**CORONA HACK CHEST X-RAY**

A Project Report

Submitted in the partial fulfilment on the requirements for

the award of the degree of

Bachelor of technology

In

Department of Computer Science And Engineering

By

L. Vamsi krishna (2010030410)

T. SAINATH (2010030383)

V. ABHIRAM (2010030174)

MITESH CHANDRA (2010039004)

SIMON SUMANTH (2010030343)

Under the supervision of

**DR. ANAL PAUL**

ASSISTANT PROFESSOR



Department Of Computer Science And Engineering

Kl University Hyderabad, Aziz Nagar, Moinabad Road, Hyderabad – 500 075, Telangana, India.

**Declaration**

The Project Report entitled “CORONA HACK CHEST X-RAY” is a record of bonafide work of <L. Vamsi Krishna(2010030410) T.SAINATH(2010030174) MITESH CNAHDRA(2010039004) V.ABHIRAM(2010030383) SIMON SUMANTH (2010030343)>, submitted in partial fulfilment for the award of B. Tech in the Department of Computer Science and Engineering to the K L University, Hyderabad. The results embodied in this report have not been copied from any other Departments/University/Institute

<signature of the students>

SIMON SUMANTH

MITESH CHANDRA

V ABHIRAM

T SAINATH REDDY

L VAMSI KRISHNA

## Certificate

This is to certify that the Project Report entitled “CORONA HACK CHEST X-RAY” is being submitted by SIMON SUMANTH(2010030343) MITESH (2010039004) V ABHIRAM(2010030383)T.SAINATH(2010030174)L.VAMSI KRISHNA(2010030410) submitted in partial fulfillment for the award of B. Tech in CORONA HACK CHEST X-RAY to the K L University, Hyderabad is a record of bonafide work carried out under our guidance and supervision.

## Signature of the Supervisor

## 

Dr. Anal Paul

**Signature of the HOD Signature of the External Examiner**

**Acknowledgedment**

First and foremost, we thank the lord almighty for all his grace & mercy showered upon us, for completing this project successfully.

We take grateful opportunity to thank our beloved Founder and Chairman who has given constant encouragement during our course and motivated us to do this project. We are grateful to our Principal **Dr. L. Koteswara Rao** who has been constantly bearing the torch for all the curricular activities undertaken by us.

We pay our grateful acknowledgement & sincere thanks to our Head of the Department **Dr. Chiranjeevi Manike** for her exemplary guidance, monitoring and constant encouragement throughout the course of the project. We thank **Dr. Anal Paul** of our department who has supported throughout this project holding a position of supervisor.

We whole heartedly thank all the teaching and non-teaching staff of our department without whom we won’t have made this project a reality. We would like to extend our sincere thanks especially to our parent, our family members and friends who have supported us to make this project a grand success.

**Abstract**

Novel coronavirus disease (nCOVID-19) is the most challenging problem for the world. The disease is caused by severe acute respiratory syndrome coronavirus-2 (SARS-COV-2), leading to high morbidity and mortality worldwide. The study reveals that infected patients exhibit distinct radiographic visual characteristics along with fever, dry cough, fatigue, dyspnea, etc. Chest X-Ray (CXR) is one of the important, non-invasive clinical adjuncts that play an essential role in the detection of such visual responses associated with SARS-COV-2 infection. However, the limited availability of expert radiologists to interpret the CXR images and subtle appearance of disease radiographic responses remains the biggest bottlenecks in manual diagnosis. In this study, we present an automatic COVID screening (ACoS) system that uses radiomic texture descriptors extracted from CXR images to identify the normal, suspected, and nCOVID-19 infected patients. The proposed system uses two-phase classification approach (normal vs. abnormal and nCOVID-19 vs. pneumonia) using majority vote based classifier ensemble of five benchmark supervised classification algorithms. The training-testing and validation of the ACoS system are performed using 2088 (696 normal, 696 pneumonia and 696 nCOVID-19) and 258 (86 images of each category) CXR images, respectively. The obtained validation results for phase-I (accuracy (ACC) = 98.062%, area under curve (AUC) = 0.956) and phase-II (ACC = 91.329% and AUC = 0.831) show the promising performance of the proposed system. Further, the Friedman post-hoc multiple comparisons and z-test statistics reveals that the results of ACoS system are statistically significant. Finally, the obtained performance is compared with the existing state-of-the-art methods.

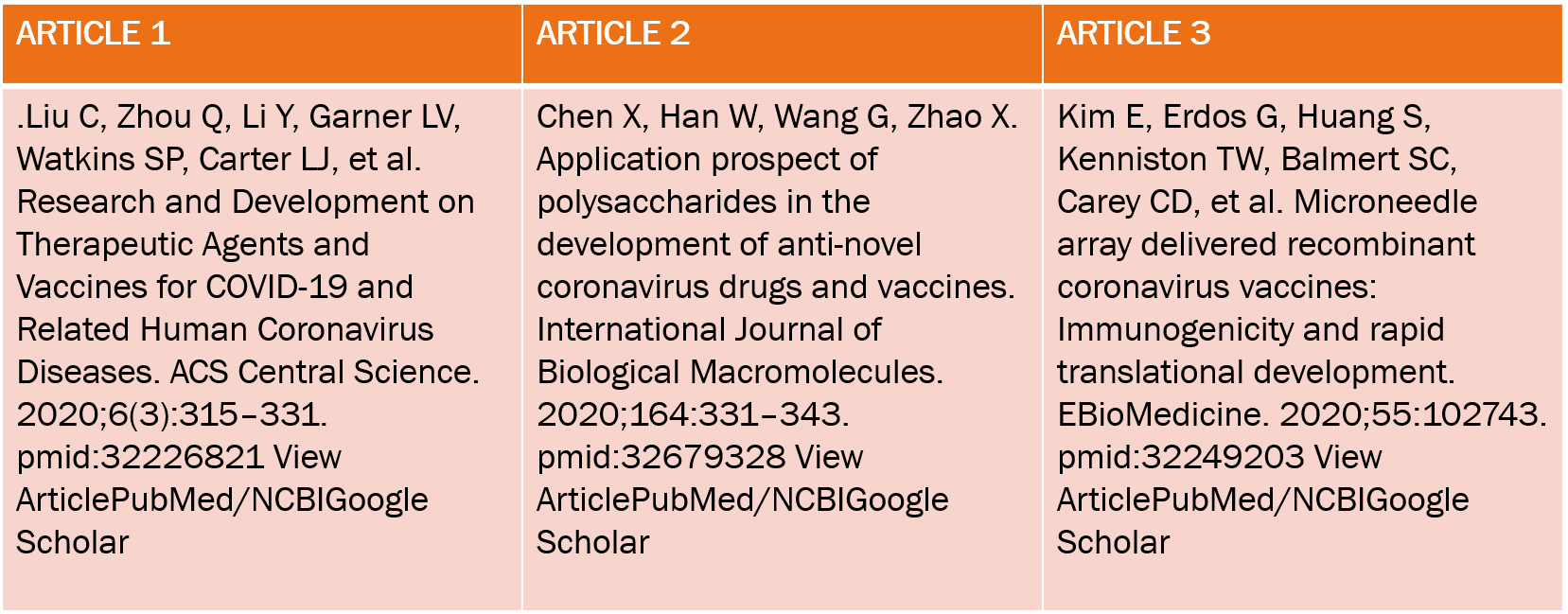
**Table of contents**

* 1. Introduction
  2. Literature Survey
  3. Hardware & Software requirements
  4. Methodology
  5. Flowchart
  6. Implementation
  7. Results Discussion
  8. Conclusion and Future Work
  9. References

**Introduction**

Coronavirus related respiratory illness usually manifests clinically as pneumonia with predominant imaging findings of an atypical or organizing pneumonia. Plain radiography is very helpful for COVID-19 disease assessment and follow-up. It gives an accurate insight into the disease course.We aimed to determine the COVID-19 disease course using chest X-ray (CXR) scoring system and correlate these with patients’ age, and outcome.Corona - COVID19 virus affects the respiratory system of healthy individual & Chest X -Ray is one of the important imaging methods to identify the corona virus.With the Chest X - Ray dataset, Develop a Machine Learning Model to classify the X Rays of Healthy vs Pneumonia (Corona) affected patients & this model powers the AI application to test the Corona Virus in Faster Phase.

**Literature Survey**



Three CNN architectures (ResNet50, InceptionV3, and InceptionRes-NetV2) were evaluated in relation to COVID-19 identification, utilizing a database of just 50 controls and 50 COVID-19 cases. ResNet50 achieved the highest accuracy of 98%.

a successful performance in diagnosis accuracy found in this research demonstrates that deep CNNs could correctly and efficiently distinguish 21,152 normal and abnormal chest radiographs. The CNN model pre-trained on datasets of adult patients and fine-tuned on pediatric patients obtained an accuracy of 94.64%, a sensitivity of 96.5% and a specificity of 92.86% for normal versus pneumonia categorization.

**HARDWARE AND SOFTWARE REQUIREMENTS**

Hardware requirements:

System: Pentium i5 processor

Hard Disk: 512GB

Monitor: 16.5inch LED

Input Devices: Keyboard, Mouse.

Software requirements:

Operating System: Windows 11

Language: Python

Software used: Vscode

**METHODOLOGY**

### **Dataset Preparation**

A chest X-ray database was used to experiment with this study. This database is currently one of the popular public X-ray databases, containing 3616 COVID-19 cases along with 10,192 healthy, 6012 lung opacity and 1345 viral pneumonia images. However, only COVID-19 (3616) and healthy (10,192) X-ray images were extracted for this study. As a result, the dataset includes studies of COVID-19 and healthy individuals with a matrix resolution of 299 × 299 system for scene-text removal, was used to remove annotations from certain images. EnsNet is capable of automatically removing all of the text or annotation from an image without any prior knowledge . Data augmentation and image enhancement techniques are performed to enhance the quantity and variety of images given to the classifier for classification. Image augmentations used include horizontal flip, rotation, width shift and height shift on all the extracted data from the original dataset.

### **Model Selection**

One of the main goals of this research is to obtain appropriate classification results utilizing freely available data (increased to high volume data by using enhancement techniques) with the combined transfer learning models. This research was undertaken to choose a CNN-based deep learning model that is appropriate for COVID-19 image classification investigation. The primary aim is to propose a modified novel deep-learning-based CNN model to gain the highest accuracy on a large volume of chest X-ray data with minimal compilation time and compare the modified novel approach (accuracy, efficiency, compilation time) with existing deep learning models on the same dataset.

1. VGG19 and VGG16

The Visual Geometry Group is abbreviated as VGG. VGG16 is built using multiple 33 kernel-sized filters sequentially (11 and 5 in the first and second convolutional layers, respectively). VGG’s input is set to a 224 × 244 RGB picture. The VGG-19 convolutional neural network was trained using over a million pictures from the ImageNet database. The network has a depth of 19 layers and is capable of classifying images of multiple classes. The VGG architectures’ primary concept is to keep the convolution size modest and constant while designing an extremely deep network.

InceptionV3

InceptionV3 makes use of label smoothing, factorized 7 × 7 convolutions, and an auxiliary classifier to transmit label information down the network, as well as batch normalization for sidehead layers. It features smaller convolutions for quicker training and lower grid size to overcome computational cost constraints. Numerous optimization methods have been proposed for an InceptionV3 model in order to relax the restrictions and facilitate model adaptability. Factorized convolutions, regularization, dimension reduction, and parallelized calculations are all included in the methods.

ResNet50 and 101

ResNet50’s architecture is divided into 4 stages. The network may accept an input image with a height, width of multiples of 32, and channel width. The network may accept an input image with a height, width of multiples of 32, and channel width Each ResNet architecture conducts initial convolution and max-pooling with a kernel size of 7 × 7 and 3 × 3, respectively. Each 2-layer block is replaced with this 3-layer bottleneck block in the 34-layer net, resulting in a 50-layer ResNet. A 101-layer ResNet is created by adding additional 3-layer blocks.

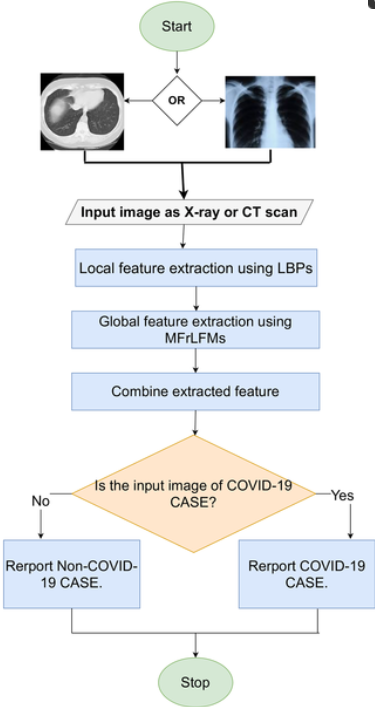
GoogLeNet

GoogLeNet is a deep convolutional neural network with 22 layers and almost 12× fewer parameters compared to Inception architecture. However, by adding more layers, the number of parameters grows, and the network may overfit. The pre-trained network accepts images with a resolution of 224 × 224. In GoogLeNet, global average pooling was utilized instead of a fully linked layer. The architecture makes use of the Activation, AveragePooling2D, and Dense layers.

MobileNetV2

MobileNetV2 introduces a new module with an inverted residual structure. With MobileNetV2, state-of-the-art object recognition and semantic segmentation are accomplished. MobileNetV2’s architecture begins with a fully convolutional layer with 32 filters and 19 residual bottleneck layers. Typically, the network requires 300 million multiply-add operations and utilizes 3.4 million parameters. Accuracy is increased by removing ReLU6 from the output of each bottleneck module.

**Flow Chart**



**IMPLEMENTATION**

**Implementation of detecting Carona using chest x-ray :**

Code :

from flask import Flask, render\_template, request, session, redirect, url\_for, flash

import os

from werkzeug.utils import secure\_filename

from tensorflow.keras.models import load\_model

import matplotlib.pyplot as plt

import cv2

import numpy as np

UPLOAD\_FOLDER = './flask app/assets/images'

ALLOWED\_EXTENSIONS = set(['png', 'jpg', 'jpeg', 'gif'])

# Create Database if it doesnt exist

app = Flask(name,static\_url\_path='/assets',

static\_folder='./flask app/assets',

template\_folder='./flask app')

app.config['UPLOAD\_FOLDER'] = UPLOAD\_FOLDER

@app.route('/')

def root():

return render\_template('index.html')

@app.route('/index.html')

def index():

return render\_template('index.html')

@app.route('/contact.html')

def contact():

return render\_template('contact.html')

@app.route('/news.html')

def news():

return render\_template('news.html')

@app.route('/about.html')

def about():

return render\_template('about.html')

@app.route('/faqs.html')

def faqs():

return render\_template('faqs.html')

@app.route('/prevention.html')

def prevention():

return render\_template('prevention.html')

@app.route('/upload.html')

def upload():

return render\_template('upload.html')

@app.route('/upload\_chest.html')

def upload\_chest():

return render\_template('upload\_chest.html')

@app.route('/upload\_ct.html')

def upload\_ct():

return render\_template('upload\_ct.html')

@app.route('/uploaded\_chest', methods = ['POST', 'GET'])

def uploaded\_chest():

if request.method == 'POST':

# check if the post request has the file part

if 'file' not in request.files:

flash('No file part')

return redirect(request.url)

file = request.files['file']

# if user does not select file, browser also

# submit a empty part without filename

if file.filename == '':

flash('No selected file')

return redirect(request.url)

if file:

# filename = secure\_filename(file.filename)

file.save(os.path.join(app.config['UPLOAD\_FOLDER'], 'upload\_chest.jpg'))

resnet\_chest = load\_model('models/resnet\_chest.h5')

vgg\_chest = load\_model('models/vgg\_chest.h5')

inception\_chest = load\_model('models/inceptionv3\_chest.h5')

xception\_chest = load\_model('models/xception\_chest.h5')

image = cv2.imread('./flask app/assets/images/upload\_chest.jpg') # read file

image = cv2.cvtColor(image, cv2.COLOR\_BGR2RGB) # arrange format as per keras

image = cv2.resize(image,(224,224))

image = np.array(image) / 255

image = np.expand\_dims(image, axis=0)

resnet\_pred = resnet\_chest.predict(image)

probability = resnet\_pred[0]

print("Resnet Predictions:")

if probability[0] > 0.5:

resnet\_chest\_pred = str('%.2f' % (probability[0]\*100) + '% COVID')

else:

resnet\_chest\_pred = str('%.2f' % ((1-probability[0])\*100) + '% NonCOVID')

print(resnet\_chest\_pred)

vgg\_pred = vgg\_chest.predict(image)

probability = vgg\_pred[0]

print("VGG Predictions:")

if probability[0] > 0.5:

vgg\_chest\_pred = str('%.2f' % (probability[0]\*100) + '% COVID')

else:

vgg\_chest\_pred = str('%.2f' % ((1-probability[0])\*100) + '% NonCOVID')

print(vgg\_chest\_pred)

inception\_pred = inception\_chest.predict(image)

probability = inception\_pred[0]

print("Inception Predictions:")

if probability[0] > 0.5:

inception\_chest\_pred = str('%.2f' % (probability[0]\*100) + '% COVID')

else:

inception\_chest\_pred = str('%.2f' % ((1-probability[0])\*100) + '% NonCOVID')

print(inception\_chest\_pred)

xception\_pred = xception\_chest.predict(image)

probability = xception\_pred[0]

print("Xception Predictions:")

if probability[0] > 0.5:

xception\_chest\_pred = str('%.2f' % (probability[0]\*100) + '% COVID')

else:

xception\_chest\_pred = str('%.2f' % ((1-probability[0])\*100) + '% NonCOVID')

print(xception\_chest\_pred)

return render\_template('results\_chest.html',resnet\_chest\_pred=resnet\_chest\_pred,vgg\_chest\_pred=vgg\_chest\_pred,inception\_chest\_pred=inception\_chest\_pred,xception\_chest\_pred=xception\_chest\_pred)

@app.route('/uploaded\_ct', methods = ['POST', 'GET'])

def uploaded\_ct():

if request.method == 'POST':

# check if the post request has the file part

if 'file' not in request.files:

flash('No file part')

return redirect(request.url)

file = request.files['file']

# if user does not select file, browser also

# submit a empty part without filename

if file.filename == '':

flash('No selected file')

return redirect(request.url)

if file:

# filename = secure\_filename(file.filename)

file.save(os.path.join(app.config['UPLOAD\_FOLDER'], 'upload\_ct.jpg'))

resnet\_ct = load\_model('models/resnet\_ct.h5')

vgg\_ct = load\_model('models/vgg\_ct.h5')

inception\_ct = load\_model('models/inception\_ct.h5')

xception\_ct = load\_model('models/xception\_ct.h5')

image = cv2.imread('./flask app/assets/images/upload\_ct.jpg') # read file

image = cv2.cvtColor(image, cv2.COLOR\_BGR2RGB) # arrange format as per keras

image = cv2.resize(image,(224,224))

image = np.array(image) / 255

image = np.expand\_dims(image, axis=0)

resnet\_pred = resnet\_ct.predict(image)

probability = resnet\_pred[0]

print("Resnet Predictions:")

if probability[0] > 0.5:

resnet\_ct\_pred = str('%.2f' % (probability[0]\*100) + '% COVID')

else:

resnet\_ct\_pred = str('%.2f' % ((1-probability[0])\*100) + '% NonCOVID')

print(resnet\_ct\_pred)

vgg\_pred = vgg\_ct.predict(image)

probability = vgg\_pred[0]

print("VGG Predictions:")

if probability[0] > 0.5:

vgg\_ct\_pred = str('%.2f' % (probability[0]\*100) + '% COVID')

else:

vgg\_ct\_pred = str('%.2f' % ((1-probability[0])\*100) + '% NonCOVID')

print(vgg\_ct\_pred)

inception\_pred = inception\_ct.predict(image)

probability = inception\_pred[0]

print("Inception Predictions:")

if probability[0] > 0.5:

inception\_ct\_pred = str('%.2f' % (probability[0]\*100) + '% COVID')

else:

inception\_ct\_pred = str('%.2f' % ((1-probability[0])\*100) + '% NonCOVID')

print(inception\_ct\_pred)

xception\_pred = xception\_ct.predict(image)

probability = xception\_pred[0]

print("Xception Predictions:")

if probability[0] > 0.5:

xception\_ct\_pred = str('%.2f' % (probability[0]\*100) + '% COVID')

else:

xception\_ct\_pred = str('%.2f' % ((1-probability[0])\*100) + '% NonCOVID')

print(xception\_ct\_pred)

return render\_template('results\_ct.html',resnet\_ct\_pred=resnet\_ct\_pred,vgg\_ct\_pred=vgg\_ct\_pred,inception\_ct\_pred=inception\_ct\_pred,xception\_ct\_pred=xception\_ct\_pred)

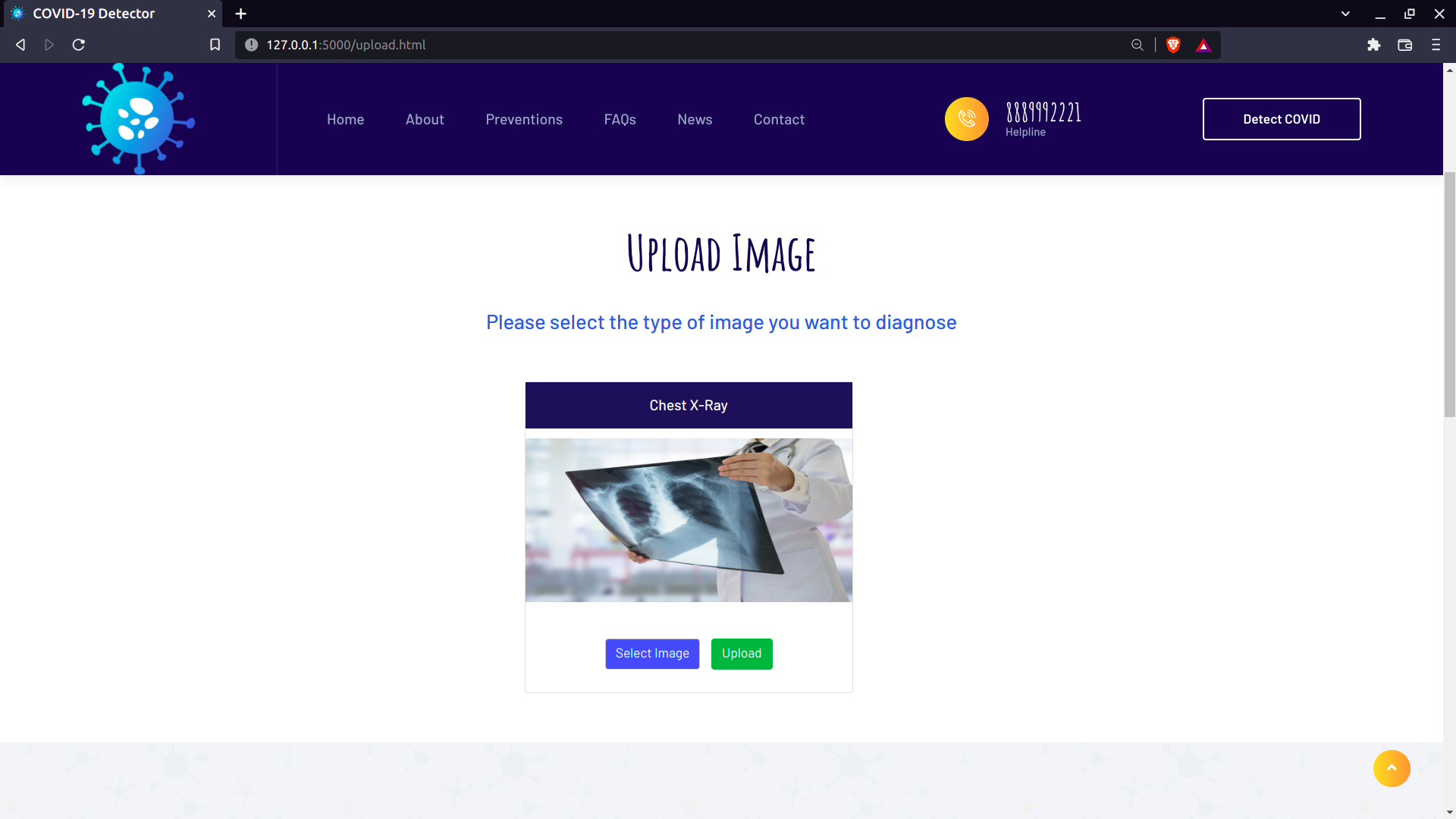
if name == 'main':

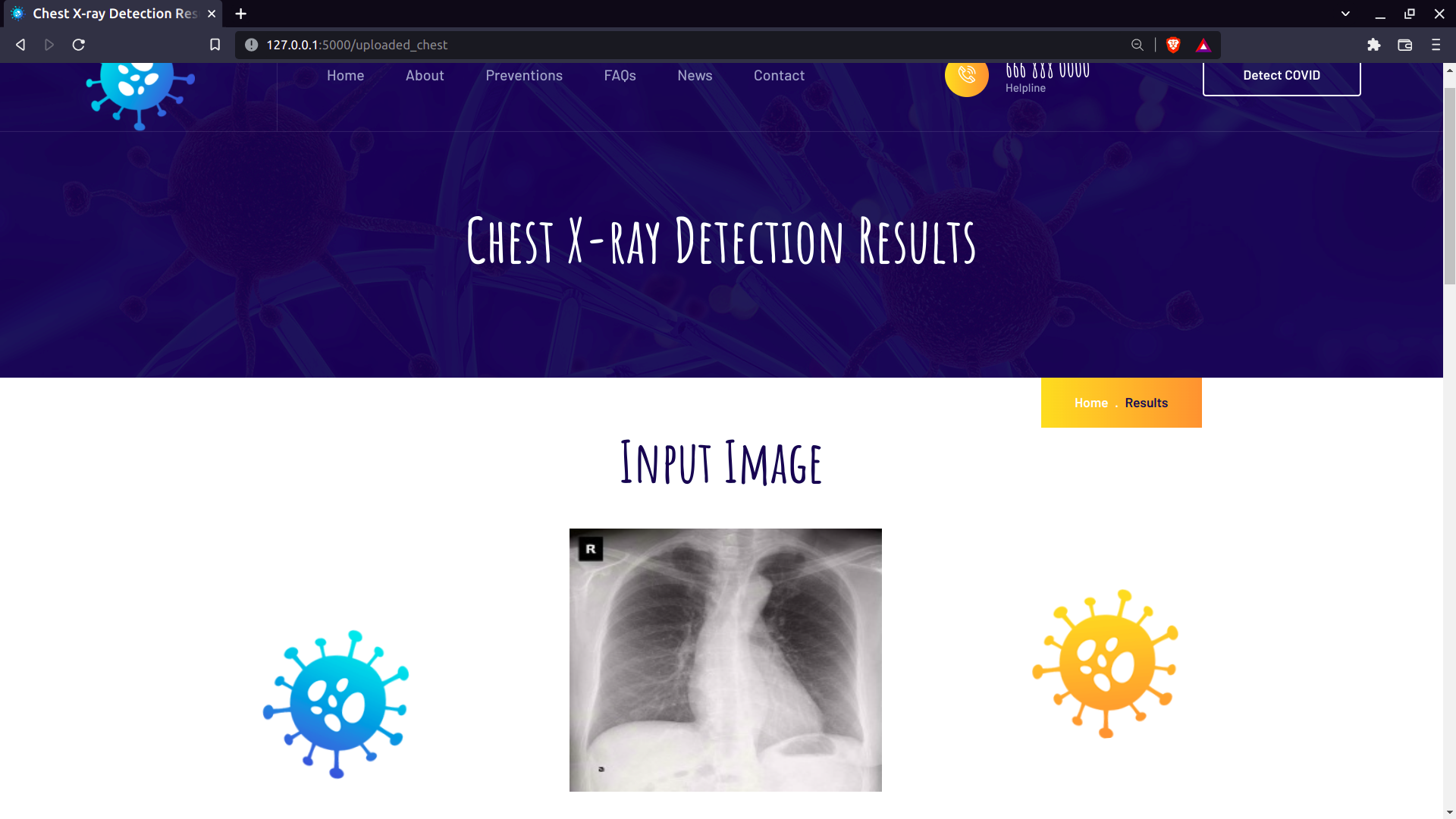
app.secret\_key = ".."

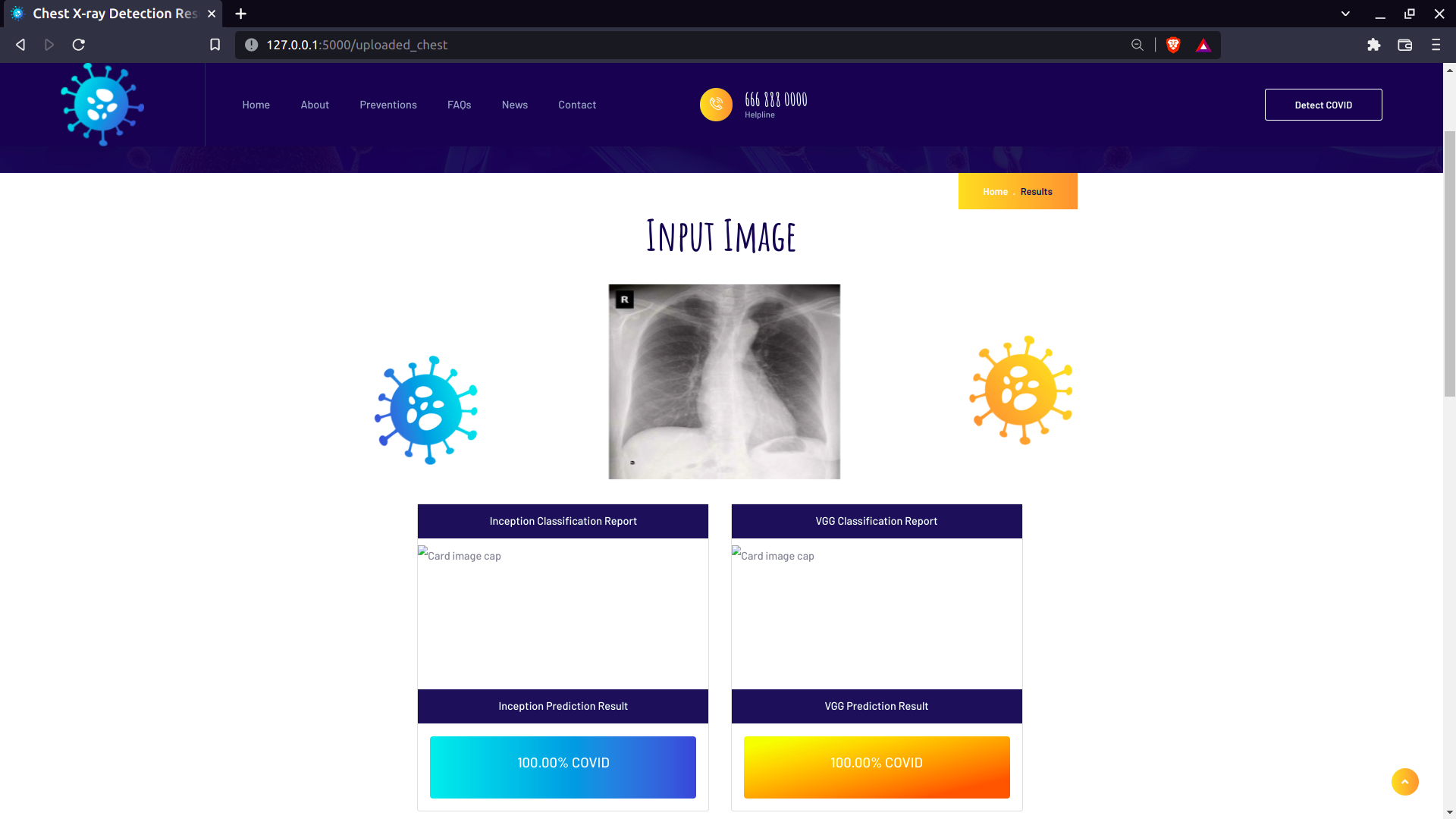
app.run()

**Results Discussion**









**Conclusion and Future Work**

We reported a deep learning framework for COVID-19 detec tion from Chest X-ray images, by fine-tuning four pre-trained convolutional models (ResNet18, ResNet50, SqueezeNet, and DenseNet-121) on our training set. We prepared a dataset of around 5k images, called COVID-Xray-5k (using images from two datasets), with the help of a board-certified radiologist to confirm the COVID-19 labels. We make this dataset publicly available for the research community to use as a benchmark

As future work, the proposed method could be implemented on a dataset with more classes of pulmonary diseases such as asthma, chronic obstructive pulmonary disease, pulmonary fibrosis, pneumonia, lung cancer and COVID-19.

Additionally, from the literature review, it was observed that there is a lack of proper feature extraction processes from image data. Therefore, a feature extraction technique will also be included in future work.

**References**

<https://www.worldometers.info/coronavirus/>

Wang. Wenling, Yanli Xu. Ruqin Gao, Roujian Lu. Kai Han, Guizhen Wu, and Wenjie Tan. "Detection of SARS-CoV-2 in Dif ferent Types of Clinical Specimens." Jama (2020).

Yang, Yang, Minghui Yang, Chenguang Shen, Fuxiang Wang, Jing Yuan, Jinxiu Li, Mingxia Zhang et al. "Laboratory diagnosis and monitoring the viral shedding of 2019-nCoV infections." medRxiv

Ai, Tao, Zhenlu Yang. Hongyan Hou, Chenao Zhan, Chong Chen, Wenzhi Lv, Qian Tao, Ziyong Sun, and Liming Xia. "Correla tion of chest CT and RT-PCR testing in coronavirus disease 2019 (COVID-19) in China: a report of 1014 cases." Radiology (2020): 200642.

Kanne, Jeffrey P., Brent P. Little, Jonathan H. Chung. BrettM. Elicker, and Loren H. Ketai. "Essentials for radiologists on COVID-19: an updateradiology scientific expert panel." Radiology (2020): 200527.

Kong. Weifang, and Prachi P. Agarwal. "Chest imaging appearance of COVID-19 infection." Radiology: Cardiothoracic Imaging 2, no. 1 (2020): e200028.

Hansell, David M., Alexander A. Bankier, Heber MacMahon,Theresa C. McLoud, Nestor L. Muller, and Jacques Remy. "Fleischner Society: glossary of terms for thoracic imaging." Radiology 246, no. 3 (2008): 697-722.

Rodrigues, J.C.L. et al. An update on COVID-19 for the radiol ogist - A British society of Thoracic Imaging statement (2020) Clinical Radiology.

https://github.com/icee8023/covid-chestxray-dataset [10] Cohen, Joseph Paul, Paul Morrison, and Lan Dao, "COVID-19 image data collection." arXiv preprint arXiv:2003.11597, 2020.