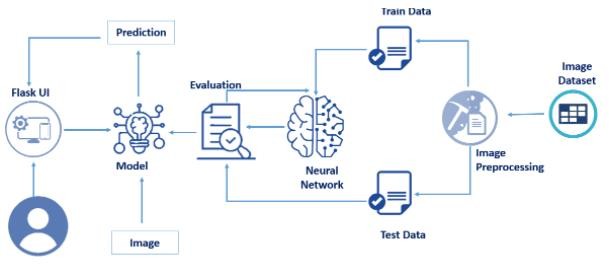


**FIGURE 13: Use case diagram**

****

**FIGURE 14: Flow chart**

* 1. **PROJECT ARCHITECTURE**



**FIGURE 15: Architecture**

* 1. **Model Design**
     1. **Project Flow:-**
* **Image Preprocessing**
  + Import image data generator library and configure it
  + Apply image data generator functionality to train and test datasets

### **Model Building**

### **Importing the required libraries for model building**

* + Initialize the model
  + Add convolution layer
  + Add max pooling layer
  + Add flatten layer
  + Add hidden layers
  + Compile the model
  + Fit and save the model

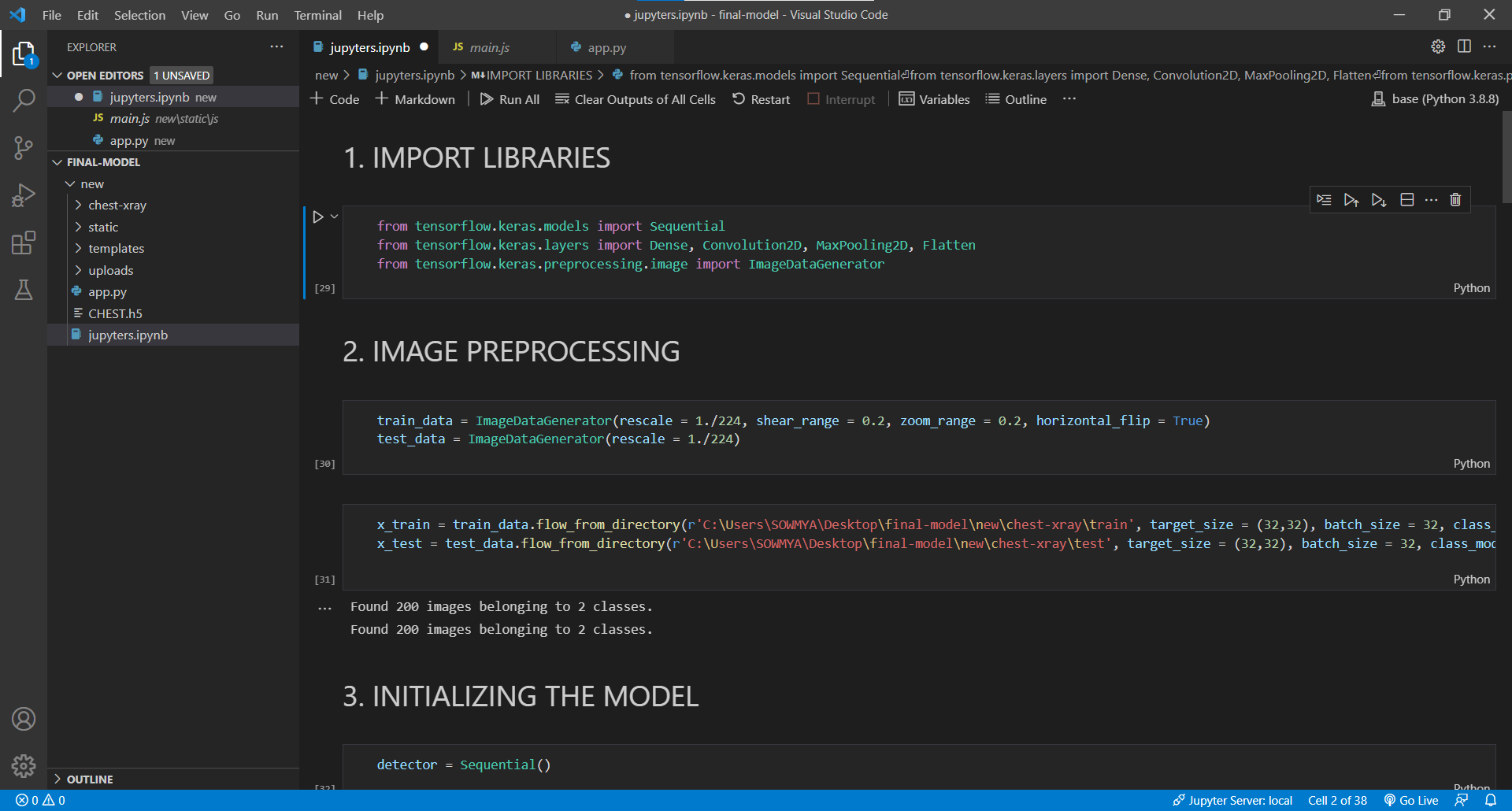
### **Test the Model**

* + Import the saved model
  + Load the test image, preprocess it and then predict and check for results

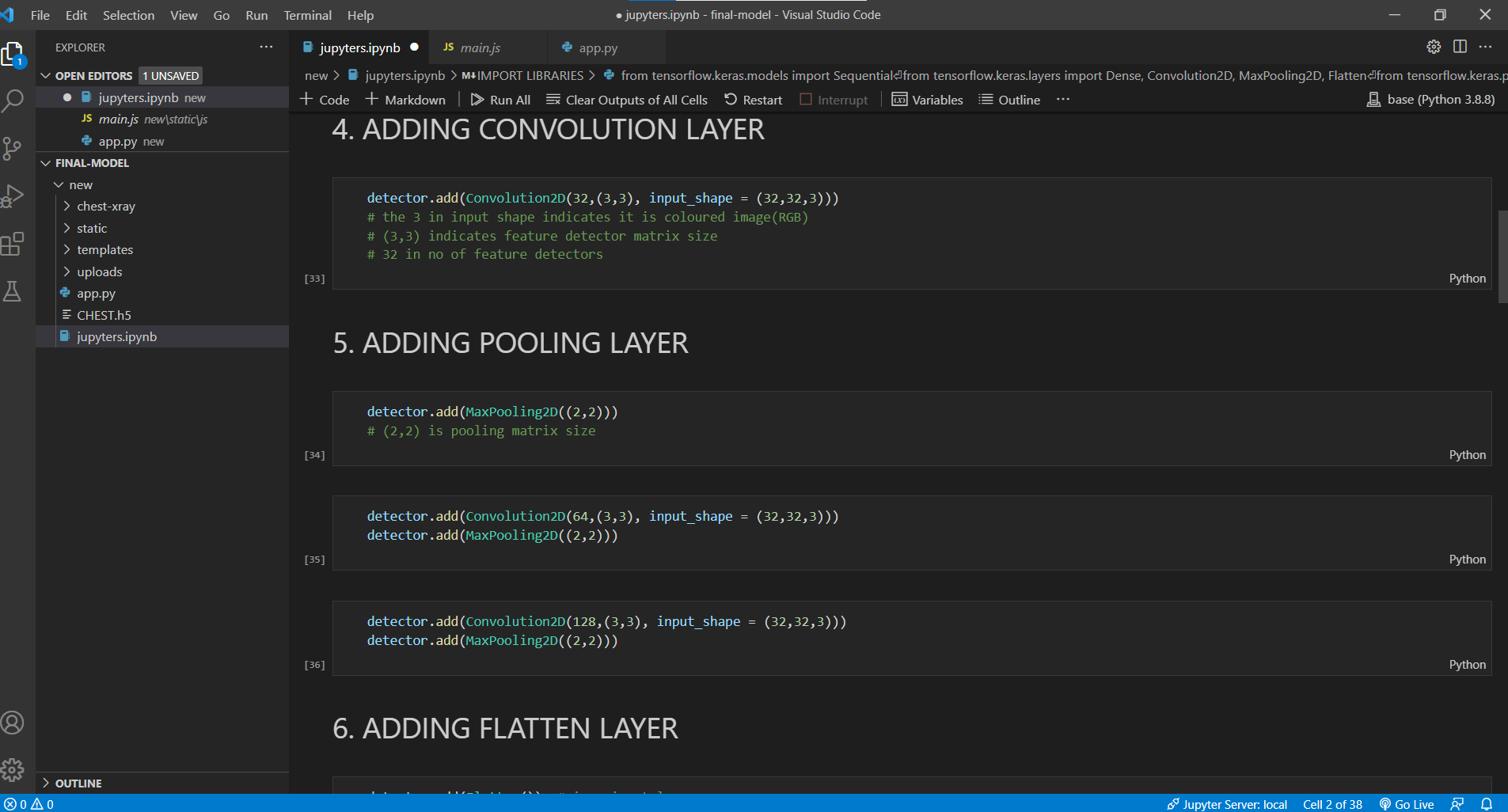
### **Application Building**

* + Build a FLASK application
  + Build the HTML page and execute it
  + Run the app

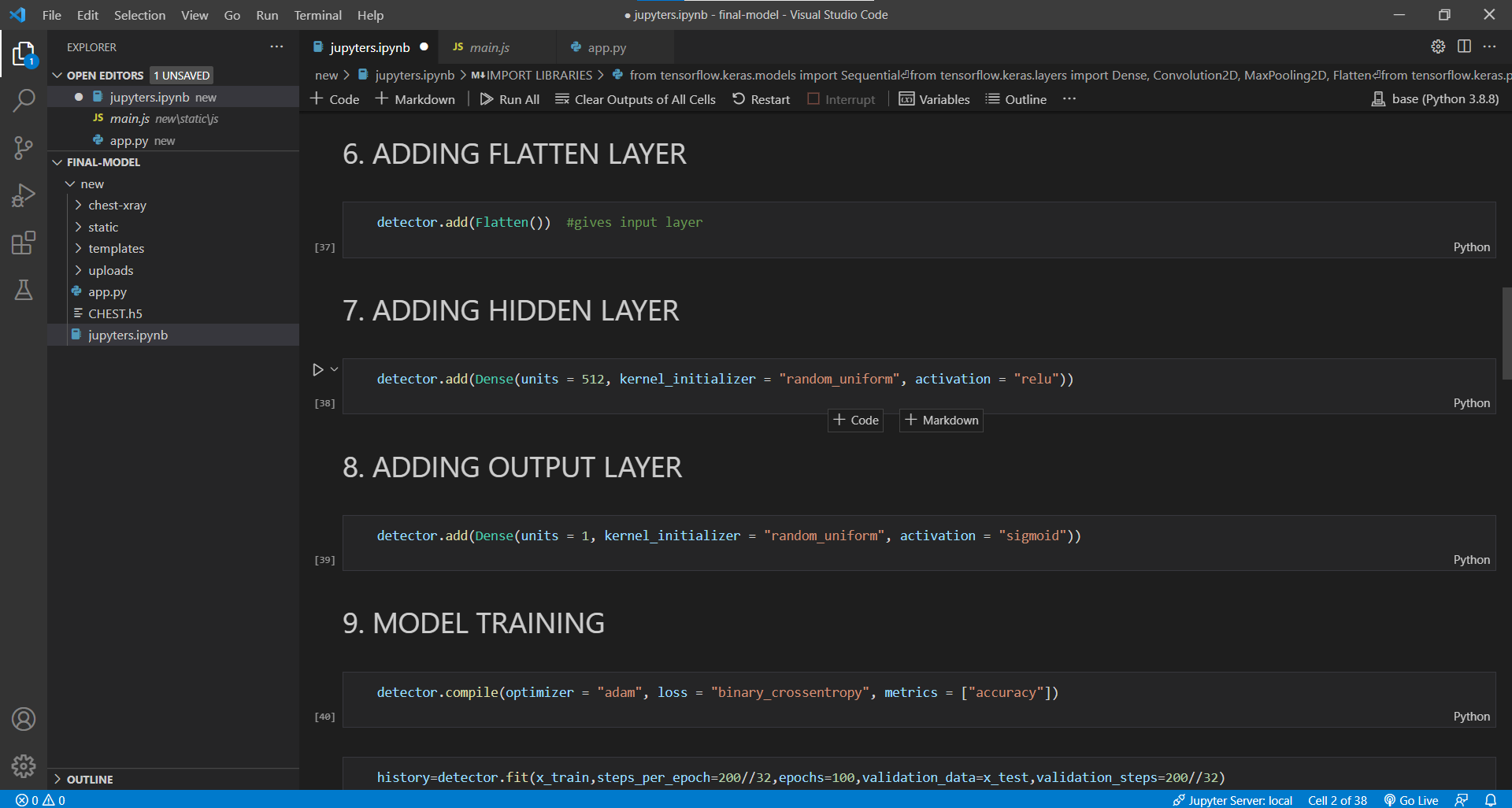
1. **CODE SNIPPETS**
   1. **Model code:**



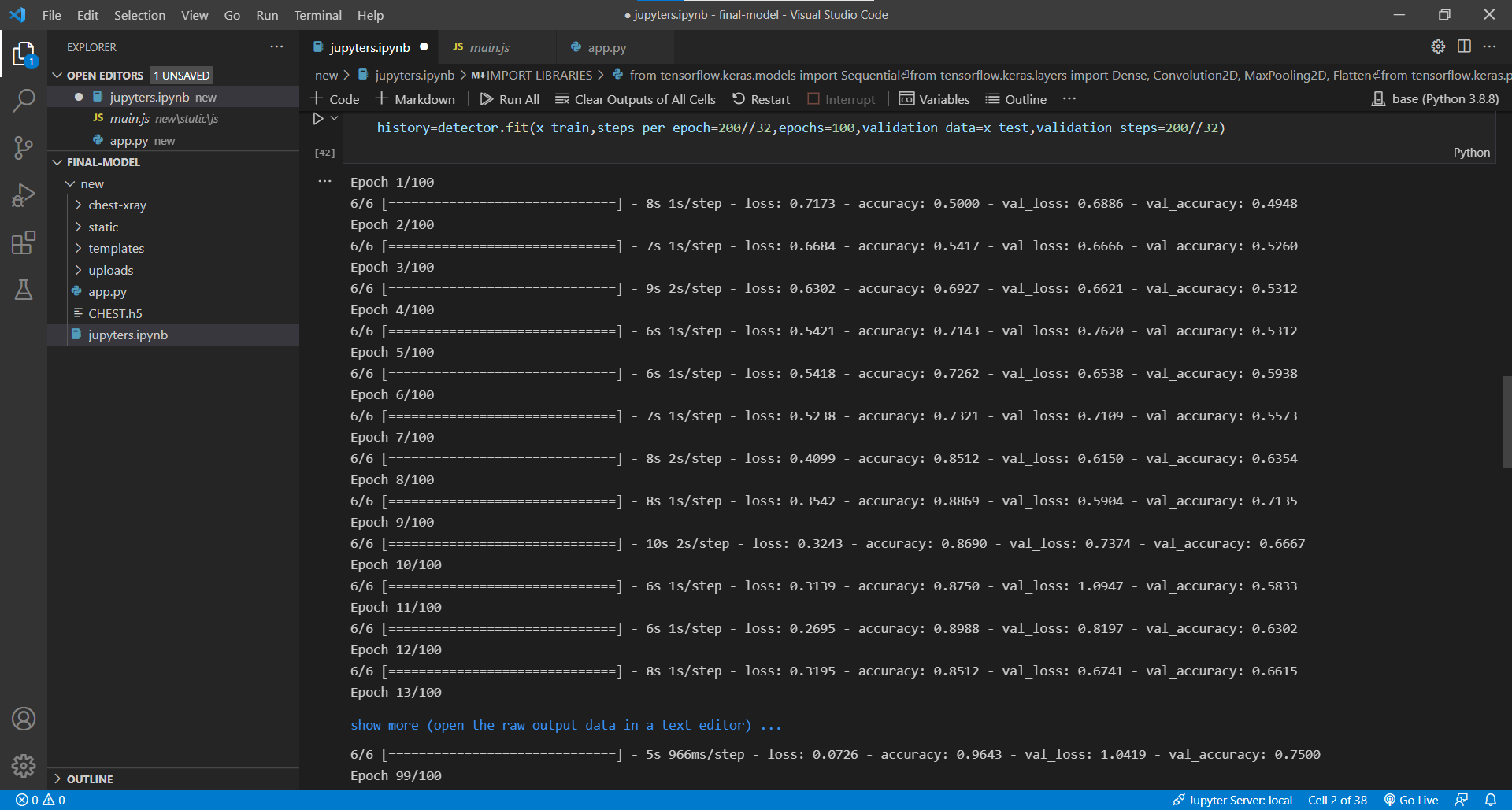
**FIGURE 16: .ipynb describing libraries importing and images preprocessing**

****

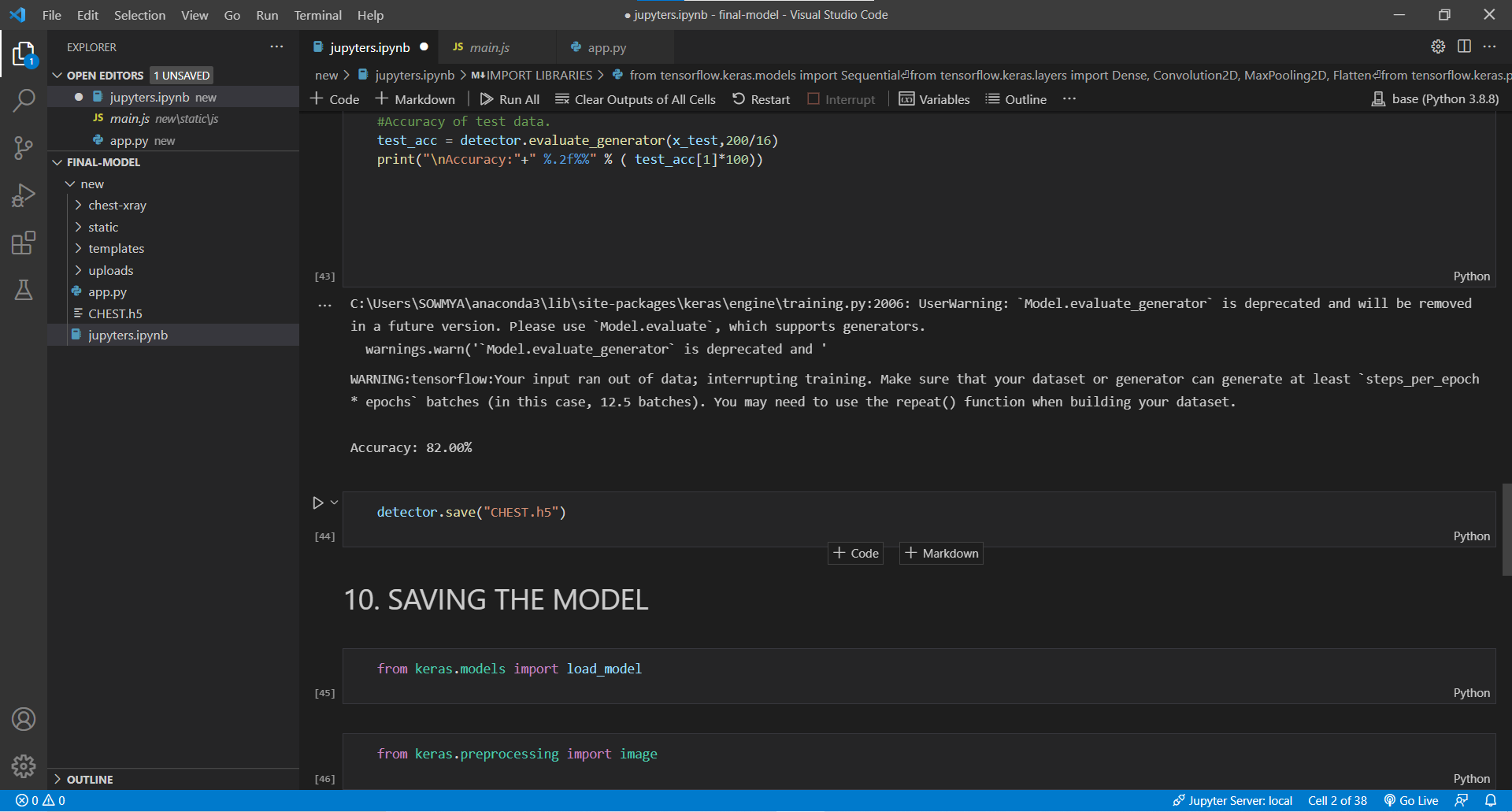
**FIGURE 17: .ipynb initialize the model and adding convolution layer**



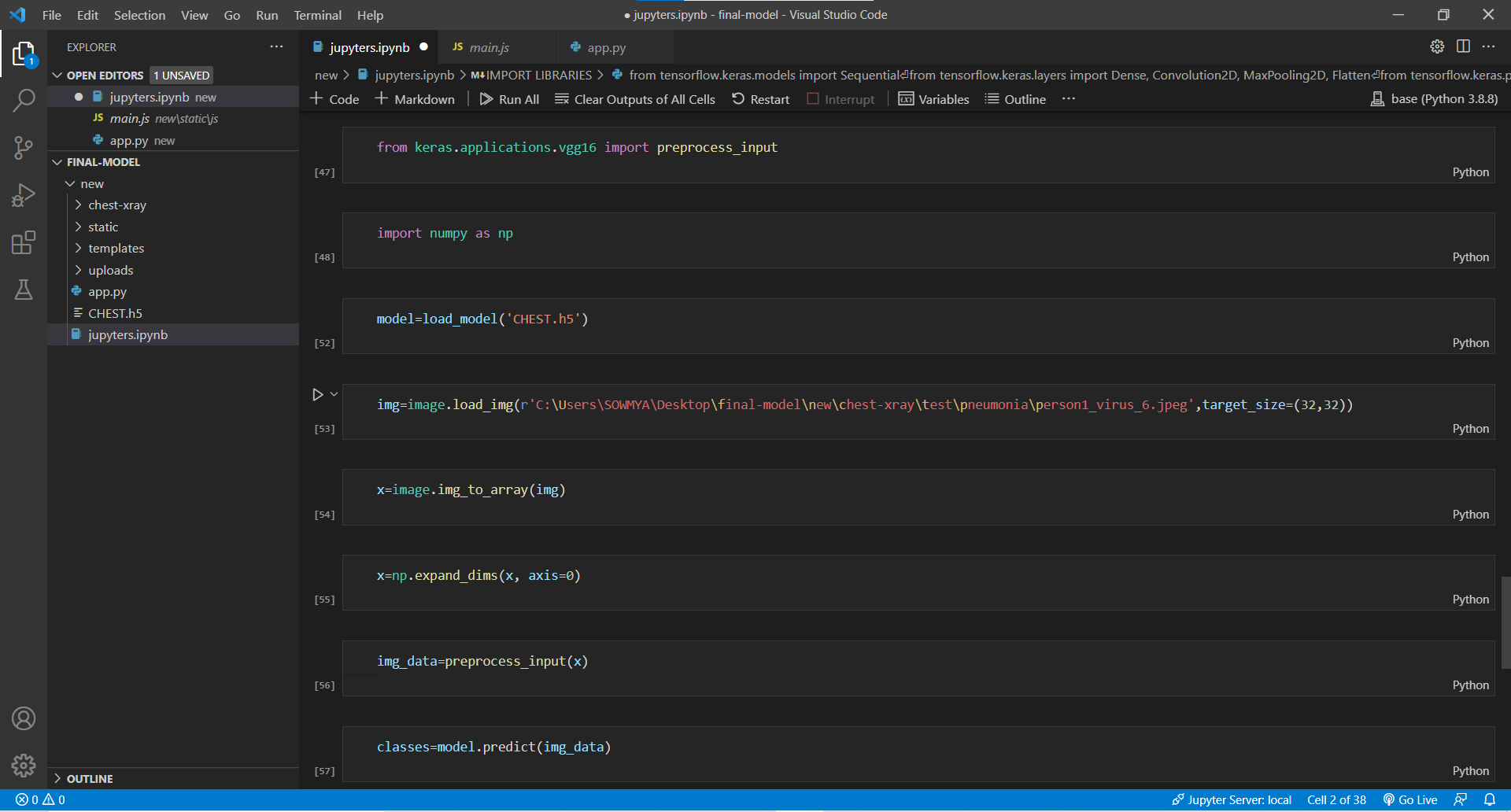
**FIGURE 18: .ipynb for adding flatten layer and hidden layer**

****

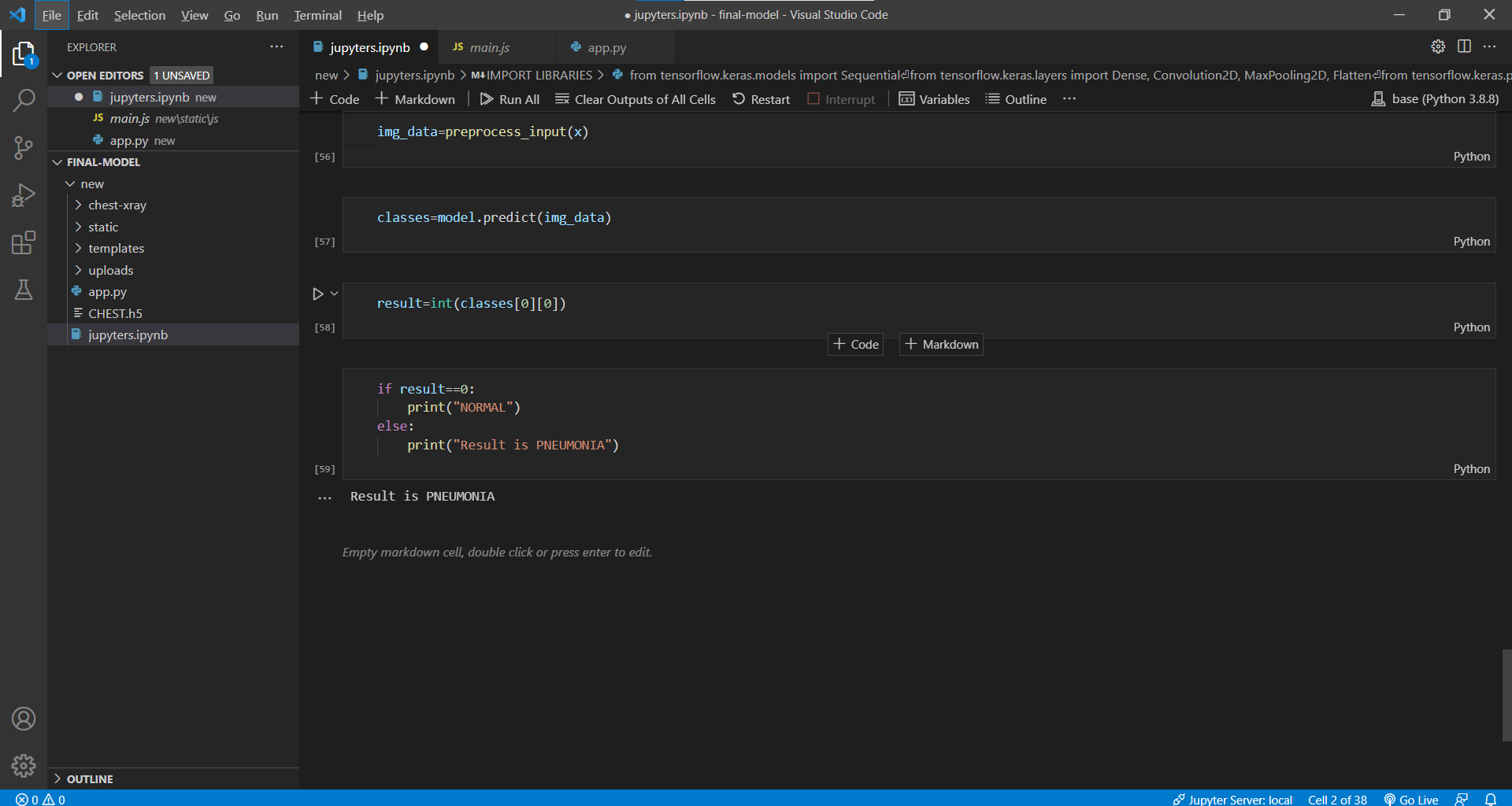
**FIGURE 19: .ipynb for adding output layer and Model training**

****

**FIGURE 20: .ipynb for saving the Model**

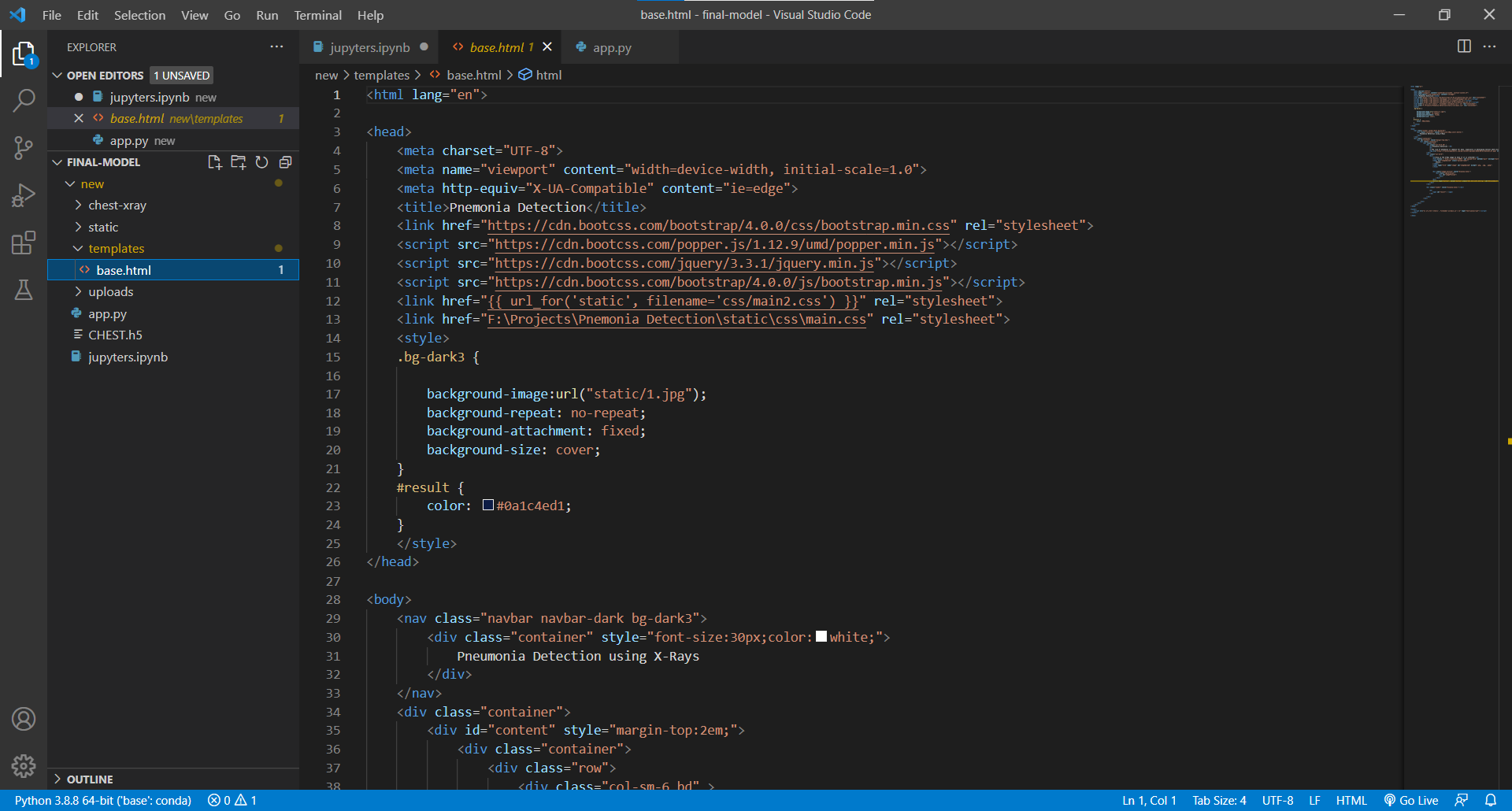
****

**FIGURE 21: .ipynb for Prediction**

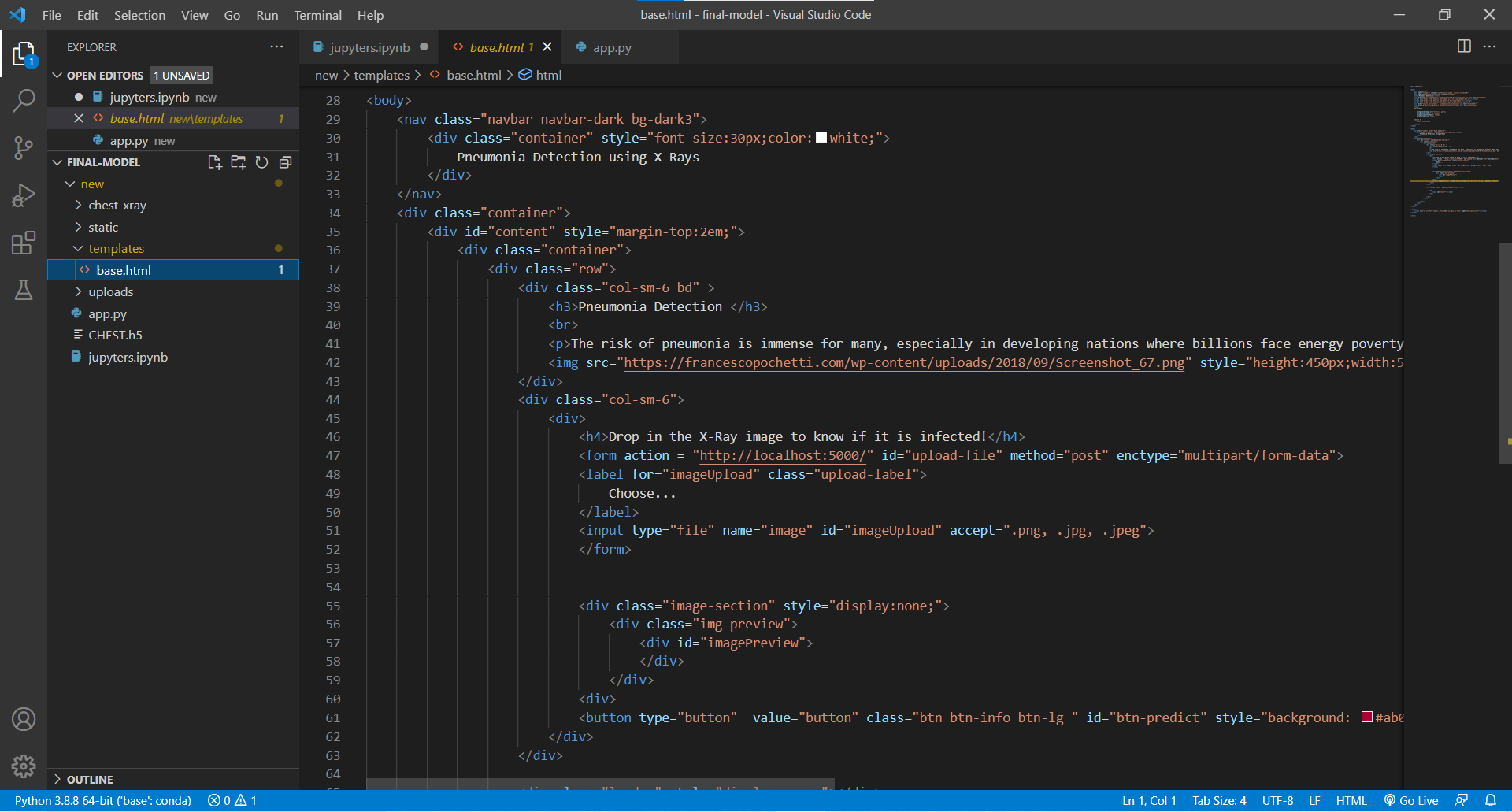
****

**FIGURE 22: .ipynb for Prediction**

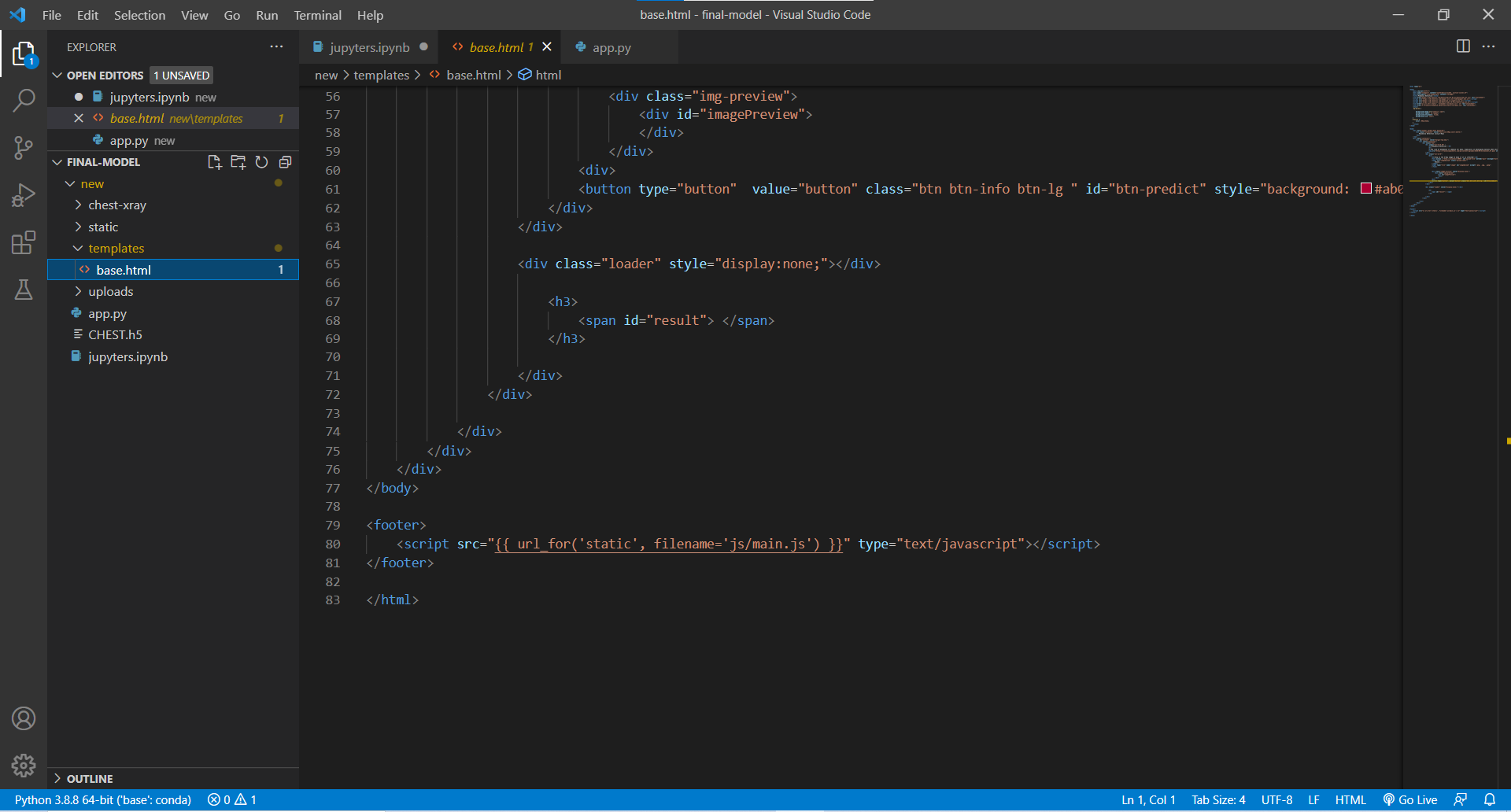
* 1. **Base.html:**

****

**FIGURE 23: base.html for describing the webpage styling**

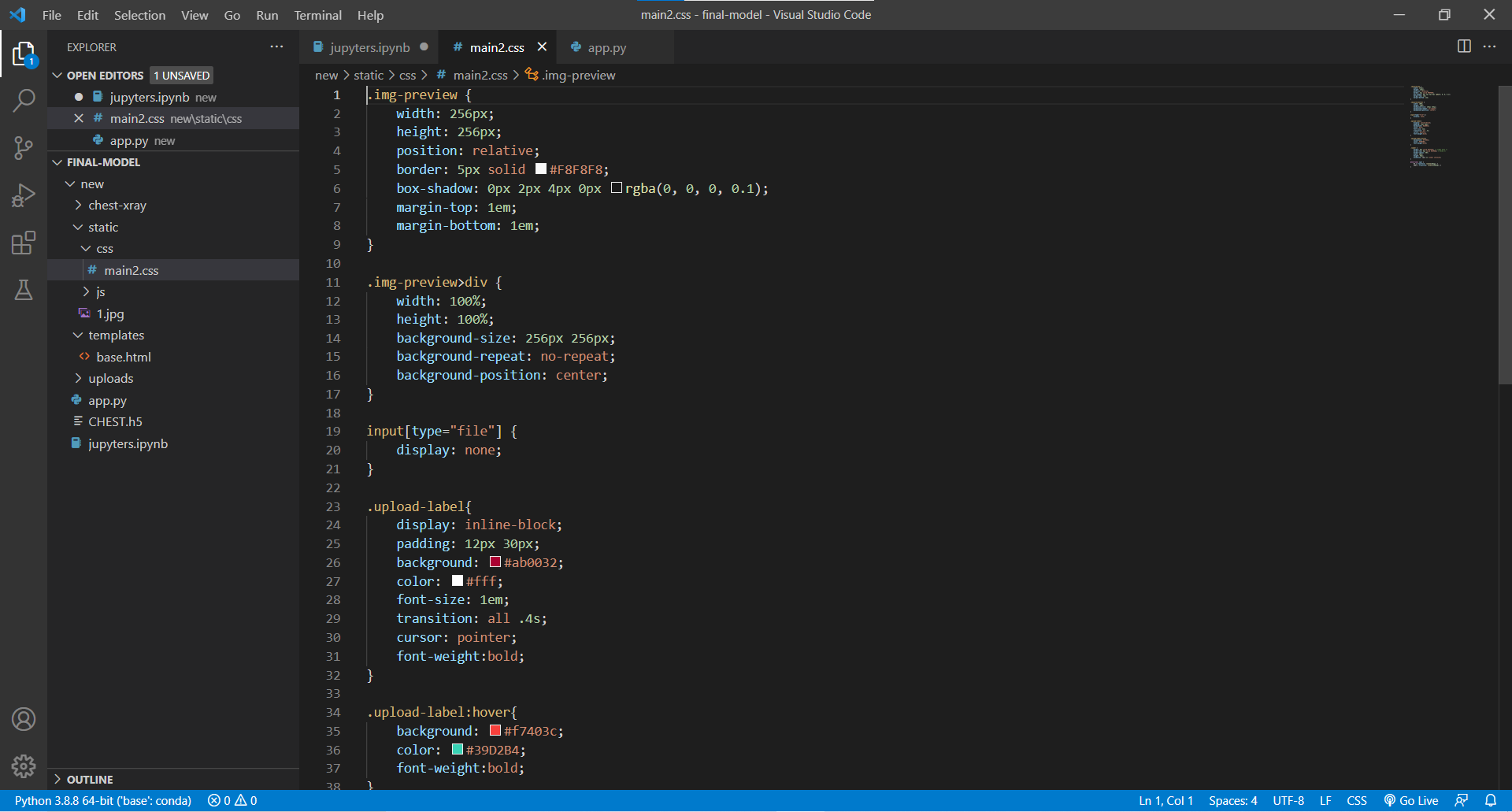
****

**FIGURE 24: base.html describing the body of webpage**

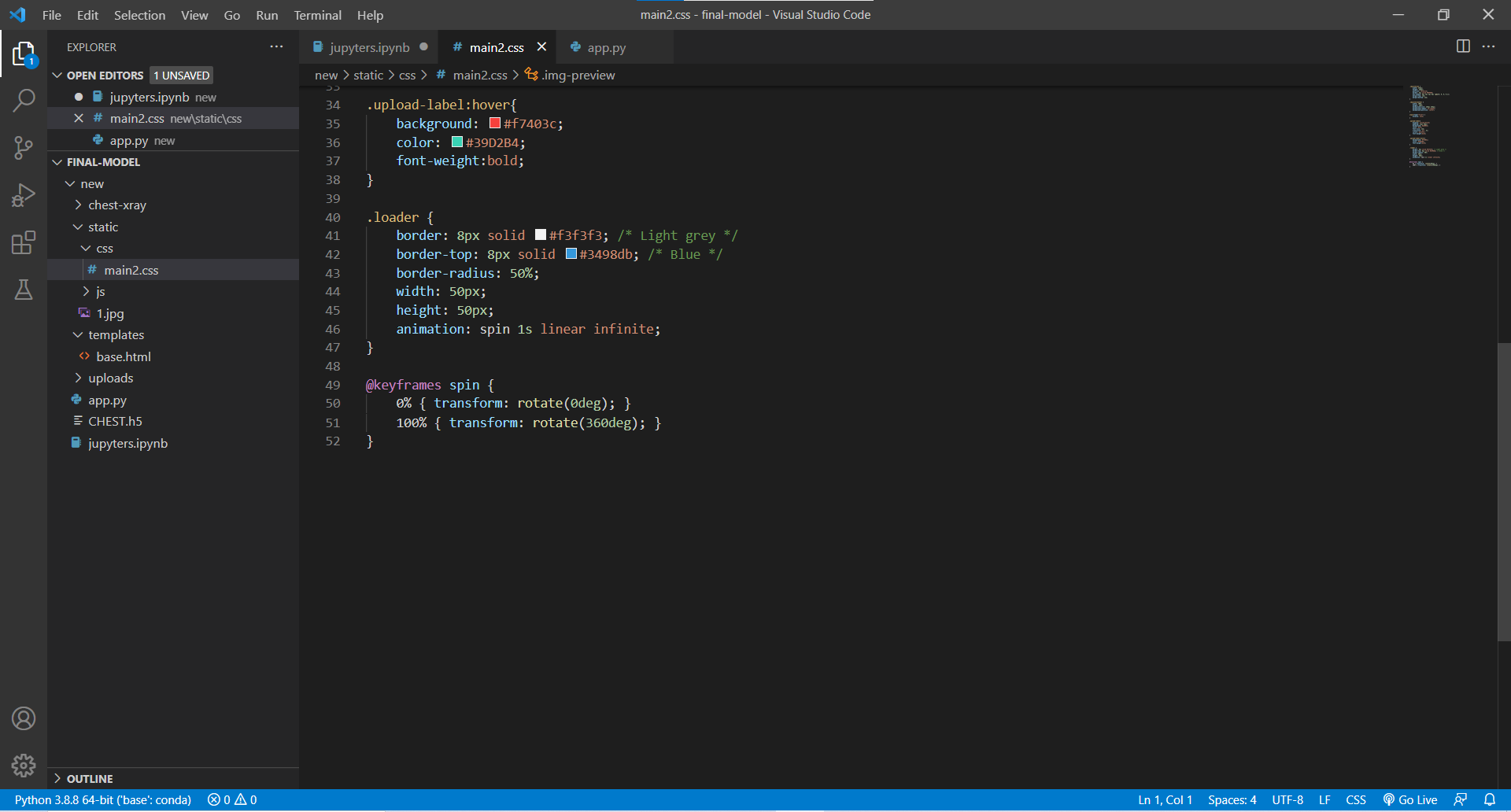
****

**FIGURE 25: base.html describing the body of webpage**

* 1. **CSS**
  + **Main2.css**

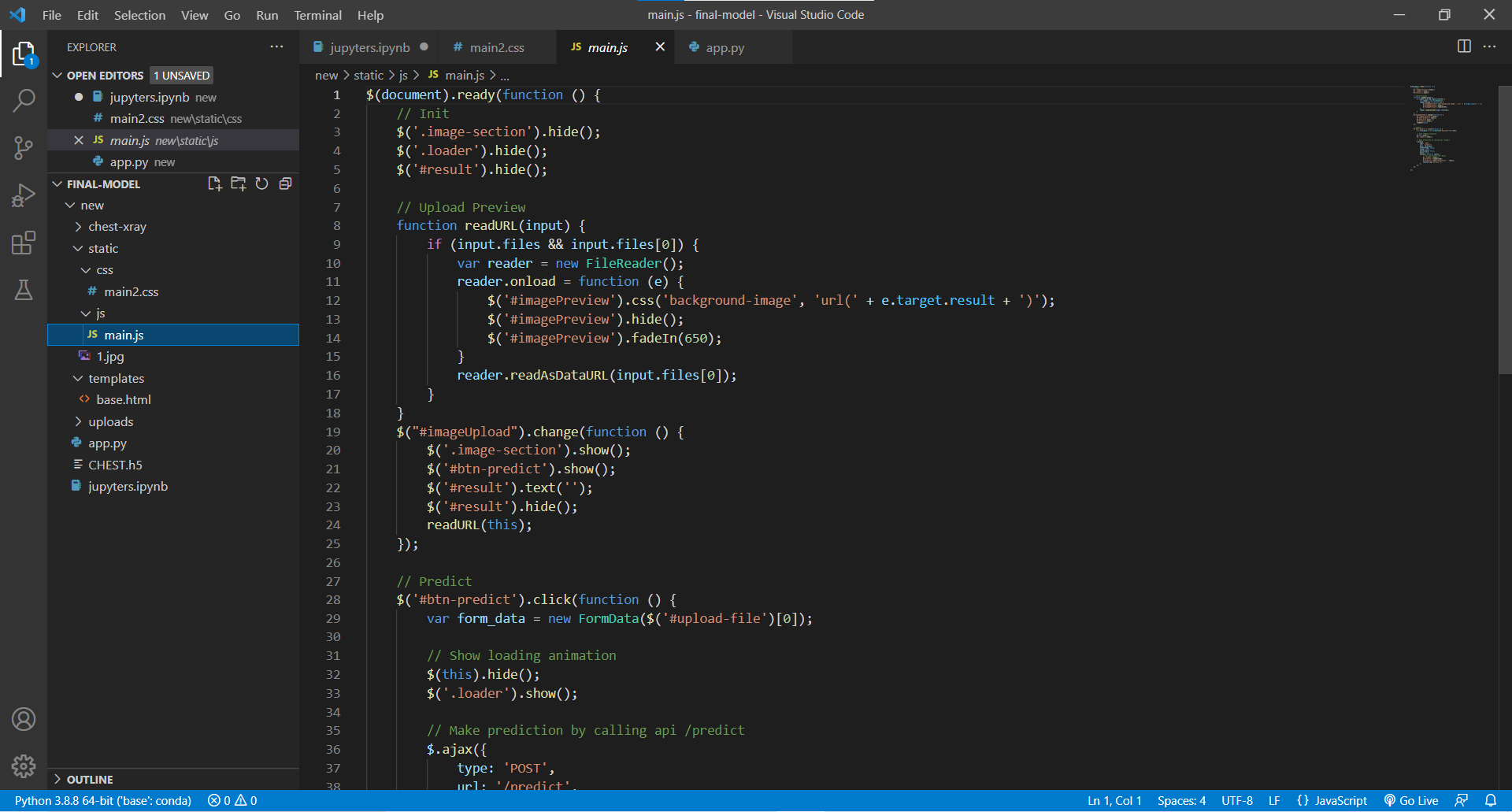


**Figure 26: CSS code for styling the webpage**

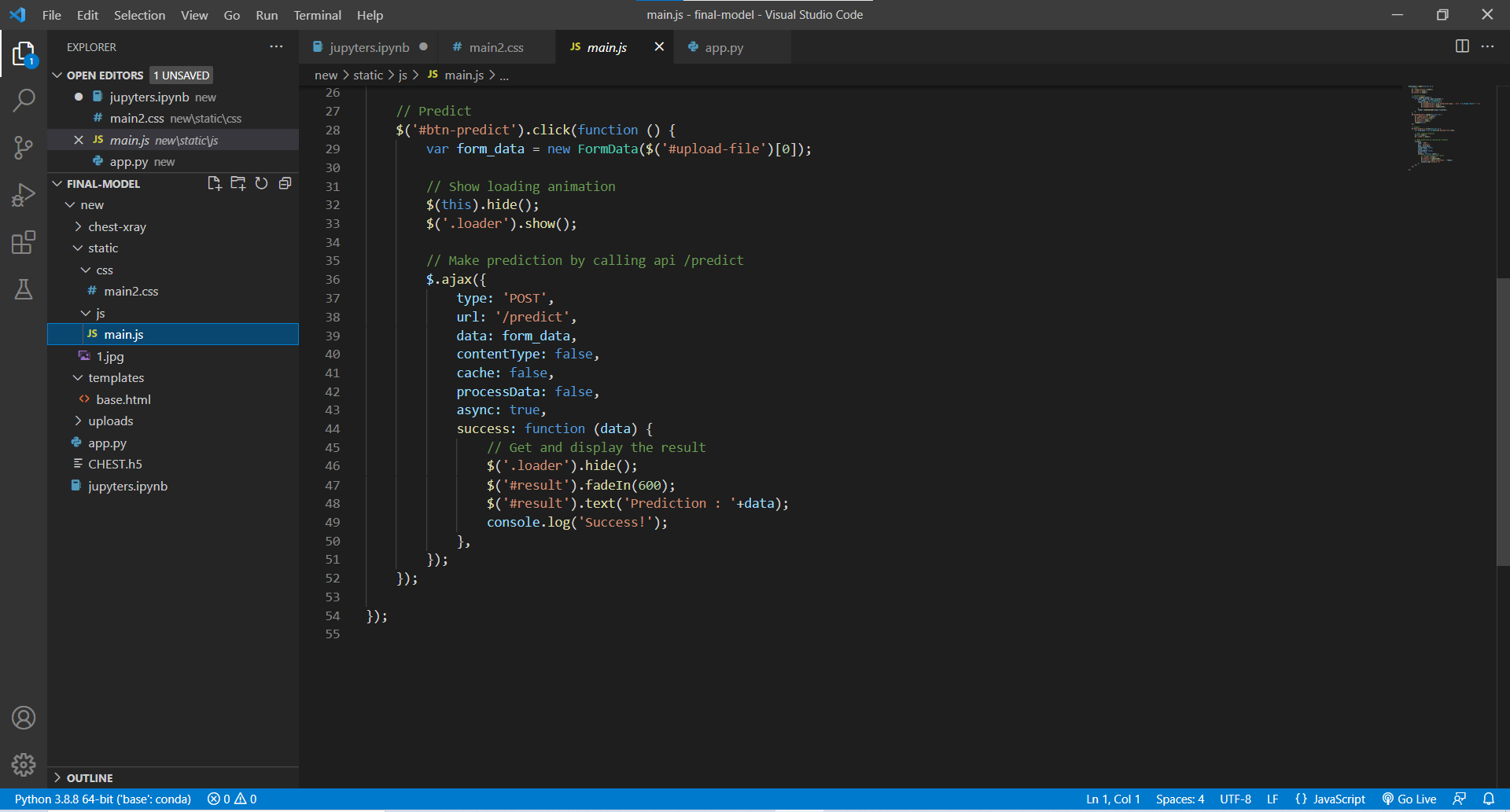
****

**Figure 27: CSS code for styling the webpage**

* 1. **JS:**

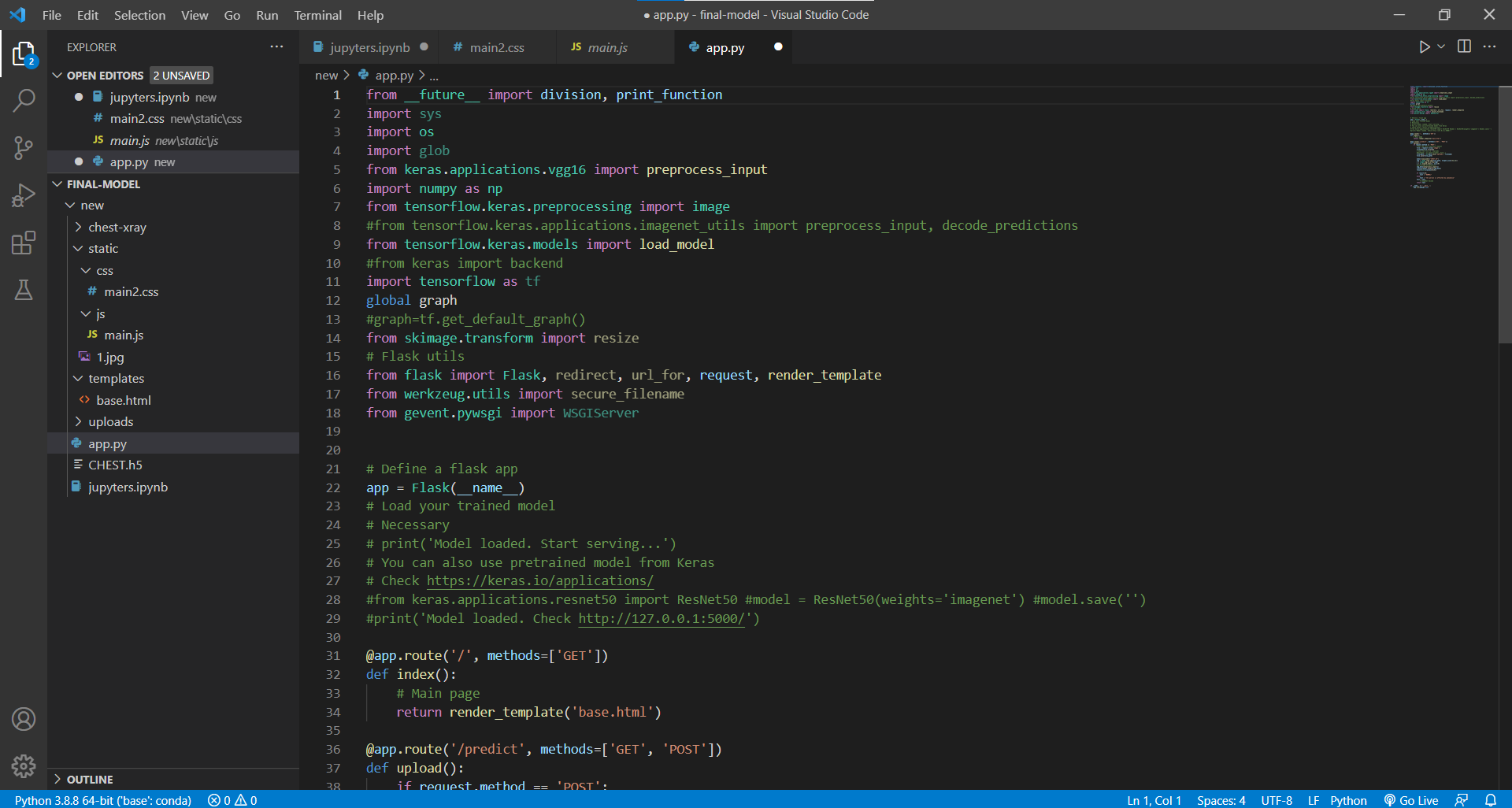
****

**Figure 28: JS code**

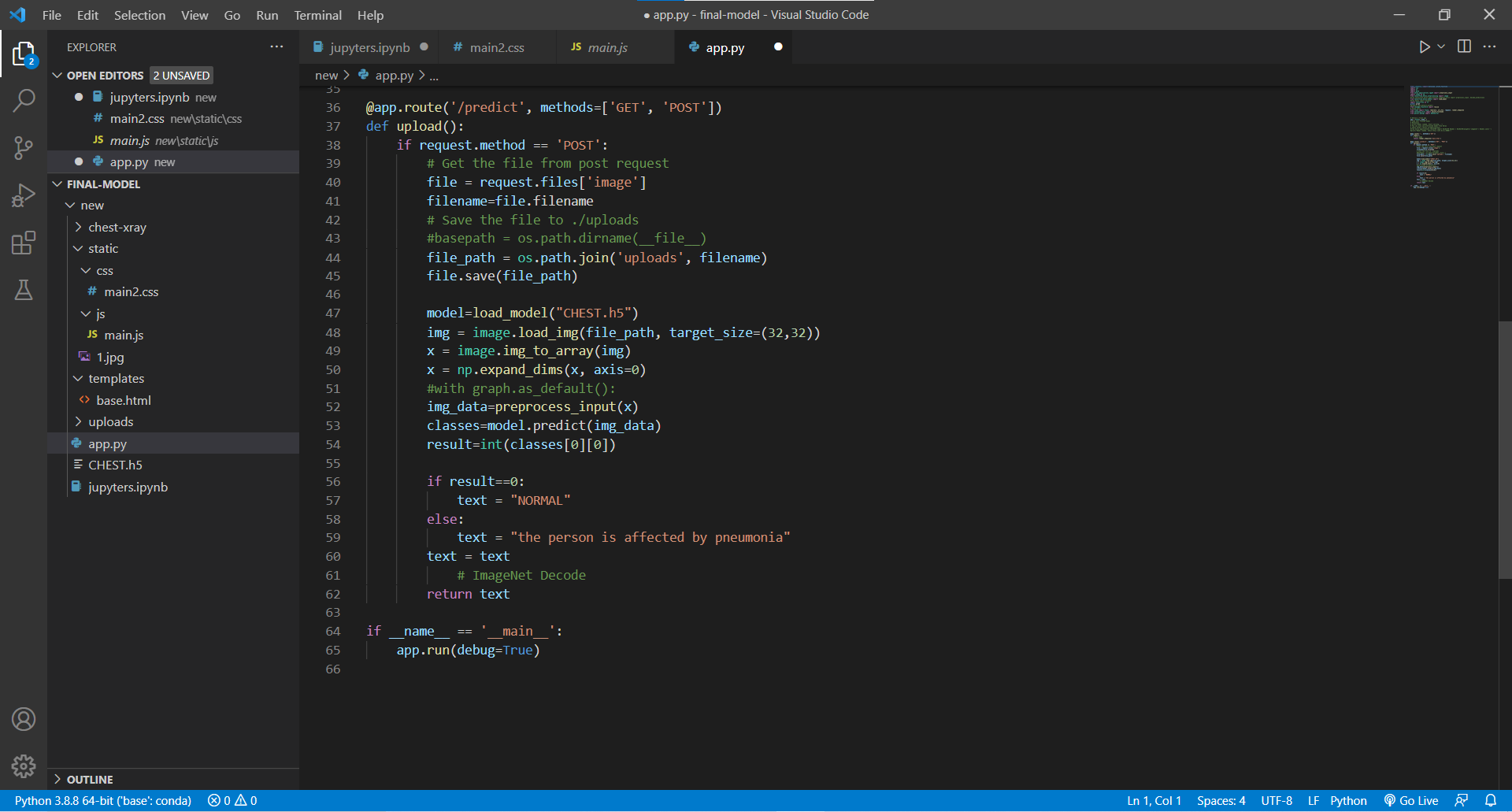
****

**Figure 29: JS code**

* 1. **APP.PY:**

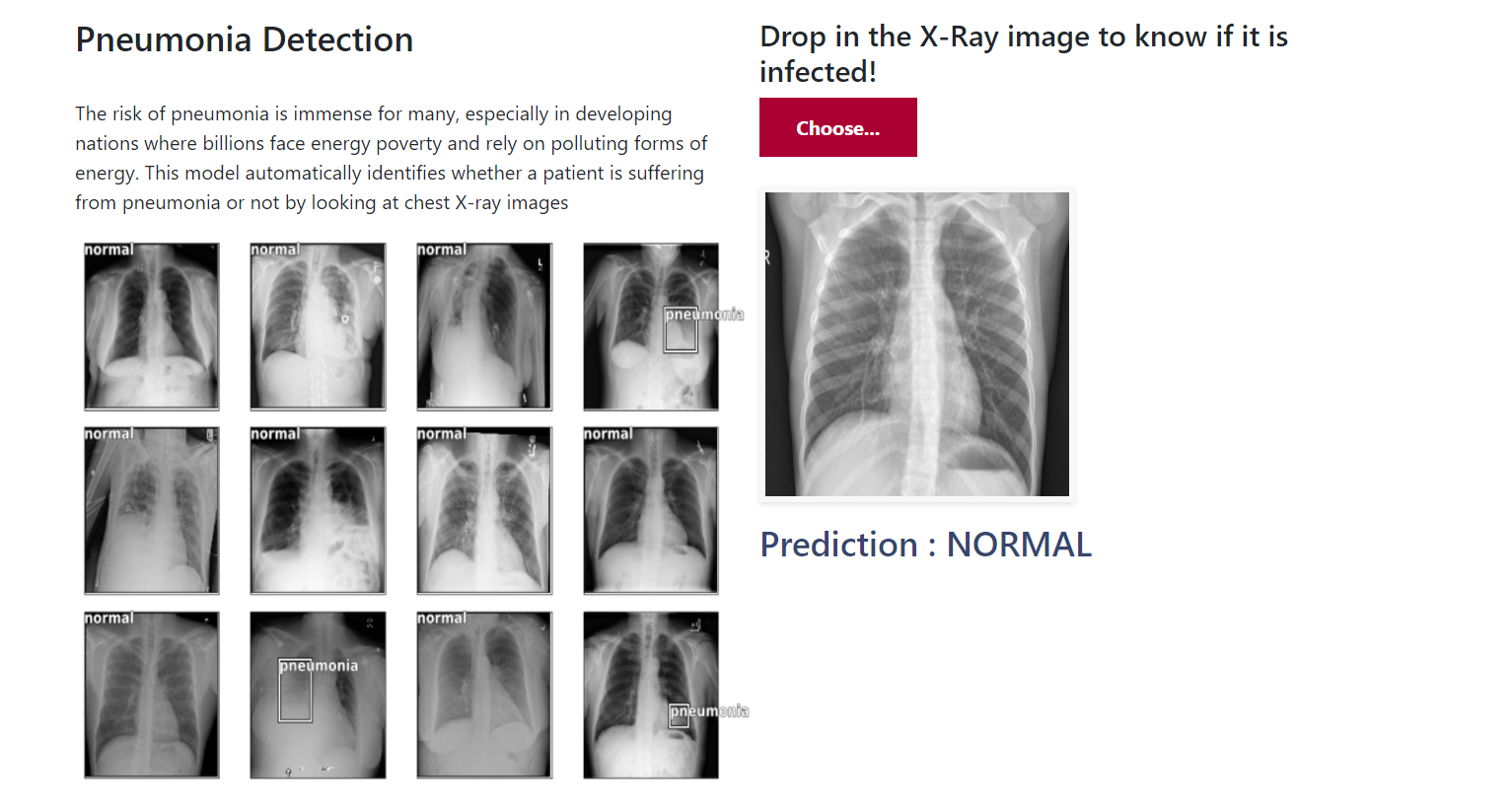
****

**Figure 30: python code**

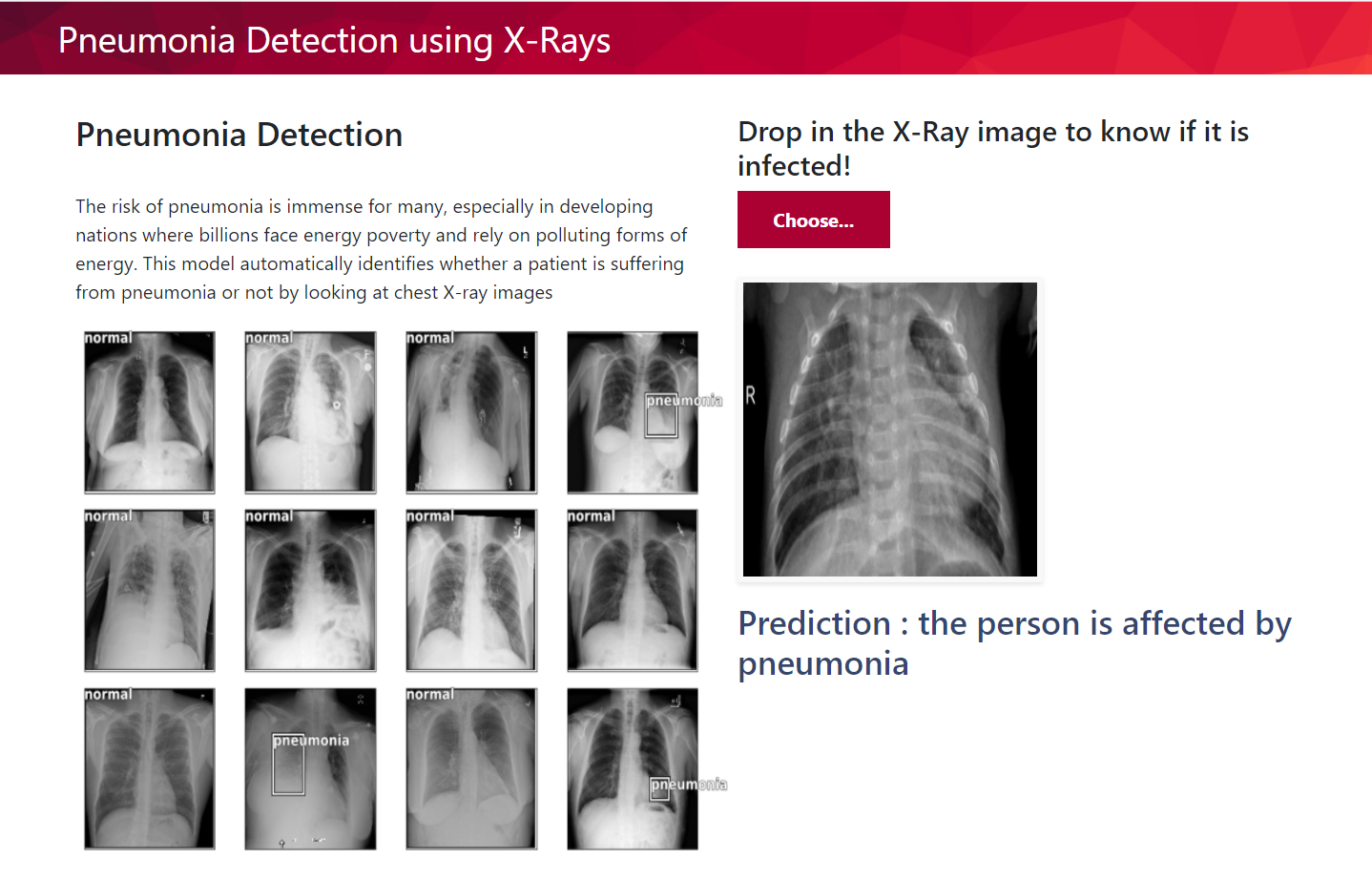
****

**Figure 31: python code**

* 1. **CONCLUSION**

****

**Figure 32: Output predicting that person is safe**

****

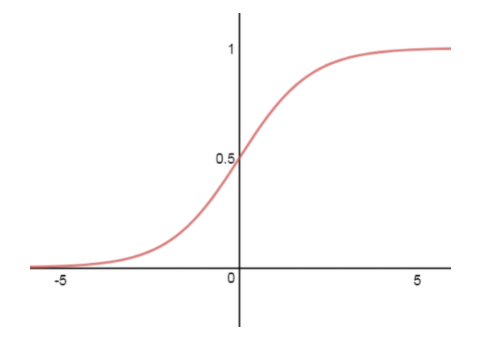
**Figure 33: Output predicting that person is affected by pneumonia**

1. **IMPLEMENTATION**
   1. **INTRODUCTION**

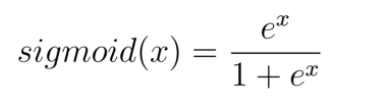
In this section we actually able to know what are the functions we used and how we implemented them in order to classify the images as normal and pneumonia and the detailed explanation of particular functions.

* 1. **Explanation of Key functions**
* **SIGMOID**

Some years ago the most common activation function you would have encountered is the sigmoid function. The sigmoid function maps the incoming inputs to a range between zero and one:

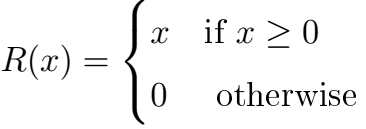


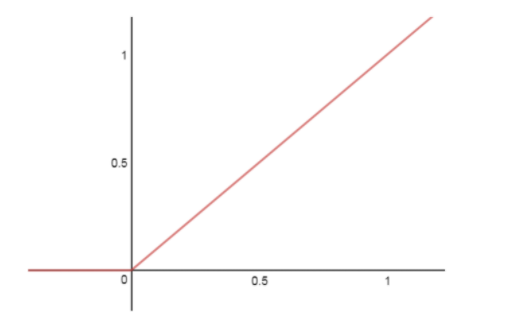
**FIGURE 34: Sigmoid function**

The sigmoid activation function is defined as follows:

### **RECTIFIEDLINEAR UNIT — RELU**

The Rectified Linear Unit (ReLU) has become very popular in the last few years. The activation is simply thresholded at zero: R(x) = max(0,x) or more precisely:





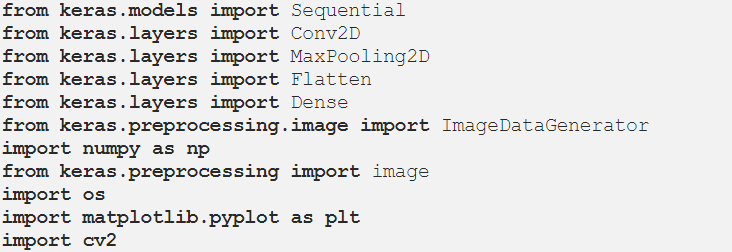
**FIGURE 35: Rectified linear unit**

* **SOFTMAX ACTIVATION FUNCTION**

We apply softmax in the last layer and only when we want the neural network to predict probability scores during classification tasks.Simply speaking, the softmax activation function forces the values of output neurons to take values between zero and one, so they can represent probability scores.

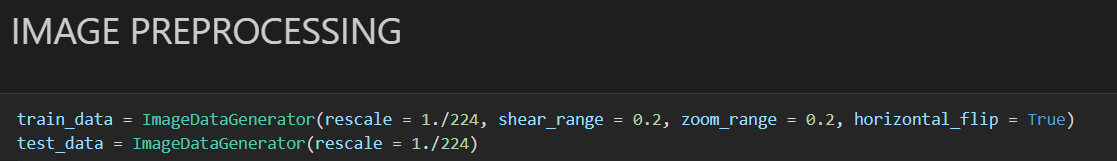
Another thing we must consider is that when we perform the classification of input features into different classes, these classes are mutually exclusive. This means that each feature vector x belongs to only one class. For example, a feature vector that is an image of a dog can’t represent a dog class with a probability of 50 percent and a cat class with a probability of 50 percent. This feature vector must represent the dog class with a probability of 100 percent.

In the case of mutually exclusive classes, the probability scores across all output neurons must sum up to one. Only in this way the neural network represents a proper probability distribution. A counterexample would be a neural network that classifies a dog’s image into the class dog with a probability of 80 percent and with a class cat probability of 60 percent.



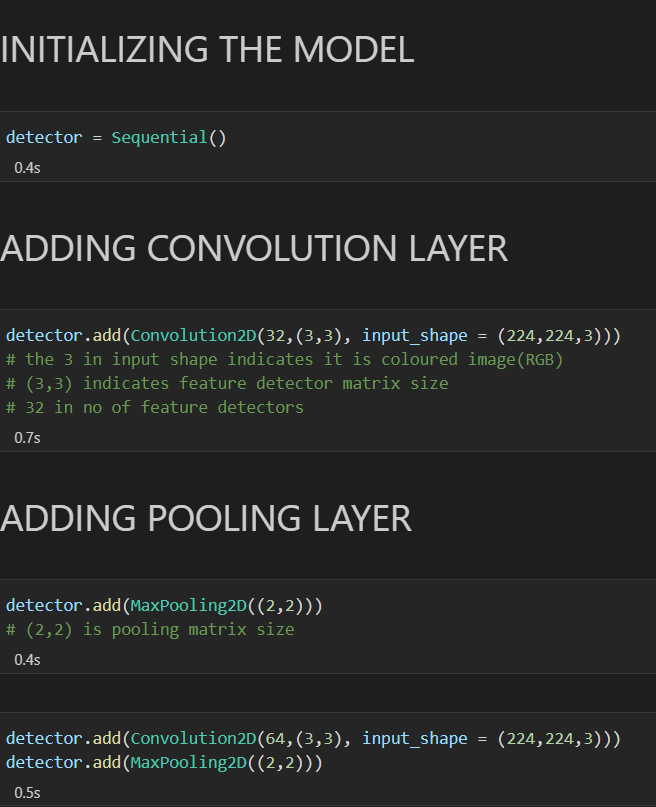
**FIGURE 36: Libraries**

1. The Keras Python library makes creating deep learning models fast and easy. The sequential API allows you to create models layer-by-layer for most problems. It is limited in that it does not allow you to create models that share layers or have multiple inputs or outputs.
2. Keras Conv2D is a 2D Convolution Layer, this layer creates a convolution kernel that is wind with layers input which helps produce a tensor of outputs. (Note:- Kernel: In image processing kernel is a convolution matrix or masks which can be used for blurring, sharpening, embossing, edge detection and more by doing a convolution between a kernel and an image.)
3. MaxPooling2D from keras.layers, which is used for pooling operation. For building this particular neural network, we are using a Maxpooling function, there exist different types of pooling operations like Min Pooling, Mean Pooling, etc. Here in MaxPooling we need the maximum value pixel from the respective region of interest.
4. Flatten from keras.layers, which is used for Flattening. Flattening is the process of converting all the resultant 2-dimensional arrays into a single long continuous linear vector.
5. Dense from keras.layers, which is used to perform the full connection of the neural network
6. Image Data Generator, which Takes a batch of images and applies a series of random transformations to each image in the batch (including random rotation, resizing, shearing, etc.) and then Replacing the original batch with the new, randomly transformed batch for training the CNN.

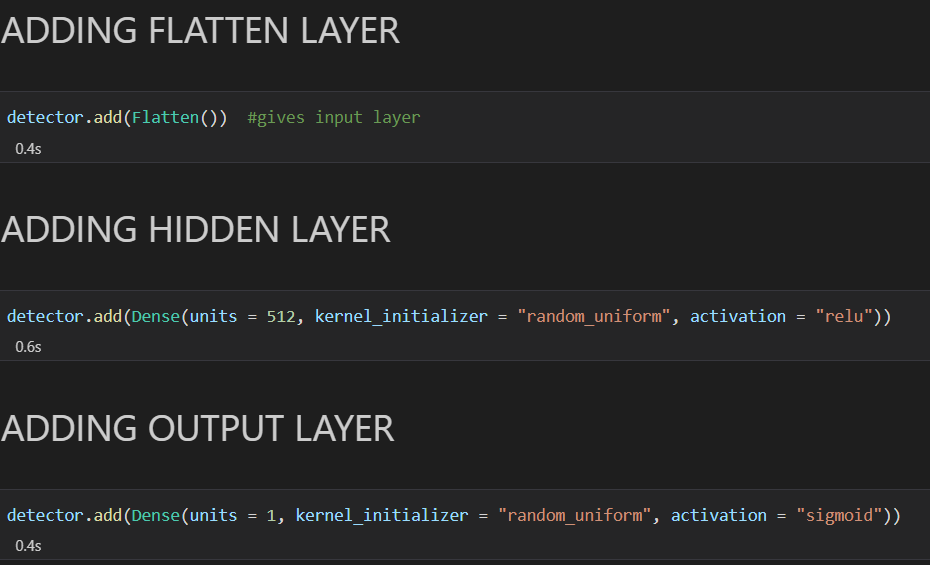


**FIGURE 37: Image preprocessing**

1. For this task we have used the ImageDataGenerator of Keras and have passed the following parameters:
2. **rescale:** rescaling factor. Defaults to None. If None or 0, no rescaling is applied, otherwise we multiply the data by the value provided (after applying all other transformations).
3. **shear\_range**: Float. Shear Intensity (Shear angle in the counter-clockwise direction in degrees). Shearing used to transform the orientation of the image.
4. **zoom\_range**: Float or [lower, upper]. The range for random zoom.
5. **horizontal\_flip**: Boolean. Randomly flip inputs horizontally.
6. **flow\_from\_directory**: Takes the path to a directory, and generates batches of augmented/normalized data.

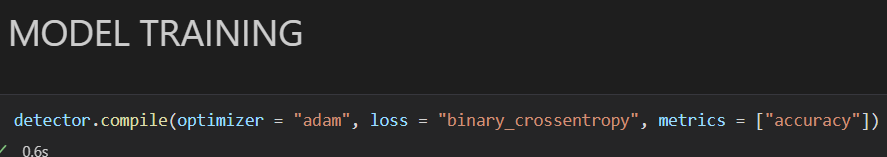


**FIGURE 38: Model initialize and adding convolution layer**



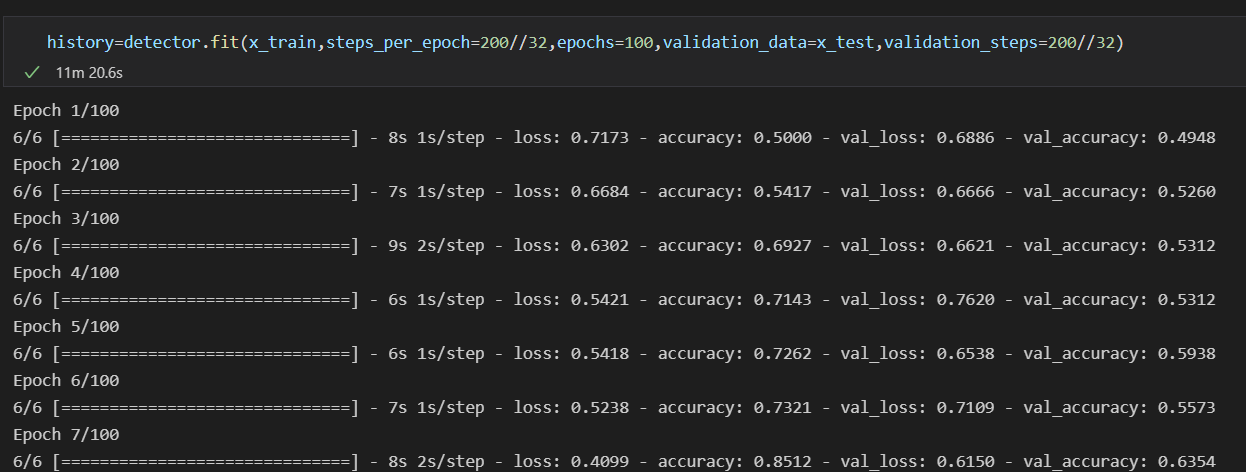
**FIGURE 39: Key functions**

* I then took the detector object and added a convolution layer by using the “Conv2D” function. The Conv2D function is taking 4 arguments:
* the first is filter this is a mandatory Conv2D parameter that defines the numbers of filters that convolutional layers will learn from i.e 32 here, filters are taken to slice through the image and map them one by one and learn different portions of an input image. Imagine a small filter sliding left to right across the image from top to bottom and that moving filter is looking for, say, a dark edge. Each time a match is found, it is mapped out onto an output image.
* the second argument is the shape each filter is going to be i.e 3x3 here,
* the third is the input shape and the type of image(RGB or Black and White)of each image i.e the input image our CNN is going to be taking is of a 64x64 resolution and “3” stands for RGB, which is a color image
* the fourth argument is the activation function we want to use, here ‘relu’ stands for a Rectified Linear Unit function. The activation function is a node that helps to decide if the neuron would fire or not. Relu sets all negative values in the matrix x to zero and all other values are kept constant.
* Now I perform pooling operation on the resultant feature maps I got after the convolution operation is done on an image. The primary aim of a pooling operation is to reduce the size of the images as much as possible.
* Next, I convert all the pooled images into a continuous vector through Flattening. Flattening is a very important step to understand. What we are basically doing here is taking the 2-D array, i.e pooled image pixels and converting them to a one-dimensional single vector.
* Now, to create a fully connected layer I have connected the set of nodes I got after the flattening step, these nodes will act as an input layer to these fully-connected layers. Dense is the function to add a fully connected layer, ‘units’ is where we define the number of nodes that should be present in this hidden layer.
* Then I have initialized the output layer which consists of a single node giving me the respective output
* Once building the CNN model is finished it is time for compiling it:



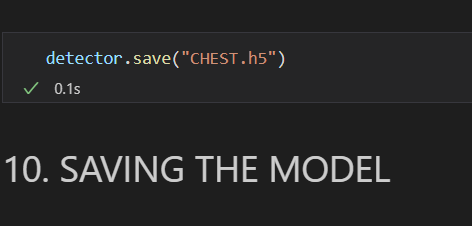
**FIGURE 40: Model training**

* Adam is an optimization algorithm that can be used instead of the classical stochastic gradient descent procedure to update network weights iterative based on training data.
* Cross-entropy loss, or log loss, measure the performance of a classification model whose output is a probability value between 0 and 1. Cross-entropy loss increases as the predicted probability diverge from the actual label.
* Finally, the metrics parameter is to choose the performance metric.

****

**FIGURE 41: Model evaluation**

* In the above code, ‘steps\_per\_epoch’ holds the number of training images, i.e the number of images the training\_set folder contains.
* And ‘epochs’, A single epoch is a single step in training a neural network; in other words, when a neural network is trained on every training samples only in one pass we say that one epoch is finished. So the training process should consist of more than one epoch.
* Once you have finally built and trained your model you can pass images to classifier.predict()(i.e. [modelname].predict()) function and get the predictions.

****

**FIGURE 42: Model saving**

* + You can also save your model for future use by using the detector.save()(i.e. [modelname].save())

1. **APPLICATION**

The following application could be used in a better understanding with if patients past history details are also uploaded, and the accuracy can be increased if the system is fed to a large number and variety of dataset. With various technologies and AI (artificial intelligence) applications coming up in the near future we can add a lot of advantage to the current system and get more realistic results. We can also apply a series of techniques involving scanned images as well as other related data points together for the classification of the dataset. Artificial Intelligence (AI) techniques could also be used to understand the different regions of the breast during classification. We can also treat patients very quickly without doing much process.

1. **ADVANTAGES**

* Helps in medical department such as in laboratories for predicting the pneumonia.
* By the reports we can analyze the stages of pneumonia that patient is suffering from.

1. **FUTURE SCOPE**

In future work, a large-sized dataset of different lung diseases will be collected to train the model to improve the healthcare sectors. Moreover, an energy evaluation measure will be used to compute a real result of the energy consumption of the proposed approach.

1. **CONCLUSION**

Throughout the process of developing the CNN model for Pneumonia prediction, we have built a model from scratch which consists of 5 layers and follows with a fully connected neural network. Then the trained model is evaluated using separate unseen data to avoid bias prediction. As the result, the accuracy of the test dataset reached 96% which indicates a decent model. This mini-project allows a beginner to obtain an overview of how to build a model to solve a real-world problem.

It is no doubt that the predictive model can be improved even better by performing data augmentation or implementing a transfer learning concept which facilitates the model a room for improvement. Therefore, this will be added as further enhancement in the upcoming stories.

Early detection of pneumonia increases the survival rate of patients. Automatic pneumonia detection from CXR images can be achieved using a CAD application. It is one of the medical systems that help to train radiologists to improve their accuracy of diagnosing pneumonia.

Developing a lightweight approach for the automatic detection of pneumonia in energy-efficient medical systems is important to improve the quality of healthcare with reduced cost and time response. In this paper, a lightweight DL approach is presented using a pretrained DenseNet-121-based feature extraction method and DNN-based feature classification method with a random search fine-tuning technique. The DenseNet-121 model is used as a feature extraction method due to its ability to provide the best representation of lung features. The use of a random search facility optimizes the efficiency and accuracy of the DNN model.

A number of experiments were performed on a dataset of CXR images and evaluated by a set of metrics. The results of the evaluation metrics demonstrate that the approach achieved 98.90% accuracy, which is higher than the state-of-the-art techniques and methods. Moreover, the time execution results show that the detection time of the approach is very low, which makes it a lightweight pneumonia detection approach, suitable for applying in energy-efficient medical systems. All results confirmed the effectiveness and efficiency of the proposed approach.

1. **BIBLIOGRAPHY**

* <https://www.hindawi.com/journals/cmmm/2021/8854892/>
* <https://www.hindawi.com/journals/wcmc/2021/5556635/>
* <https://towardsdatascience.com/pneumonia-detection-using-convolutional-neural-network-12b94aeb1206>
* <https://towardsdatascience.com/chest-x-rays-pneumonia-detection-using-convolutional-neural-network-63d6ec2d1dee>
* <https://www.mayoclinic.org/diseases-conditions/pneumonia/symptoms-causes/syc-20354204>
* <https://github.com/sibeltan/pneumonia_detection_CNN/blob/master/Pneumonia_Detection_Presentation.pdf>

lattening layer and fully connected layers. After the input image passes

through the convolutional layer and the pooling layer, it is fed into the ﬂattening

layer. This layer ﬂattens out the input image into a column, further reducing its

computational complexity. This is then fed into the fully connected layer/dense layer.

The fully connected layer [21] has multiple layers, and every node in the ﬁrst layer

is connected to every node in the second layer. Each layer in the fully connected

layer extracts features, and on this basis, the network makes a prediction [22,23].

This process is known as forward propagation. After forward propagation, a cost

function is calculated. It is a measure of performance of a neural network model.

The cost function used in all four models is categorical cross-entropy. After the cost

function is calculated, back propagation takes place. This process is repeated until

the network achieves optimum performance. Adam optimization algorithm has been

used in all four models.

Reducing overﬁtting.

Flattening layer and fully connected layers. After the input image passes

through the convolutional layer and the pooling layer, it is fed into the ﬂattening

layer. This layer ﬂattens out the input image into a column, further reducing its

computational complexity. This is then fed into the fully connected layer/dense layer.

The fully connected layer [21] has multiple layers, and every node in the ﬁrst layer

is connected to every node in the second layer. Each layer in the fully connected

layer extracts features, and on this basis, the network makes a prediction [22,23].

This process is known as forward propagation. After forward propagation, a cost

function is calculated. It is a measure of performance of a neural network model.

The cost function used in all four models is categorical cross-entropy. After the cost

function is calculated, back propagation takes place. This process is repeated until

the network achieves optimum performance. Adam optimization algorithm has been

used in all four models.

Reducing overﬁtting.

Flattening layer and fully connected layers. After the input image passes

through the convolutional layer and the pooling layer, it is fed into the ﬂattening

layer. This layer ﬂattens out the input image into a column, further reducing its

computational complexity. This is then fed into the fully connected layer/dense layer.

The fully connected layer [21] has multiple layers, and every node in the ﬁrst layer

is connected to every node in the second layer. Each layer in the fully connected

layer extracts features, and on this basis, the network makes a prediction [22,23].

This process is known as forward propagation. After forward propagation, a cost

function is calculated. It is a measure of performance of a neural network model.

The cost function used in all four models is categorical cross-entropy. After the cost

function is calculated, back propagation takes place. This process is repeated until

the network achieves optimum performance. Adam optimization algorithm has been

used in all four models.

Reducing overﬁtting.

Flattening layer and fully connected layers. After the input image passes

through the convolutional layer and the pooling layer, it is fed into the ﬂattening

layer. This layer ﬂattens out the input image into a column, further reducing its

computational complexity. This is then fed into the fully connected layer/dense layer.

The fully connected layer [21] has multiple layers, and every node in the ﬁrst layer

is connected to every node in the second layer. Each layer in the fully connected

layer extracts features, and on this basis, the network makes a prediction [22,23].

This process is known as forward propagation. After forward propagation, a cost

function is calculated. It is a measure of performance of a neural network model.

The cost function used in all four models is categorical cross-entropy. After the cost

function is calculated, back propagation takes place. This process is repeated until

the network achieves optimum performance. Adam optimization algorithm has been

used in all four models.

Reducing overﬁt

through the convolutional layer and the pooling layer, it is fed into the ﬂattening

layer. This layer ﬂattens out the input image into a column, further reducing its

computational complexity. This is then fed into the fully connected layer/dense layer.

The fully connected layer [21] has multiple layers, and every node in the ﬁrst layer

is connected to every node in the second layer. Each layer in the fully connected

layer extracts features, and on this basis, the network makes a prediction [22,23].

This process is known as forward propagation. After forward propagation, a cost

function is calculated. It is a measure of performance of a neural network model.

The cost function used in all four models is categorical cross-entropy. After the cost

function is calculated, back propagation takes place. This process is repeated until

the network achieves optimum performance. Adam optimization algorithm has been

used in all four models