

COVID DETECTION PROJECT EXPLANATION

1. Imports and Setup

- The code starts by importing necessary libraries such as `tensorflow`, `cv2`, `numpy`, `os`, `pickle`, and others.
- These libraries are required for tasks like reading images, handling deep learning models, performing predictions, and saving the model.

2. Data Loading and Preprocessing

- The dataset is loaded from a specific directory containing chest X-ray images of COVID, Normal, and Viral Pneumonia cases.
- Each image is read using OpenCV (`cv2.imread()`), resized to 150x150, and normalized by dividing pixel values by 255.0 (for scaling).
- The labels for the images are set as 'COVID', 'Normal', and 'Pneumonia', and these are encoded using `LabelEncoder`.

3. Train-Test-Validation Split

- The dataset is split into training, testing, and validation sets using `train_test_split()`.
- The training set is used to fit the model, the validation set helps monitor performance during training, and the test set is used for final evaluation.

4. Model Architecture

- A CNN model is built with layers such as `Conv2D` (for feature extraction), `MaxPooling2D` (for down-sampling), `Dropout` (for regularization), and `Dense` layers (for classification).
- The model is compiled using the `adam` optimizer, categorical cross-entropy loss, and accuracy as the performance metric.

5. Training the Model

- The model is trained using the `.fit()` function, which takes training data, epochs, batch size, and validation data as inputs.
- The model learns to classify the images over 20 epochs with a batch size of 40.

6. Model Evaluation

- After training, predictions are made on the test set using the `.predict()` function.

- A confusion matrix and classification report are generated to assess the model's performance on test data.
- A heatmap visualizes the confusion matrix using `seaborn`.

7. Saving the Model

- The trained model is saved as `covid-detection-model.h5`, and the label encoder is saved as a pickle file for future use.

8. Real-Time Prediction

- A real-time detection system is implemented where users can provide an image path, and the system will predict if the image shows COVID, Normal, or Pneumonia.
- The image is preprocessed and passed to the trained model to make a prediction, and the confidence score is displayed alongside the image.

9. Batch Prediction Example

- The prediction system can also handle multiple images from the dataset (e.g., every 3rd image in a list of 300), showing predictions for each with a 2-second delay.

This breakdown covers all the major components of the COVID detection project, from loading the dataset to real-time predictions and saving the model for reuse.