Project-Performance

Model Performance Test

Date	16 November 2023
Team ID	Team-592120
Project Name	Deep Learning Model For Eye Disease Prediction
Maximum Marks	10 Marks

Metrics:

Confusion Matrix:

```
[62]  1 from sklearn.metrics import accuracy_score,f1_score,confusion_matrix
```

```
[74]  1 confusion_matrix(y_test,prediction)

    array([[11,  1,  0],
           [ 2, 15,  0],
           [ 1,  0,  0]])
```

Accuracy-Score:

```
1 accuracy_score(y_test,prediction)

0.8666666666666667
```

Classification-Report:

```
1 print(classification_report(y_test,prediction))
```

```
              precision    recall  f1-score   support

           2       0.79      0.92      0.85        12
           3       0.94      0.88      0.91        17
           4       0.00      0.00      0.00         1

    accuracy                           0.87        30
   macro avg       0.57      0.60      0.59        30
weighted avg       0.85      0.87      0.85        30

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatic
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatic
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classificatic
  _warn_prf(average, modifier, msg_start, len(result))
```

Hyper-Parameter Tuning:

```
1 params={
2     'max_depth':[9,10,11],
3     'min_samples_leaf':[2,3],
4     'n_estimators':[90,95,100,110],
5     'max_features':[2,3,4,5]
6 }
```

Validation-Method:

Cross-Fold Validation with 2 folds

```
[66]  1 from sklearn.model_selection import GridSearchCV
```

```
[67]  1 grid_search=GridSearchCV(estimator=rf,
2                               param_grid=params,
3                               cv=2,
4                               verbose=1,
5                               scoring="accuracy")
```

```
[68]  1 grid_search.fit(x_train,y_train)
```

```
Fitting 2 folds for each of 96 candidates, totalling 192 fits
```

```
            GridSearchCV
▸ estimator: RandomForestClassifier
    ▸ RandomForestClassifier
```

```python
[69]    1 grid_search.best_score_
```

0.8135593220338984

```python
     1 rf_best=grid_search.best_estimator_
     2 rf_best
```

▼	RandomForestClassifier
RandomForestClassifier(max_depth=9, max_features=2, min_samples_leaf=2, n_estimators=90)	

```python
     1 rf_classify=RandomForestClassifier(random_state=42,
     2                                    n_jobs=-1,
     3                                    max_depth=9,
     4                                    min_samples_split=2,
     5                                    max_features='sqrt',
     6                                    n_estimators=90,
     7                                    bootstrap=True)
```

```python
[72]    1 rf_classify.fit(x_train,y_train)
```

▼	RandomForestClassifier
RandomForestClassifier(max_depth=9, n_estimators=90, n_jobs=-1, random_state=42)	