

Covid-19 Classification App

This is a simple web application to detect Covid-19 from Chest X-ray images using a deep learning model.

Features

- Upload an image to classify it as Covid, Normal, or Viral Pneumonia.
- Displays the uploaded image and the predicted class.
- Provides information about the model and its training details.
- Contact information for inquiries and support.

Installation

1. Clone the repository:

```
git clone https://github.com/yourusername/covid_classification_app.git
cd covid_classification_app
```

2. Install the required dependencies:

```
pip install -r requirements.txt
```

3. Download the pre-trained model and place it in the specified path:

```
F:\Route\Sessions\object tracking\covid_19_model.h5
```

Usage

1. Run the Streamlit app:

```
streamlit run covid_app.py
```

2. Open your web browser and go to <http://localhost:8501>.
3. Upload a Chest X-ray image and click the "Predict" button to see the classification result.

Project Structure

- [covid_app.ipynb](#): Jupyter notebook containing the code for the Streamlit app.
- [covid_app.py](#): Python script generated from the Jupyter notebook.
- [test](#): Directory containing sample images for testing.

Model Details

Overview

This Covid-19 Classification model is a deep learning convolutional neural network (CNN) designed to classify chest X-ray images into three categories:

- **Covid:** Indicating a positive Covid-19 case.
- **Normal:** Indicating a healthy individual.
- **Viral Pneumonia:** Indicating pneumonia caused by viruses other than Covid-19.

Training Details

- **Dataset:** The model was trained on a diverse dataset containing thousands of chest X-ray images from various sources.
- **Epochs:** 50
- **Batch Size:** 32
- **Optimizer:** Adam
- **Loss Function:** Categorical Crossentropy
- **Accuracy:** Achieved an accuracy of **95%** on the validation set.

Limitations

- The model's performance is dependent on the quality and diversity of the training data.
- It may not generalize well to images from different sources or with varying image qualities.

Future Improvements

- Incorporate more diverse datasets to improve generalization.
- Implement techniques like transfer learning to enhance performance.
- Develop a more robust preprocessing pipeline to handle various image qualities.

Contact

For any inquiries or support, please reach out to us:

- **Email:** mostafa.abdelsalam14@gmail.com
- **LinkedIn:** [Our LinkedIn](#)
- **GitHub:** [Our GitHub](#)

License

This project is licensed under the MIT License.