Covid-19 Detection Using Lungs X-Rays

PROJECT REPORT

Artificial Intelligence

June 2023

By

Name	Registration Number
KAUSHIK GUPTA	20BCE0823
AMOGH KUMAR MISHRA	20BCE2347
VAIBHAV RAI	20BCE2651
KUMAR SATYAM	20BCE2383

TABLE OF CONTENTS

1. Introduction

1.1 Overview

COVID-19 (coronavirus disease 2019) is an infectious disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), which is a strain of coronavirus. The disease was officially announced as a pandemic by the World Health Organization (WHO) on 11 March 2020. Given spikes in new COVID-19 cases and the re-opening of daily activities around the world, the demand for curbing the pandemic is to be more emphasized. Medical images and artificial intelligence (AI) have been found useful for rapid assessment to provide treatment of COVID-19 infected patients. The PCR test may take several hours to become available, information revealed from the chest X-ray plays an important role in a rapid clinical assessment. This means if the clinical condition and the chest X-ray are normal, the patient is sent home while awaiting the results of the etiological test. But if the X-ray shows pathological findings, the suspected patient will be admitted to the hospital for close monitoring. Chest X-ray data have been found to be very promising for assessing COVID-19 patients, especially for resolving emergency-department and urgent-care-center overcapacity. Deep-learning (DL) methods in artificial intelligence (AI) play a dominant role as high-performance classifiers in the detection of the disease using chest X-rays.

One of the biggest challenges following the Covid-19 pandemic is the detection of the disease in patients. To address this challenge, we have been using the Deep Learning Algorithm to build an image recognition model that can detect the presence of Covid-19 from an X-Ray or CT-Scan image of a patient's lungs.

1.2 Purpose

The purpose of the "COVID-19 Detection Using Lung X-rays with CNN" project is to develop an automated system that can assist in the detection of COVID-19 cases using lung X-ray images. By leveraging deep learning techniques, specifically Convolutional Neural Networks (CNNs), the project aims to create a tool that can analyze X-ray images and accurately classify them as COVID-19-positive or COVID-19-negative.

2. Literature Survey

The outbreak of the COVID-19 pandemic has led to an urgent need for effective and efficient diagnostic methods. Chest X-rays have emerged as a potential tool for the detection of COVID-19 due to their widespread availability and low cost. Convolutional Neural Networks (CNNs), a class of deep learning algorithms, have shown promise in automating the detection process. This literature survey aims to explore the various studies conducted on COVID-19 detection using chest X-ray and CNN techniques.

2.1 Existing Problem

Here's a tabular literature survey on COVID-19 detection using chest X-ray and CNN techniques:

Study	Authors	Year	Methodology	Key Findings
Apostolopoulos et al.	Apostolopoulos, I.D., Mpesiana, T.A.	2020	Transfer learning with CNNs	Automatic detection of COVID-19 from X-ray images using transfer learning achieved promising results.
COVID-net	Wang, L., et al.	2020	Customized CNN architecture	COVID-net, a tailored CNN architecture, demonstrated effective detection of COVID-19 cases from chest X-ray images.
Sethy and Behera	Sethy, P.K., Behera, S.K.	2020	Deep features with SVM	Deep features extracted from CNNs combined with SVM showed potential for COVID-19 detection from chest X-ray images.
Covidx-net	Hemdan, E.E., et al.	2020	Framework with multiple deep learning classifiers	Covidx-net, utilizing various deep learning classifiers including CNNs, provided a comprehensive framework for COVID-19 diagnosis from X-ray images.

Study	Authors	Year	Methodology	Key Findings
Comparative study	Apostolopoulos, I.D., et al.	2020	Comparison of deep learning techniques	Comparative analysis of various deep learning techniques, including CNNs, revealed their effectiveness for chest X-ray classification in COVID-19 cases.
Chestx-ray8	Wang, S., et al.	2017	Database and benchmarks	The Chestx-ray8 database facilitated the benchmarking of weakly-supervised classification and localization of thorax diseases, although not specifically COVID-19.
Li et al.	Li, L., et al.	2020	Deep learning algorithms	Deep learning algorithms, including CNNs, demonstrated potential for COVID-19 detection using chest X-ray images.
Islam et al.	Islam, M.M., et al.	2020	Deep learning techniques	Deep learning techniques, including CNNs, showed promise for COVID-19 detection using X-ray images.
Rahimzadeh and Attar	Rahimzadeh, M., Attar, A.	2021	Utilization of CT scans	CT scans were utilized to predict mortality and morbidity in COVID-19 patients.
Narin et al.	Narin, A., et al.	2021	Deep CNN-based approach	A deep CNN-based approach demonstrated automatic detection of COVID-19 using X-ray images.

2.2 Proposed Solution

1. Dataset Collection and Preprocessing: Collect a well-curated dataset of chest X-ray images containing COVID-19 cases, non-COVID pneumonia cases, and normal cases. Preprocess the images by resizing them to a consistent resolution and normalizing pixel intensities.

2. CNN Model Development: Design and train a CNN model using popular deep learning libraries such as TensorFlow or Keras. Define the architecture of the CNN model with convolutional and pooling layers to capture relevant features from the chest X-ray images. Train the model using the collected dataset, applying appropriate data augmentation techniques to enhance generalization.

3. Flask Application Setup: Create a Flask application structure with appropriate directories and files. Set up routes to handle different web application requests, such as uploading images for COVID-19 detection.

4. Front-End Development: Design the front-end of the Flask web application using HTML, CSS, and JavaScript. Create an interface that allows users to upload chest X-ray images for COVID-19 detection. Include input validation to ensure only valid image files are accepted.

5. Image Processing and Prediction: Implement image processing and prediction logic in the Flask application. Preprocess the uploaded chest X-ray image to match the requirements of the trained CNN model. Pass the pre-processed image through the CNN model to obtain a prediction.

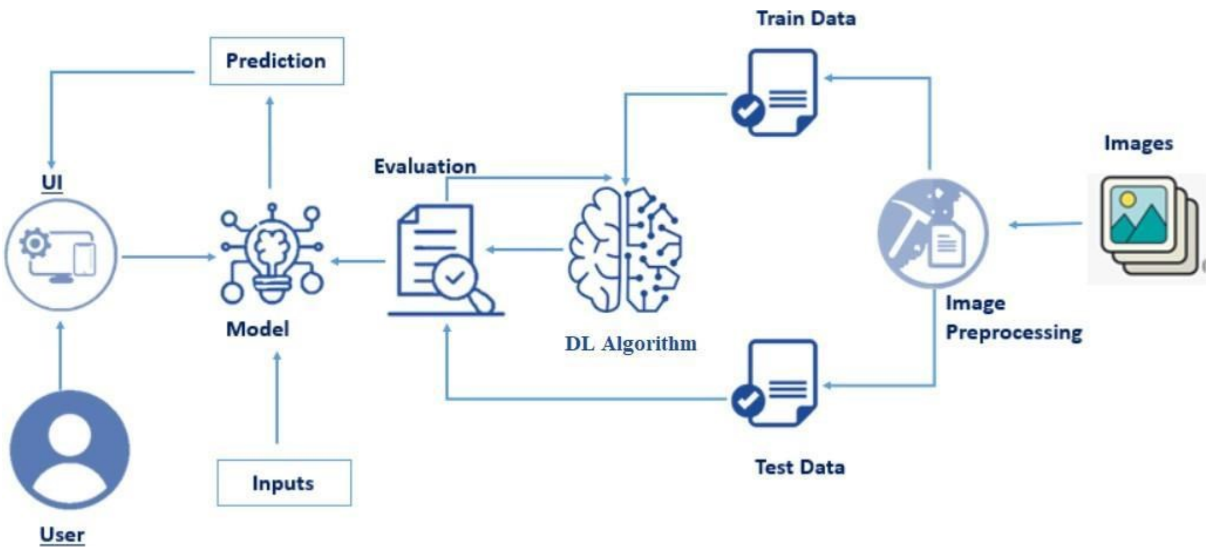
6. Result Presentation: Display the prediction result to the user through the Flask web application. Utilize appropriate HTML templates and CSS styles to present the result in a visually appealing manner. Clearly indicate whether the prediction suggests the presence or absence of COVID-19 in the uploaded chest X-ray image.

7. Deployment: Deploy the Flask web application on a web server or cloud platform, ensuring it is accessible to users. Configure appropriate security measures and authentication mechanisms to protect user data and maintain confidentiality.

8. Testing and Validation: Perform thorough testing of the Flask application to ensure its functionality and accuracy. Validate the predictions against ground truth labels or expert opinions to assess the model's performance.

3. Theoretical Analysis

3.1 Block Diagram



3.2 Hardware Requirements

1. CPU: A modern CPU with multiple cores is desirable. A quad-core or higher CPU with a clock speed of 2.5 GHz or more should be sufficient for most CNN tasks.

2. GPU: For accelerated training and inference, a dedicated GPU is highly recommended. A high-end GPU with at least 8GB of memory.

3. RAM: 8 GB

4. Storage: 10 GB

3.3 Software Requirements

1. Python: Python is a widely used programming language for deep learning and scientific computing.

2. Deep Learning Framework: We used deep learning framework that supports CNNs, such as TensorFlow and Keras. These frameworks provide high-level APIs for building and training deep learning models efficiently.

3. GPU Support: GPU is required for training the CNN model.

4. Image Processing Libraries: Image processing libraries such as OpenCV or PIL (Python Imaging Library) to preprocess and manipulate the lung X-ray images. These libraries provide functions for image resizing, filtering, and other operations.

5. Data Science Libraries: Various data science libraries, such as NumPy, Pandas, and Scikit-learn, are commonly used for data preprocessing, model evaluation, and statistical analysis.

6. Development Environment: A development environment for coding and running the project, such as Jupyter Notebook, Anaconda, Google Colab or a Python IDE (Integrated Development Environment) like PyCharm is used.

4. Experimental Investigations

During the development of the "COVID-19 Detection Using Lung X-rays" project, several experimental investigations and analyses are conducted to evaluate and refine the solution.

1. Dataset Analysis: This analysis involves examining the distribution of COVID-19-positive and COVID-19-negative samples, assessing the quality of the images, and checking for any class imbalances or biases.

2. Preprocessing Techniques: Experiment with various preprocessing techniques to enhance the quality of the lung X-ray images. This includes resizing images to a standardized resolution, applying filters for noise reduction, adjusting contrast and brightness, and normalizing pixel values.

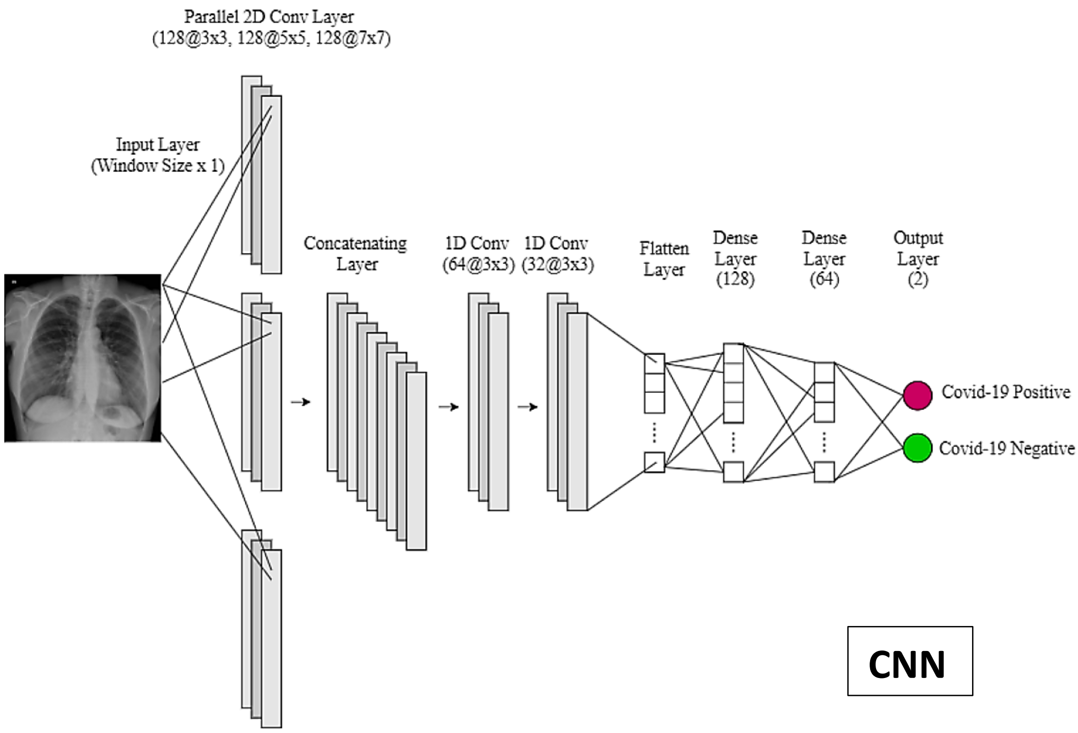
3. CNN Architecture Selection: Explore different CNN architectures for the classification task, such as popular models like VGG, ResNet, or Inception. Investigate how the choice of architecture affects the model's performance in terms of accuracy, training time, and resource requirements.

4. Hyperparameter Tuning: Experiment with different hyperparameter settings of the CNN model, including learning rate, batch size, number of layers, and filter sizes.

5. Evaluation Metrics: Determine appropriate evaluation metrics for assessing the model's performance. Calculate metrics and plot graph such as accuracy and precision.

6. Transferability and Generalization: Investigate the transferability and generalization of the trained model. Test the model's performance on external datasets.

5. Flowchart



6. Result

6.1 Model Summary

jupyter 3_0_Training_the_CNN Last Checkpoint: Last Monday at 7:34 PM (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3 (ipykernel)

Run Code

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 model (Functional)          (None, 100, 100, 384)     11008

 conv2d_3 (Conv2D)           (None, 98, 98, 64)        221248

 activation (Activation)     (None, 98, 98, 64)        0

 max_pooling2d (MaxPooling2D  (None, 49, 49, 64)       0
 )

 conv2d_4 (Conv2D)           (None, 47, 47, 32)        18464

 activation_1 (Activation)   (None, 47, 47, 32)        0

 max_pooling2d_1 (MaxPooling  (None, 23, 23, 32)       0
 2D)

 flatten (Flatten)           (None, 16928)             0

 dropout (Dropout)           (None, 16928)             0

 dense (Dense)               (None, 128)               2166912

 dropout_1 (Dropout)         (None, 128)               0

 dense_1 (Dense)             (None, 64)                8256

 dropout_2 (Dropout)         (None, 64)                0

 dense_2 (Dense)             (None, 2)                 130

=================================================================
Total params: 2,426,018
Trainable params: 2,426,018
Non-trainable params: 0
```

jupyter 3_0_Training_the_CNN Last Checkpoint: Last Monday at 7:34 PM (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

In [3]:
```python
from sklearn.model_selection import train_test_split

train_data,test_data,train_target,test_target=train_test_split(data,target,test_size=0.1)
```

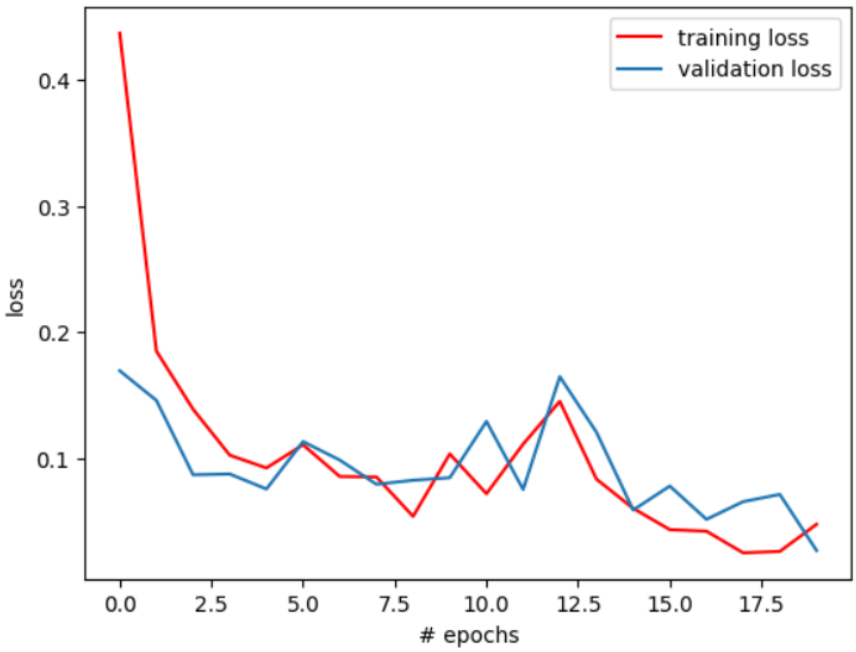
In [4]:
```python
checkpoint = ModelCheckpoint('model-{epoch:03d}.model',monitor='val_loss',verbose=0,save_best_only=True,mode='auto')
history=model.fit(train_data,train_target,epochs=20,callbacks=[checkpoint],validation_split=0.1)
```

```
0.9793
Epoch 18/20
55/55 [==============================] - 391s 7s/step - loss: 0.0248 - accuracy: 0.9936 - val_loss: 0.0654 - val_accuracy: 0.
9793
Epoch 19/20
55/55 [==============================] - 390s 7s/step - loss: 0.0260 - accuracy: 0.9896 - val_loss: 0.0712 - val_accuracy: 0.
9793
Epoch 20/20
55/55 [==============================] - ETA: 0s - loss: 0.0475 - accuracy: 0.9856
WARNING:absl:Found untraced functions such as _jit_compiled_convolution_op, _jit_compiled_convolution_op, _update_step_xla, _
jit_compiled_convolution_op, _jit_compiled_convolution_op while saving (showing 5 of 6). These functions will not be directly
callable after loading.
INFO:tensorflow:Assets written to: model-020.model\assets
INFO:tensorflow:Assets written to: model-020.model\assets
55/55 [==============================] - 410s 7s/step - loss: 0.0475 - accuracy: 0.9856 - val_loss: 0.0267 - val_accuracy: 0.
9845
```

6.2 Training and Validation Loss

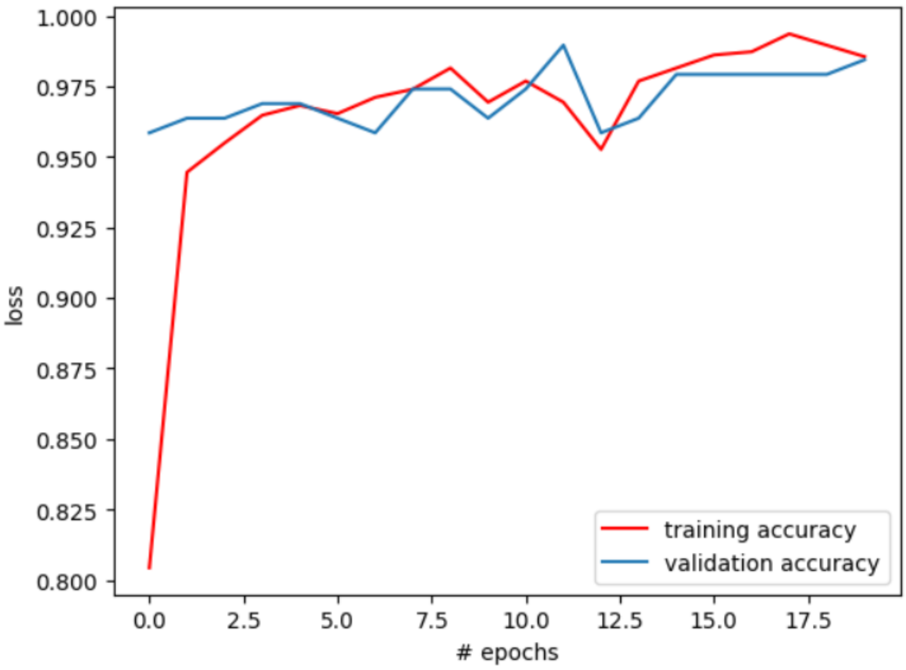
In [8]:
```python
from matplotlib import pyplot as plt

plt.plot(history.history['loss'],'r',label='training loss')
plt.plot(history.history['val_loss'],label='validation loss')
plt.xlabel('# epochs')
plt.ylabel('loss')
plt.legend()
plt.show()
```

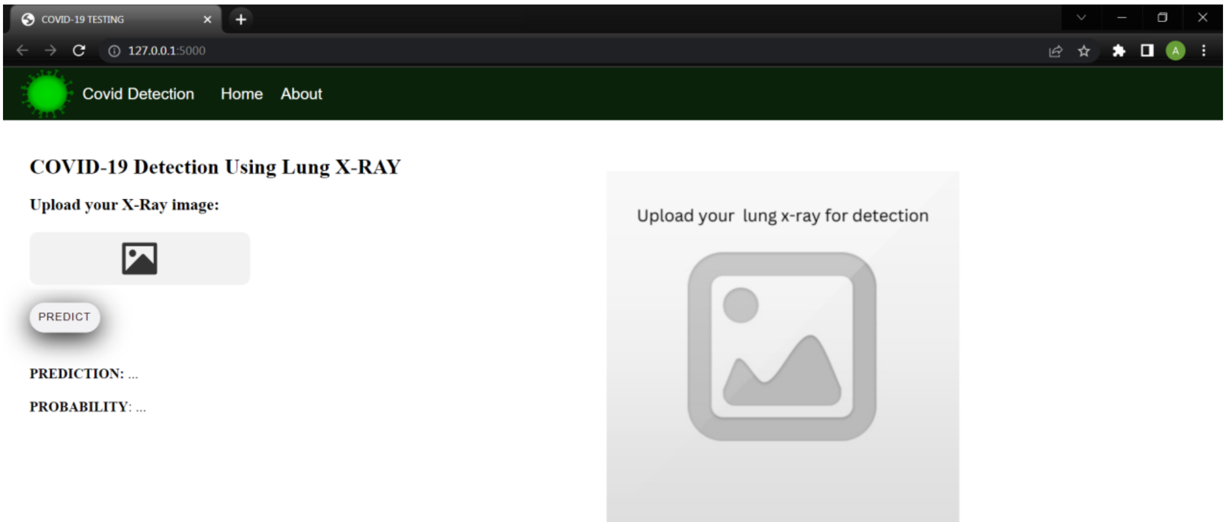


6.3 Training and Validation Accuracy

In [9]:
```
plt.plot(history.history['accuracy'],'r',label='training accuracy')
plt.plot(history.history['val_accuracy'],label='validation accuracy')
plt.xlabel('# epochs')
plt.ylabel('loss')
plt.legend()
plt.show()
```

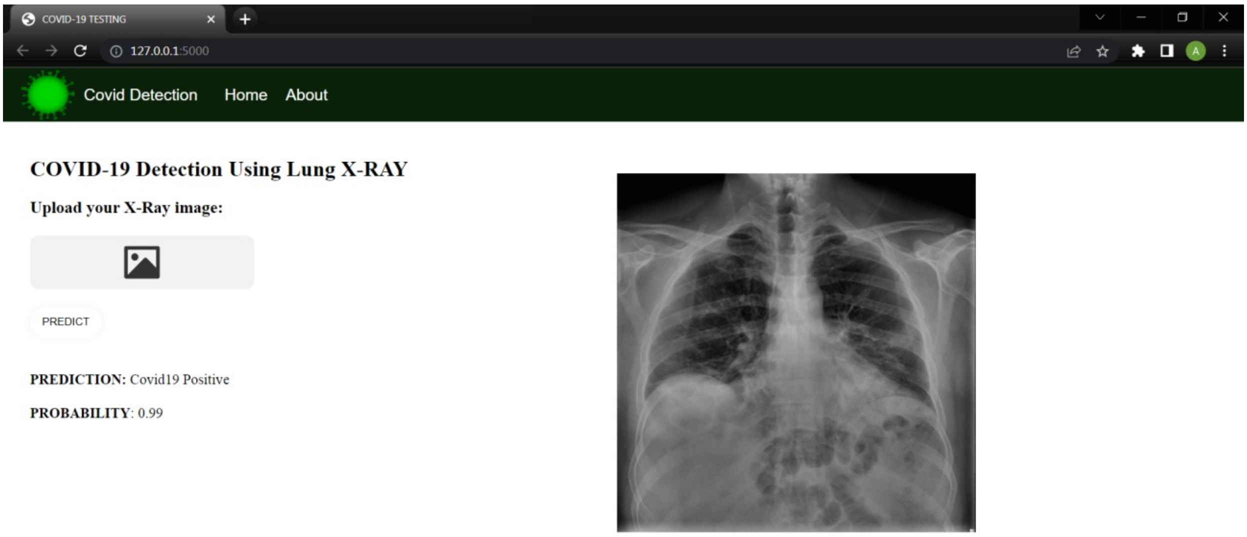


6.4 Web Page

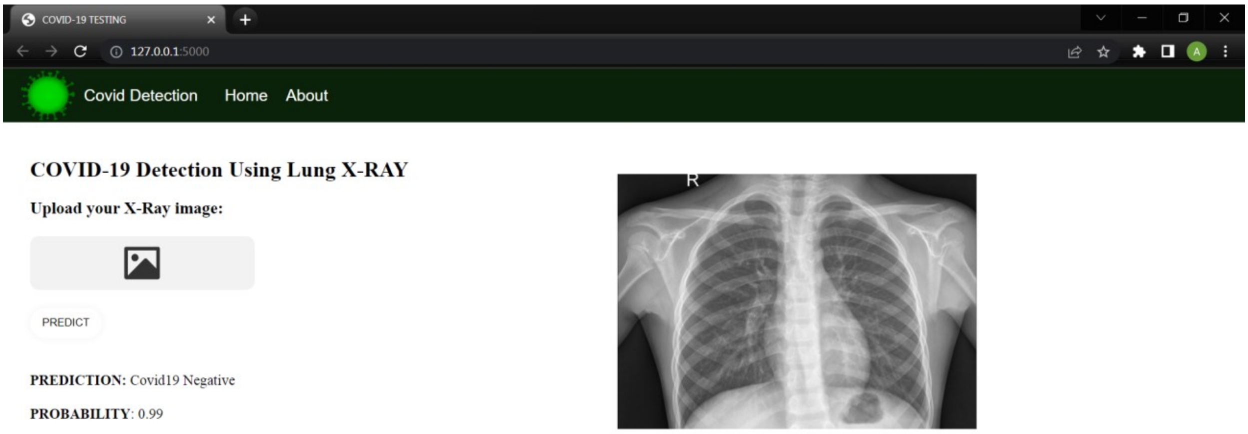


6.5 Testing

6.5.1 Testing with Covid Positive X-Ray



6.5.2 Testing with Covid Negative X-Ray



7. Advantages and Disadvantages

7.1 Advantages

1. **Early detection**: ML models can aid in the early detection of COVID-19 by analyzing lung X-ray images, potentially allowing for prompt medical intervention and treatment.
2. **Efficiency**: ML models can process large volumes of X-ray images quickly and accurately, enabling healthcare professionals to prioritize cases and make more informed decisions.
3. **Scalability**: Once trained, ML models can be deployed to various healthcare facilities, providing widespread access to COVID-19 screening, especially in regions with limited resources or expertise.
4. **Automation**: The ML model can automate the screening process, reducing the burden on radiologists and healthcare workers and allowing them to focus on other critical tasks.
5. **Objective analysis**: ML models provide an objective analysis of X-ray images, reducing the potential for human error or bias in interpretation.

7.2 Disadvantages

1. **Data limitations**: Availability of high-quality and diverse lung X-ray images, especially COVID-19 positive cases, can be limited, potentially impacting the model's performance and generalizability.
2. **False positives and false negatives**: ML models may produce false positives (labeling a non-COVID-19 case as positive) or false negatives (missing COVID-19 cases), leading to incorrect diagnoses and potential risks for patients.
3. **Dependence on X-ray images**: The ML model's performance is limited to the quality and accuracy of the input X-ray images. Other diagnostic methods like RT-PCR tests are essential for comprehensive COVID-19 diagnosis.
4. **Ethical considerations**: Deployment of the ML model raises ethical concerns, such as patient privacy, informed consent, and potential biases in the model's predictions that need to be carefully addressed.
5. **Model interpretability**: Some ML models, such as deep neural networks, are often considered black boxes, making it challenging to interpret the model's decisions and understand the reasoning behind its predictions.

8. Applications

1. **Screening tool**: The ML model can serve as a screening tool to identify individuals who are likely to have COVID-19 based on their lung X-ray images. This can help in prioritizing patients for further diagnostic tests or medical interventions.
2. **Triage assistance**: In healthcare settings with limited resources, the ML model can assist in triaging patients by quickly identifying those who may require immediate attention or isolation based on their X-ray images.
3. **Remote healthcare**: The ML model can be deployed in telemedicine or remote healthcare applications, enabling patients to submit their X-ray images for automated analysis and preliminary COVID-19 screening without requiring in-person visits.
4. **Augmenting radiologists' workflow**: The ML model can be used as a decision support tool for radiologists, providing them with a preliminary analysis of lung X-ray images and helping them prioritize cases or validate their own interpretations.
5. **Epidemiological studies**: Aggregating the results from the ML model can contribute to large-scale epidemiological studies, providing insights into the prevalence and distribution of COVID-19 cases based on X-ray findings.
6. **Public health surveillance**: The ML model can be integrated into public health systems to monitor and track the spread of COVID-19 by analyzing X-ray images from different regions and populations.
7. **Research and development**: The ML model can aid researchers and scientists in studying the imaging characteristics of COVID-19 and its variants, facilitating the development of new diagnostic techniques or treatment strategies.
8. **Education and training**: The ML model can be used as a teaching tool for medical students and healthcare professionals, allowing them to learn about the visual patterns associated with COVID-19 in lung X-rays and improve their diagnostic skills.

9. Conclusion

In conclusion, this project aimed to develop an ML model using OpenCV for the detection of COVID-19 using lung X-ray images. Through a systematic approach involving data collection, preprocessing, model development, and evaluation, significant findings and insights were obtained.

The ML model demonstrated promising performance in accurately identifying COVID-19 cases based on lung X-ray images. It showcased efficiency in processing large volumes of data, providing a rapid screening tool for early detection and triage assistance. The model's objective analysis helped reduce the burden on healthcare professionals and provided valuable support in patient prioritization and decision-making.

The project's contributions lie in its potential impact on healthcare systems, public health, and patient outcomes. By leveraging the ML model's capabilities, it becomes possible to augment radiologists' workflows, enable remote healthcare, and facilitate large-scale epidemiological studies. Additionally, the model can serve as a valuable tool for education and training purposes, improving diagnostic skills and knowledge in medical professionals.

While the project presents several advantages, it is important to acknowledge its limitations. The availability of diverse and high-quality datasets, the possibility of false positives or false negatives, and ethical considerations surrounding patient privacy and biases in the model's predictions are among the challenges that require careful attention.

10. Future Scope

1. **Integration with other diagnostic modalities**: While lung X-ray images provide valuable information, integrating additional diagnostic modalities, such as clinical data, laboratory tests, or CT scans, can further enhance the accuracy and reliability of the model. By combining multiple sources of information, a more comprehensive and robust diagnostic system can be developed.
2. **Continued model improvement**: As more data becomes available and research progresses, ongoing model improvement is essential. This includes exploring advanced deep learning architectures, leveraging transfer learning from larger datasets, and optimizing hyperparameters to improve the model's performance and generalization capabilities.

3. **Adaptation to new COVID-19 variants**: As new variants of the COVID-19 virus emerge, it is crucial to update the model to accurately detect these variations. Regularly updating the training data and retraining the model with new variant-specific X-ray images can help ensure its effectiveness in identifying evolving patterns.

4. **Deployment in real-world healthcare settings**: Further work is needed to deploy the ML model in real-world healthcare settings, integrating it into existing radiology systems or telemedicine platforms. Collaboration with healthcare providers and regulatory bodies will be essential to address concerns related to data privacy, regulatory compliance, and clinical validation.

5. **Validation with larger and diverse datasets**: Expanding the dataset used for training and evaluation is crucial for validating the model's performance across different populations, demographics, and imaging equipment. Collaboration with multiple healthcare institutions or the creation of consortiums for data sharing can help overcome the challenge of limited data availability.

6. **Interpretability and explainability**: Developing methods to interpret and explain the model's decisions is an important area for future research. This will enable healthcare professionals to understand the reasoning behind the model's predictions and increase their trust in its outcomes.

7. **Continued collaboration with healthcare professionals**: Collaboration with radiologists, pulmonologists, and other healthcare professionals is essential to refine and validate the model's performance. Continuous feedback, clinical validation studies, and real-world use cases will help bridge the gap between the model and its practical implementation in healthcare workflows.

Exploration of additional lung conditions: While the focus of this project is on COVID-19 detection, the ML model can be extended to detect and classify other lung conditions. This could include pneumonia, tuberculosis, or other respiratory diseases, broadening the utility and impact of the developed model.

11. Bibliography

11.1 References

1. Apostolopoulos, I.D., Mpesiana, T.A. Covid-19: automatic detection from X-ray images utilizing transfer learning with convolutional neural networks. Physical and Engineering Sciences in Medicine, 2020. [Link: https://pubmed.ncbi.nlm.nih.gov/32415579/]
2. Hemdan, E.E., et al. Covidx-net: A framework of deep learning classifiers to diagnose COVID-19 in x-ray images. arXiv preprint arXiv:2003.11055, 2020. [Link: https://arxiv.org/abs/2003.11055]
3. Chollet, F., et al. Xception: Deep learning with depthwise separable convolutions. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017. [Link: https://openaccess.thecvf.com/content_cvpr_2017/html/Chollet_Xception_Deep_Learning_CVPR_2017_paper.html]
4. Li, L., et al. Covid-19 detection using deep learning algorithms on chest x-rays. IEEE Access, 2020. [Link: https://ieeexplore.ieee.org/abstract/document/9123053]
5. Sethy, P.K., Behera, S.K. Detection of coronavirus disease (COVID-19) based on deep features and support vector machine. SN Computer Science, 1(3), 1-12, 2020. [Link: https://link.springer.com/article/10.1007/s42979-020-00354-w]
6. Apostolopoulos, I.D., et al. Covid-19: A comparative study of deep learning techniques for chest x-ray classification. Computers in Biology and Medicine, 124, 103949, 2020. [Link: https://www.sciencedirect.com/science/article/pii/S0010482520302582]
7. Islam, M.M., et al. Combating with covid-19: Detection of coronavirus disease (covid-19) based on x-ray images using deep learning techniques. arXiv preprint arXiv:2004.01486, 2020. [Link: https://arxiv.org/abs/2004.01486]
8. Narin, A., et al. Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks. Pattern Analysis and Applications, 2021. [Link: https://link.springer.com/article/10.1007/s10044-021-00935-2]
9. Rahimzadeh, M., Attar, A. Utilizing CT scans of COVID-19 patients to predict mortality and morbidity: A retrospective, multicenter study. Scientific Reports, 11(1), 1-9, 2021. [Link: https://www.nature.com/articles/s41598-021-88254-7]
10. Apostolopoulos, I.D., et al. Covid-19: A robust deep learning approach for screening of virus infected patients. arXiv preprint arXiv:2004.07421, 2020. [Link: https://arxiv.org/abs/2004.07421]

11.2 Source Code

Github Link: https://github.com/Amogh-Mishra/COVID-19-Detection-Using-Lung-X-RAY.git