

Initial Model Training Code, Model Validation and Evaluation Report

Project Name: COVID-19 Chest X-Ray Image Classification

Data Split

- Training set: 80% of images (from train folder)
- Validation set: 20% of images (subset of train using ImageDataGenerator)
- Testing set: Separate test folder used for unbiased evaluation

Models Trained

CNN approache were explored:

- **Model 1:** Pretrained **VGG16** (transfer learning) with custom dense layers for classification.

Initial Model Training Code :

```
[4]: from keras.applications import VGG16
    from tensorflow.keras.models import Model
    from tensorflow.keras.layers import Dense, Flatten, Dropout, Input

[5]: base_model = VGG16(input_shape = (64, 64, 3), include_top=False , weights="imagenet")

[6]: inp = base_model.input
    x = base_model.output
    x = Flatten()(x)
    x = Dense(1024, activation = 'relu')(x)
    x = Dropout(0.5)(x)

    output = Dense(3, activation = 'softmax')(x)
    model = Model(inputs = inp, outputs = output)

[7]: model.summary()

Model: "functional"
```

```
[8]: model.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metrics = ['accuracy'])

[20]: test_loss, test_acc = model.evaluate(test_set)
41/41 ————— 36s 851ms/step - accuracy: 0.8968 - loss: 0.4168

[10]: history = model.fit(training_set, epochs = 2, validation_data = val_set)
Epoch 1/2
129/129 ————— 611s 5s/step - accuracy: 0.6361 - loss: 1.0127 - val_accuracy: 0.6663 - val_loss: 0.7671
Epoch 2/2
129/129 ————— 597s 5s/step - accuracy: 0.7711 - loss: 0.5901 - val_accuracy: 0.8638 - val_loss: 0.4536
```

Model Validation and Evaluation Report :

Model	Summary	Training and Validation Metrics
Model - A : VGG - 16	<p>Achieved high accuracy by leveraging pretrained ImageNet weights.</p> <p>Fine-tuned dense layers to adapt to chest X-ray dataset.</p> <pre>[8]: model.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metrics = ['accuracy']) [20]: test_loss, test_acc = model.evaluate(test_set) [10]: history = model.fit(training_set, epochs = 2, validation_data = val_set)</pre>	<pre>model.compile(loss = 'categorical_crossentropy', optimizer = 'adam', metrics = ['accuracy']) test_loss, test_acc = model.evaluate(test_set) 41/41 ————— 36s 851ms/step - accuracy: 0.8968 - loss: 0.4168 history = model.fit(training_set, epochs = 2, validation_data = val_set) Epoch 1/2 129/129 ————— 611s 5s/step - accuracy: 0.6361 - loss: 1.0127 - val_accuracy: 0.6663 - val_loss: 0.7671 Epoch 2/2 129/129 ————— 597s 5s/step - accuracy: 0.7711 - loss: 0.5901 - val_accuracy: 0.8638 - val_loss: 0.4536</pre>