Project Development Phase
Model Performance Test

Date	10 November 2022
Team ID	PNT2022TMID34161
Project Name	Project -AI-based localization and classification of skin disease with erythema
Maximum Marks	10 Marks

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1	Model Summary		Model: "sequential" _____ Layer (type) Output Shape Param # ========== keras_layer (KerasLayer) multiple 21802784 _____ dense (Dense) multiple 2049 ========== Total params: 21,804,833 Trainable params: 2,049 Non-trainable params: 21,802,784 _____
2.	Accuracy	**Validation Accuracy-**	Training the Model

```
model_name = f"benign-vs-malignant_{batch_size}_{optimizer}"
tensorboard = tf.keras.callbacks.TensorBoard(log_dir=os.path.join("logs", model_name))
# saves model checkpoint whenever we reach better weights
modelcheckpoint = tf.keras.callbacks.ModelCheckpoint(model_name + "_{val_loss:.3f}.h5", save_best_only=True, verbose=1)

history = m.fit(train_ds, validation_data=valid_ds, steps_per_epoch=n_training_samples // batch_size,

validation_steps=n_validation_samples // batch_size, verbose=1, epochs=100, callbacks=[tensorboard, modelcheckpoint])
```

```
Train for 31 steps, validate for 2 steps
Epoch 1/100
30/31 [==============================>.] - ETA: 9s - loss: 0.4609 - accuracy: 0.7760
Epoch 00001: val_loss improved from inf to 0.49703, saving model to benign-vsmalignant_64_rmsprop_0.497.h5
31/31 [==============================] - 282s 9s/step - loss: 0.4646 - accuracy: 0.7722 - val_loss: 0.4970 - val_accuracy: 0.8125
<..SNIPED..>
Epoch 27/100
30/31 [==============================>.] - ETA: 0s - loss: 0.2982 - accuracy: 0.8708
Epoch 00027: val_loss improved from 0.40253 to 0.38991, saving model to benign-vsmalignant_64_rmsprop_0.390.h5
31/31 [==============================] - 21s 691ms/step - loss: 0.3025 - accuracy: 0.8684 - val_loss: 0.3899 - val_accuracy: 0.8359
<..SNIPED..>
Epoch 41/100
30/31 [==============================>.] - ETA: 0s - loss: 0.2800 - accuracy: 0.8802
Epoch 00041: val_loss did not improve from 0.38991
31/31 [==============================] - 21s 690ms/step - loss: 0.2829 - accuracy: 0.8790 - val_loss: 0.3948 - val_accuracy: 0.8281
Epoch 42/100
```

			30/31 [==========>.] - ETA: 0s - loss: 0.2680 - accuracy: 0.8859 Epoch 00042: val_loss did not improve from 0.38991 31/31 [==========] - 21s 693ms/step - loss: 0.2722 - accuracy: 0.8831 - val_loss: 0.4572 - val_accuracy: 0.8047

		Training Accuracy	valid_ds = valid_ds.map(process_path) train_ds = train_ds.map(process_path) # test_ds = test_ds for image, label in train_ds.take(1): print("Image shape:", image.shape) print("Label:", label.numpy()) Image shape: (299, 299, 3) Label: 0 # training parameters batch_size = 64 optimizer = "rmsprop" def prepare_for_training(ds, cache=True, batch_size=64, shuffle_buffer_size=1 # `prefetch` lets the dataset fetch batches in the background while the model # is training. ds = ds.prefetch(buffer_size=tf.data.experimental.AUTOTUNE) return ds valid_ds = prepare_for_training(valid_ds, batch_size=batch_size, cache="valid-cached-data") train_ds = prepare_for_training(train_ds, batch_size=batch_size, cache="train-cached-data") batch = next(iter(valid_ds)) def show_batch (batch): plt.figure(figsize=(12,12)) for n in range(25): ax = plt.subplot(5,5,n+1) plt.imshow(batch[0][n])

3.	Confidence Score (Only Yolo Projects)	Class Detected - Confidence Score -	None