Department of CSE-CYS

20CYS215

Machine Learning in Cyber Security

Assignment Report

<u>Team members</u>

Rajkushal [ch.sc.u4cys23009]

Sai Ruthwik [ch.sc.u4cys23006]

Topic

Exploring Image Feature Extraction Techniques and Analyse their impact on Classification.

Abstract

Feature extraction is a crucial step in computer vision, where raw image data is transformed into meaningful features for classification and detection. This study explores traditional feature extraction techniques such as Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), Gray-Level Co-occurrence Matrix (GLCM), and Oriented FAST and Rotated BRIEF (ORB). Additionally, deep learning-based feature extraction using Convolutional Neural Networks (CNNs), including EfficientNetB0, DenseNet121, MobileNetV2, VGG16, and ResNet50, is examined. The effectiveness of these techniques is evaluated using the CIFAR-10 dataset with multiple classifiers (Random Forest, Logistic Regression, K-Nearest Neighbors, and Support Vector Machine). The results demonstrate that deep learning-based features outperform traditional methods in accuracy and generalization, though at the cost of increased computational requirements.

Table of Contents

1. Introduction

Feature extraction plays a key role in image classification, enabling models to learn representations that improve accuracy. Traditional methods like HOG, LBP, and GLCM rely on manually engineered features, while deep learning models such as ResNet50 and EfficientNetB0 learn hierarchical representations from data. This study evaluates these techniques on the CIFAR-10 dataset and assesses their classification performance using multiple machine learning models.

2. Literature Review

2.1 Importance of Feature Extraction

Feature extraction reduces dimensionality, enhances classification performance, and improves robustness to variations such as lighting and scale.

2.2 Traditional Feature Extraction Methods

- **HOG:** Captures edge and gradient structures for object detection.
- **LBP:** Encodes local texture information.
- **GLCM:** Extracts texture features based on pixel intensity relationships.
- **ORB:** Identifies keypoints and descriptors for object recognition.

2.3 Deep Learning-Based Feature Extraction

- CNN-based models extract deep hierarchical features.
- Networks used: EfficientNetB0, DenseNet121, MobileNetV2, VGG16, ResNet50.

- **Advantage:** Higher accuracy and generalization.
- **Limitation:** Computationally expensive.

3. Methodology

3.1 Dataset and Preprocessing

- **Dataset:** CIFAR-10 (5000 sampled images across 10 classes).
- **Preprocessing:** Image normalization, train-test split (70%-30%).

3.2 Feature Extraction Techniques

- Traditional: HOG, LBP, GLCM, ORB.
- Deep Learning: CNN-based feature extractors with pre-trained models.

3.3 Classification Models

- Random Forest, Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machine (SVM).

4. Experimentation

- Feature extraction performed on preprocessed images.
- **Training:** Models trained on extracted features.
- **Evaluation Metrics:** Accuracy, classification report, and confusion matrix.

5. Results and Discussion

5.1 Performance Metrics

Feature Extraction	Classifier	Accuracy
HOG	Random Forest	35.8%
HOG	Logistic Regression	34.8%

HOG	KNN	33.6%
LBP	Random Forest	22.0%
LBP	Logistic Regression	25.0%
LBP	KNN	20.0%
ResNet50	Random Forest	64.2%
ResNet50	Logistic Regression	69.8%
ResNet50	KNN	54.8%

5.2 Comparative Discussion

- **Traditional methods (HOG, LBP, GLCM, ORB)** provide interpretable features but perform poorly in complex patterns.
- **Deep learning methods (ResNet50, EfficientNetB0, etc.)** outperform traditional techniques but require significant computational power.
- **Trade-offs:** While deep learning excels in accuracy, traditional methods are more efficient for resource-constrained applications.

6. Conclusion and Future Work

6.1 Summary of Findings

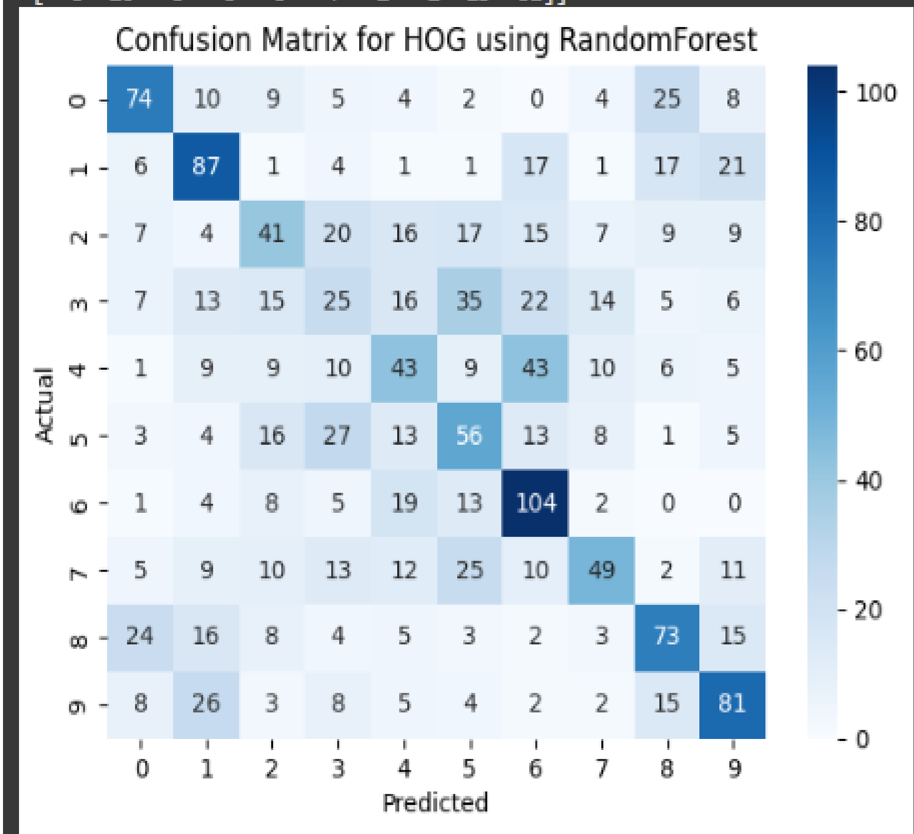
- CNN-based feature extraction significantly improves classification accuracy.
- Traditional methods, though less accurate, are computationally efficient.
- Hybrid approaches combining both methods could offer optimal performance.

6.2 Future Work

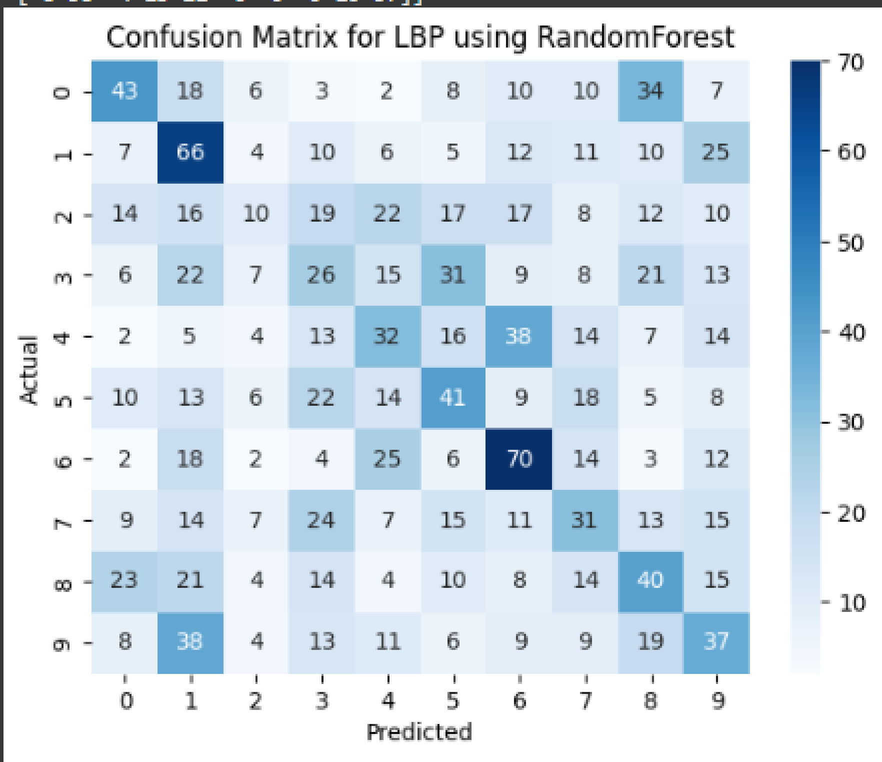
- Exploring feature fusion techniques.
- Applying models to larger datasets.

- Implementing lightweight deep learning architectures for real-time applications.

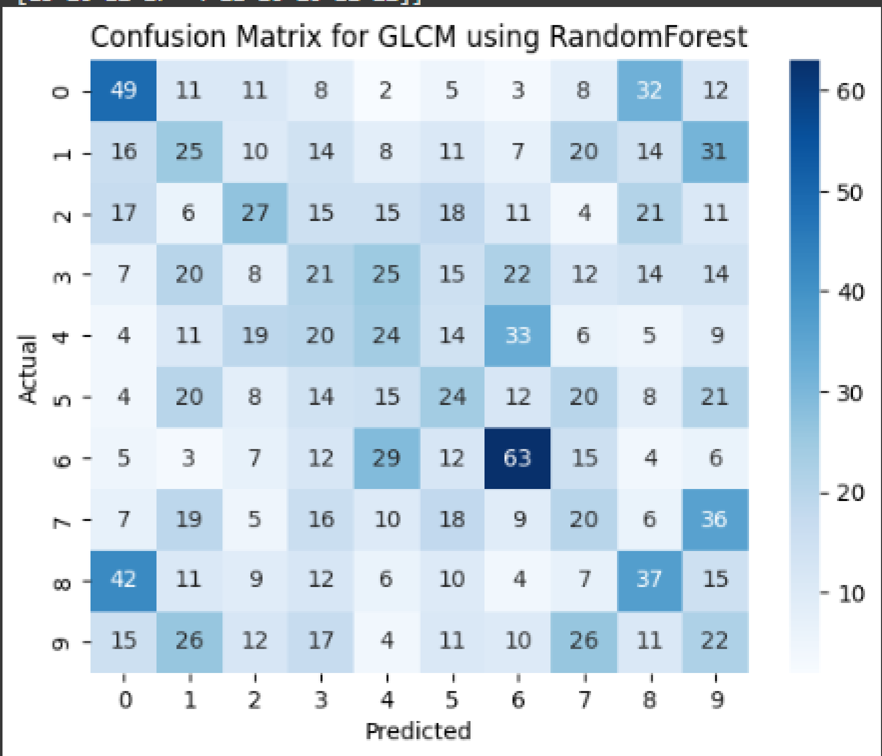
```
Evaluating HOG Features with RandomForest...
              precision    recall  f1-score   support

           0       0.54      0.52      0.53       141
           1       0.48      0.56      0.51       156
           2       0.34      0.28      0.31       145
           3       0.21      0.16      0.18       158
           4       0.32      0.30      0.31       145
           5       0.34      0.38      0.36       146
           6       0.46      0.67      0.54       156
           7       0.49      0.34      0.40       146
           8       0.48      0.48      0.48       153
           9       0.50      0.53      0.51       154

    accuracy                           0.42      1500
   macro avg       0.42      0.42      0.41      1500
weighted avg       0.42      0.42      0.41      1500

Confusion Matrix:
[[ 74  10   9   5   4   2   0   4  25   8]
 [  6  87   1   4   1   1  17   1  17  21]
 [  7   4  41  20  16  17  15   7   9   9]
 [  7  13  15  25  16  35  22  14   5   6]
 [  1   9   9  10  43   9  43  10   6   5]
 [  3   4  16  27  13  56  13   8   1   5]
 [  1   4   8   5  19  13 104   2   0   0]
 [  5   9  10  13  12  25  10  49   2  11]
 [ 24  16   8   4   5   3   2   3  73  15]
 [  8  26   3   8   5   4   2   2  15  81]]
```



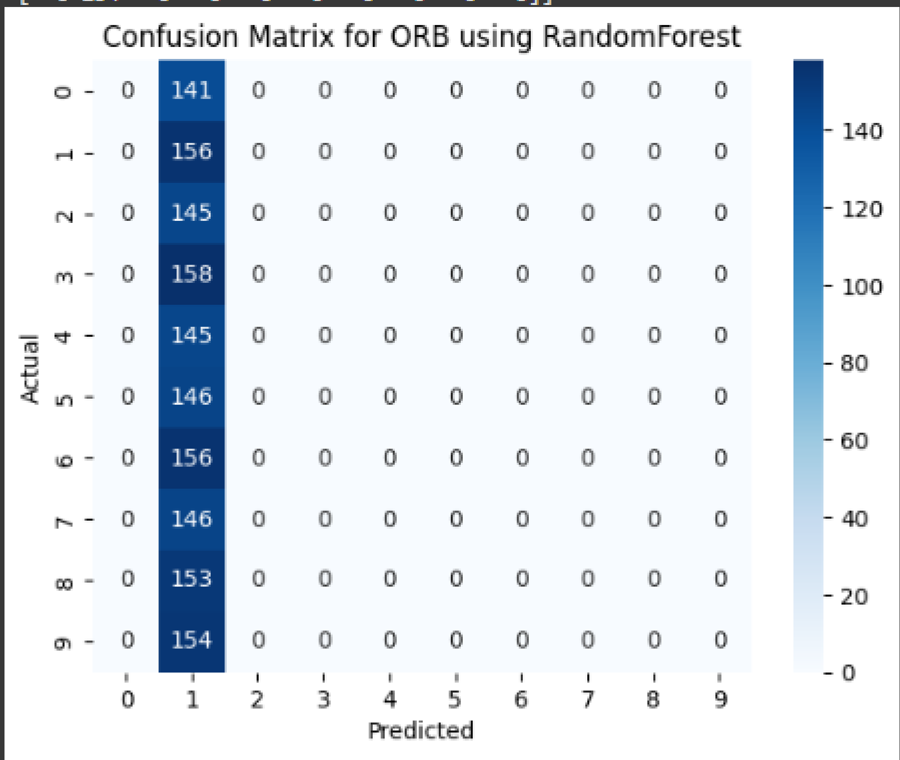
```
Evaluating LBP Features with RandomForest...
              precision    recall  f1-score   support

           0       0.35      0.30      0.32       141
           1       0.29      0.42      0.34       156
           2       0.19      0.07      0.10       145
           3       0.18      0.16      0.17       158
           4       0.23      0.22      0.23       145
           5       0.26      0.28      0.27       146
           6       0.36      0.45      0.40       156
           7       0.23      0.21      0.22       146
           8       0.24      0.26      0.25       153
           9       0.24      0.24      0.24       154

    accuracy                           0.26      1500
   macro avg       0.26      0.26      0.25      1500
weighted avg       0.26      0.26      0.26      1500

Confusion Matrix:
[[43 18  6  3  2  8 10 10 34  7]
 [ 7 66  4 10  6  5 12 11 10 25]
 [14 16 10 19 22 17 17  8 12 10]
 [ 6 22  7 26 15 31  9  8 21 13]
 [ 2  5  4 13 32 16 38 14  7 14]
 [10 13  6 22 14 41  9 18  5  8]
 [ 2 18  2  4 25  6 70 14  3 12]
 [ 9 14  7 24  7 15 11 31 13 15]
 [23 21  4 14  4 10  8 14 40 15]
 [ 8 38  4 13 11  6  9  9 19 37]]
```



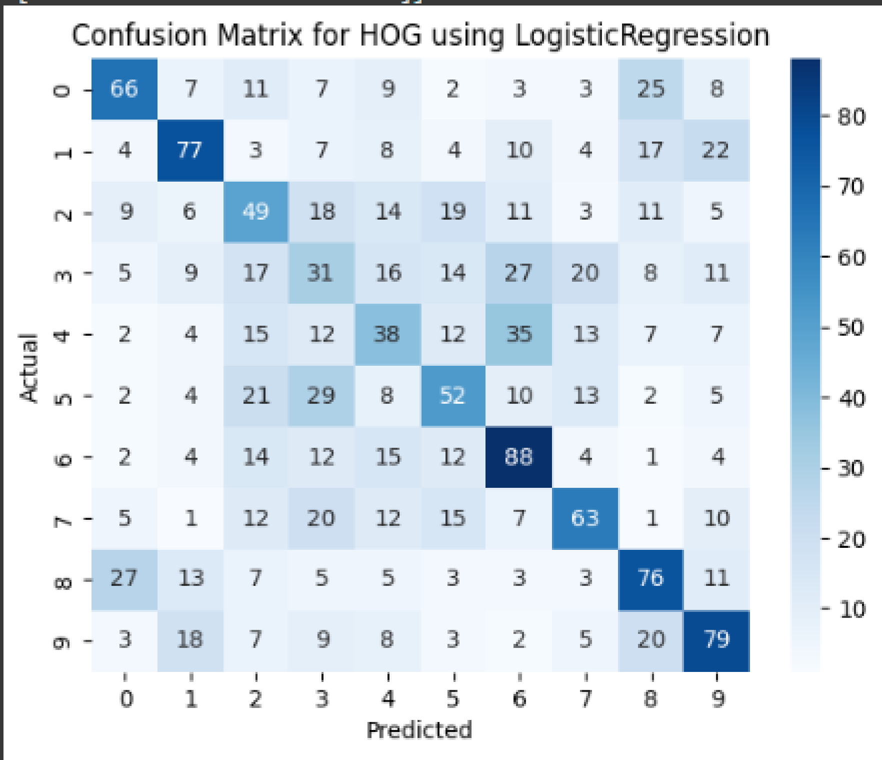
```
Evaluating GLCM Features with RandomForest...
              precision    recall  f1-score   support

           0       0.30      0.35      0.32       141
           1       0.16      0.16      0.16       156
           2       0.23      0.19      0.21       145
           3       0.14      0.13      0.14       158
           4       0.17      0.17      0.17       145
           5       0.17      0.16      0.17       146
           6       0.36      0.40      0.38       156
           7       0.14      0.14      0.14       146
           8       0.24      0.24      0.24       153
           9       0.12      0.14      0.13       154

    accuracy                           0.21      1500
   macro avg       0.21      0.21      0.21      1500
weighted avg       0.21      0.21      0.21      1500

Confusion Matrix:
[[49 11 11  8  2  5  3  8 32 12]
 [16 25 10 14  8 11  7 20 14 31]
 [17  6 27 15 15 18 11  4 21 11]
 [ 7 20  8 21 25 15 22 12 14 14]
 [ 4 11 19 20 24 14 33  6  5  9]
 [ 4 20  8 14 15 24 12 20  8 21]
 [ 5  3  7 12 29 12 63 15  4  6]
 [ 7 19  5 16 10 18  9 20  6 36]
 [42 11  9 12  6 10  4  7 37 15]
 [15 26 12 17  4 11 10 26 11 22]]
```



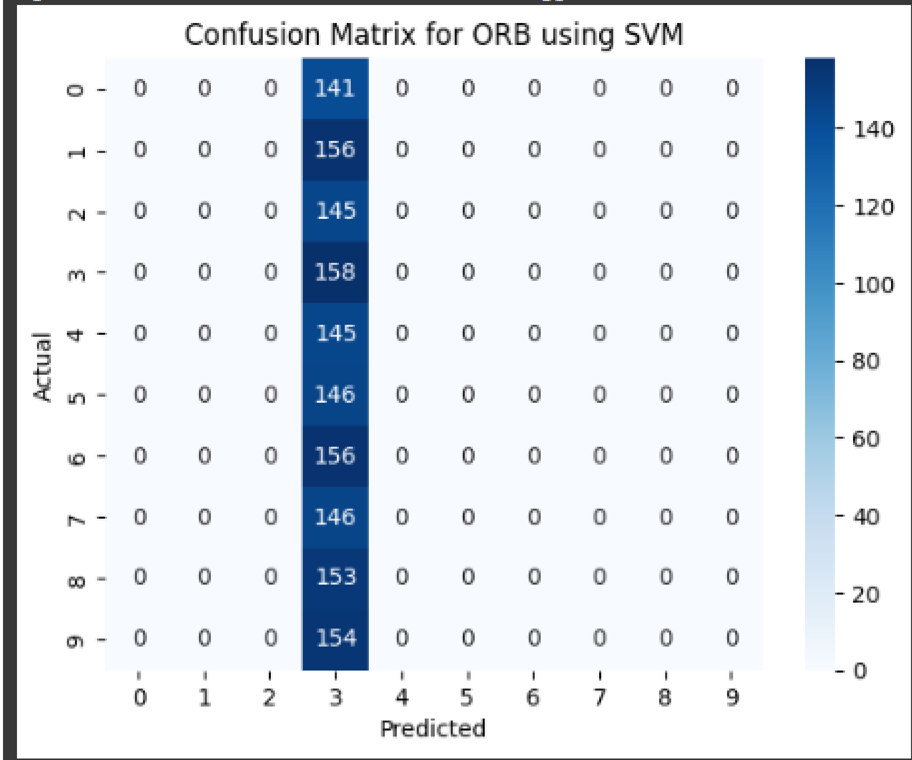
```
Evaluating ORB Features with RandomForest...
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       141
           1       0.10      1.00      0.19       156
           2       0.00      0.00      0.00       145
           3       0.00      0.00      0.00       158
           4       0.00      0.00      0.00       145
           5       0.00      0.00      0.00       146
           6       0.00      0.00      0.00       156
           7       0.00      0.00      0.00       146
           8       0.00      0.00      0.00       153
           9       0.00      0.00      0.00       154

    accuracy                           0.10      1500
   macro avg       0.01      0.10      0.02      1500
weighted avg       0.01      0.10      0.02      1500

Confusion Matrix:
[[  0 141   0   0   0   0   0   0   0   0]
 [  0 156   0   0   0   0   0   0   0   0]
 [  0 145   0   0   0   0   0   0   0   0]
 [  0 158   0   0   0   0   0   0   0   0]
 [  0 145   0   0   0   0   0   0   0   0]
 [  0 146   0   0   0   0   0   0   0   0]
 [  0 156   0   0   0   0   0   0   0   0]
 [  0 146   0   0   0   0   0   0   0   0]
 [  0 153   0   0   0   0   0   0   0   0]
 [  0 154   0   0   0   0   0   0   0   0]]
```



```
Evaluating HOG Features with LogisticRegression...
              precision    recall  f1-score   support

           0       0.53      0.47      0.50       141
           1       0.54      0.49      0.52       156
           2       0.31      0.34      0.33       145
           3       0.21      0.20      0.20       158
           4       0.29      0.26      0.27       145
           5       0.38      0.36      0.37       146
           6       0.45      0.56      0.50       156
           7       0.48      0.43      0.45       146
           8       0.45      0.50      0.47       153
           9       0.49      0.51      0.50       154

    accuracy                           0.41      1500
   macro avg       0.41      0.41      0.41      1500
weighted avg       0.41      0.41      0.41      1500

Confusion Matrix:
[[66  7 11  7  9  2  3  3 25  8]
 [ 4 77  3  7  8  4 10  4 17 22]
 [ 9  6 49 18 14 19 11  3 11  5]
 [ 5  9 17 31 16 14 27 20  8 11]
 [ 2  4 15 12 38 12 35 13  7  7]
 [ 2  4 21 29  8 52 10 13  2  5]
 [ 2  4 14 12 15 12 88  4  1  4]
 [ 5  1 12 20 12 15  7 63  1 10]
 [27 13  7  5  5  3  3  3 76 11]
 [ 3 18  7  9  8  3  2  5 20 79]]
```



```
Confusion Matrix:
[[  0   0   0 141   0   0   0   0   0   0]
 [  0   0   0 156   0   0   0   0   0   0]
 [  0   0   0 145   0   0   0   0   0   0]
 [  0   0   0 158   0   0   0   0   0   0]
 [  0   0   0 145   0   0   0   0   0   0]
 [  0   0   0 146   0   0   0   0   0   0]
 [  0   0   0 156   0   0   0   0   0   0]
 [  0   0   0 146   0   0   0   0   0   0]
 [  0   0   0 153   0   0   0   0   0   0]
 [  0   0   0 154   0   0   0   0   0   0]]
```



Final Results Table (Traditional Features & Classifiers):

	Feature	Classifier	Accuracy
0	HOG	RandomForest	0.422000
1	LBP	RandomForest	0.264000
2	GLCM	RandomForest	0.208000
3	ORB	RandomForest	0.104000
4	HOG	LogisticRegression	0.412667
5	LBP	LogisticRegression	0.246667
6	GLCM	LogisticRegression	0.234667
7	ORB	LogisticRegression	0.105333
8	HOG	KNN	0.397333
9	LBP	KNN	0.190000
10	GLCM	KNN	0.192667
11	ORB	KNN	0.104000
12	HOG	SVM	0.388000
13	LBP	SVM	0.254667
14	GLCM	SVM	0.234000
15	ORB	SVM	0.105333