**CNN\_Covid:**

**Web Application:**

**Frontend: HTML, CSS, jQuery**

**Backend: Flask**

**Database: MySQL**

**CNN model:**  
To predict Covid-19 from x-ray images

**Dataset:**  
<https://github.com/ieee8023/covid-chestxray-dataset>

Created a basic CNN model suing Keras

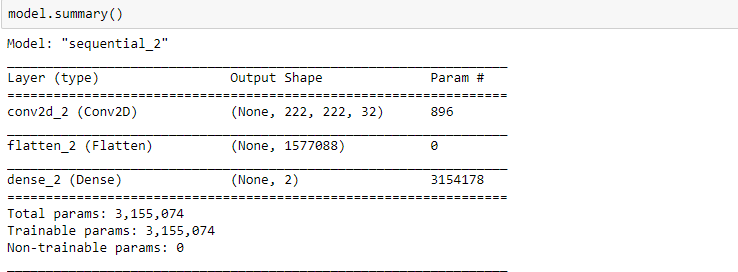
Trained it with x-rays of patients with Covid-19 and without Covid-19

Validated the model on a validation set

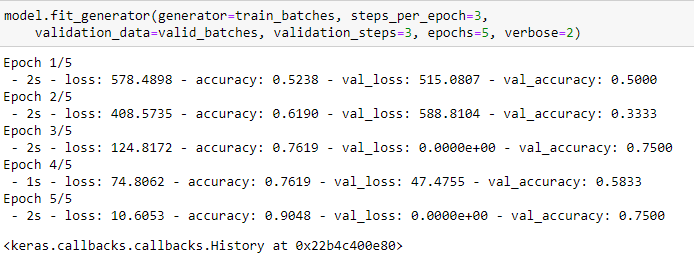
Tested on new data for accuracy

Compiled and saved the model as .hdf5 file for reusability

**Model:**



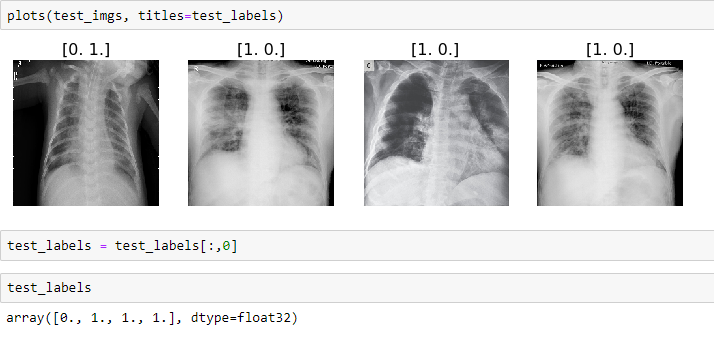
**Validation and Testing:**

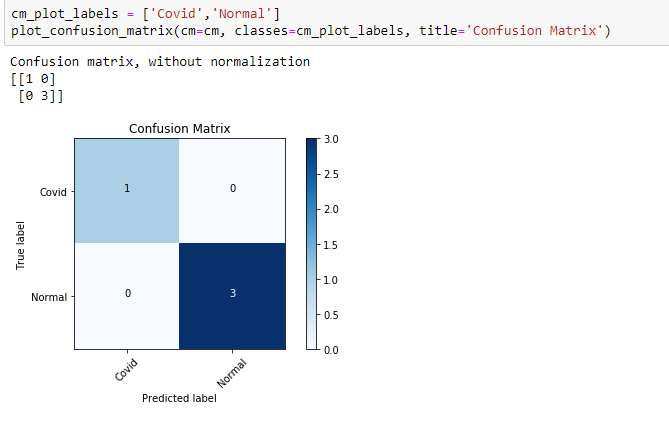


**Results:**

**0 -> Covid**

**1 -> Normal**





**Its proven that the model works with 100% accuracy**

**Working:**

1. User logs into the website
2. Selects the model to use
3. Selects an image to upload
4. Enters the query (What the user wants to know)
5. The image is pre-processed by jQuery to fit the model constraints
6. A message containing (Model, Image, Query) is sent to the server
7. The message data is parsed and relevant model is selected
8. The image is served to the model and result data is sent back to the frontend as JSON
9. In frontend, jQuery processes the json data and displays the results

**Important files:**

CNN\_Covid:

CNN.ipynb: Python notebook for the CNN model

Data: Datasets for the CNN model

Flask\_app: Contains the entire application code

app.py: Entire backend logic

predict\_app.py: Working of CNN model

basic\_cnn\_model.hdf5: Pre-compiled CNN model

static: Contains all the CSS and JS files

templates: Contains all the HTML templates

**MySQL:**

* Created a database: myflaskapp
* Created a table: users
* Users stores the username (primary key) and hashed password
* Backend creates a new row for registrations
* Backend checks a specific row for logins

**Future work:**

**Adding models:**

More models can added to the backend file directly or made as API’s which the app can call and fetch data

**Relating input query and model:**

**Database:**

Migrate to Firebase for better functionality when deployed

**App:**

**ReactNative + Flask API + Firebase**

* Each model is converted to a Flask API
* The API’s are hosted on AWS or Heroku or any other endpoints
* App Frontend and Backend is created using ReactNative
* The app has the same functionalities of the web application
* Th app calls the API to fetch results and display them
* The app connects to Firebase for user authentication