Project Development Phase
Model Performance Test

Date	9 November 2023
Team ID	Team-592895
Project Name	Lip reading using deep learning
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.	Metrics	**Classification Model:**	
2.	Tune the Model	Hyperparameter Tuning	

[10]:

```
from sklearn.metrics import confusion_matrix, accuracy_score, classification_report

# Assuming you have ground truth values (y_true) and predicted values (y_pred) for classification
y_true = ...
y_pred = ...

# Compute the confusion matrix
conf_matrix = confusion_matrix(y_true, y_pred)

# Compute the accuracy score
accuracy = accuracy_score(y_true, y_pred)

# Generate the classification report
class_report = classification_report(y_true, y_pred)

# Print and display the results
print("Confusion Matrix:")
print(conf_matrix)

print("\nAccuracy Score:", accuracy)

print("\nClassification Report:")
print(class_report)
```

```python
# Assuming you have a function for creating and compiling the model named create_model()

# Hyperparameter tuning
learning_rates = [0.001, 0.0001, 0.00001]
batch_sizes = [16, 32, 64]

best_accuracy = 0
best_hyperparameters = None

for lr in learning_rates:
    for batch_size in batch_sizes:
        model = create_model()  # Make sure to define this function
        model.compile(optimizer=Adam(learning_rate=lr), loss=CTCLoss)  # Assuming Adam optimizer and CTCLoss

        # Train the model on the training set
        model.fit(train, epochs=50, batch_size=batch_size, callbacks=[checkpoint_callback, schedule_callback])

        # Evaluate on the validation set
        val_accuracy = model.evaluate(val)

        # Update best hyperparameters if the current model performs better
        if val_accuracy > best_accuracy:
            best_accuracy = val_accuracy
            best_hyperparameters = {'learning_rate': lr, 'batch_size': batch_size}

print("Best Hyperparameters:", best_hyperparameters)
```