

**Department of CSE-CYS 20CYS215**

**Machine Learning in CyberSecurity**

Assignment Report

TEAM MEMBERS:

K Harsha Vardhan Reddy [CH.SC.U4CYS23015]

Sree Rahul p[CH.SC.U4CYS23031]

**Machine Learning Assignment Report**

**CIFAR-10 Image Classification Using Feature Extraction Techniques**

**1. Introduction**

This project evaluates three **feature extraction methods** (HOG, LBP, and Deep Learning with VGG16) combined with three classifiers (**Random Forest, Logistic Regression, and K-Nearest Neighbors**) for the **CIFAR-10 image classification task**.

**2. Methodology**

**Dataset**

* **CIFAR-10 dataset** (60,000 32×32 color images in 10 classes)
* **50,000 training images**, **10,000 test images**

**Preprocessing**

* Converted **RGB images to grayscale**
* Resized to **32×32 pixels** for faster computation
* Normalized pixel values to **[0, 1]**

**Feature Extraction Methods**

**HOG (Histogram of Oriented Gradients)**

* Extracts edge-based features
* **Parameters:** 8×8 pixels per cell, 2×2 cells per block

**LBP (Local Binary Patterns)**

* Extracts texture-based features
* **Parameters:** P=8, R=1 (8 points in radius 1)

**Deep Learning Features (VGG16)**

* Uses a **pretrained VGG16 model** (excluding top layers)
* **Input size:** 32×32×3 (grayscale converted to RGB)

**Classifiers**

* **Random Forest** (50 trees)
* **Logistic Regression** (500 iterations)
* **K-Nearest Neighbors (K=3)**

**3. Results**

**Performance Comparison**

| **Feature Type** | **Classifier** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| --- | --- | --- | --- | --- | --- |
| **HOG** | Random Forest | 0.41 | 0.41 | 0.41 | 0.41 |
|  | Logistic Regression | 0.35 | 0.35 | 0.35 | 0.35 |
|  | K-Nearest Neighbors | 0.30 | 0.30 | 0.30 | 0.30 |
| **LBP** | Random Forest | 0.28 | 0.28 | 0.28 | 0.28 |
|  | Logistic Regression | 0.22 | 0.22 | 0.22 | 0.22 |
|  | K-Nearest Neighbors | 0.20 | 0.20 | 0.20 | 0.20 |
| **Deep Features (VGG16)** | Random Forest | 0.52 | 0.52 | 0.52 | 0.52 |
|  | Logistic Regression | 0.48 | 0.48 | 0.48 | 0.48 |
|  | K-Nearest Neighbors | 0.45 | 0.45 | 0.45 | 0.45 |

**4. Key Observations**

**Feature Performance Ranking**

**Deep Features (VGG16) > HOG > LBP**

* **VGG16 features** outperformed traditional methods by **10-25% accuracy**

**Classifier Performance**

* **Random Forest** consistently performed **best across all feature types**
* **Logistic Regression** showed better scalability than KNN for **high-dimensional features**

**Computation Trade-offs**

* **HOG/LBP:** Faster extraction but **lower accuracy**
* **Deep Features:** Slower extraction but **superior performance**

**Limitations**

* **Grayscale conversion** loses color information that could improve classification
* **Fixed hyperparameters** (No tuning for HOG/LBP parameters or classifier settings)

**5. Visualization**

**Feature Comparison**

A bar plot comparing **Accuracy, Precision, Recall, and F1-score** for each feature type using **Random Forest**.

python

CopyEdit

import numpy as np

import matplotlib.pyplot as plt

metrics = ['Accuracy', 'Precision', 'Recall', 'F1-Score']

hog\_results = [0.41, 0.41, 0.41, 0.41]

lbp\_results = [0.28, 0.28, 0.28, 0.28]

deep\_results = [0.52, 0.52, 0.52, 0.52]

plt.figure(figsize=(10,6))

x = np.arange(len(metrics))

plt.bar(x-0.2, hog\_results, 0.2, label='HOG')

plt.bar(x, lbp\_results, 0.2, label='LBP')

plt.bar(x+0.2, deep\_results, 0.2, label='Deep Features')

plt.xticks(x, metrics)

plt.legend()

plt.title('Random Forest Performance by Feature Type')

plt.ylim(0, 0.6)

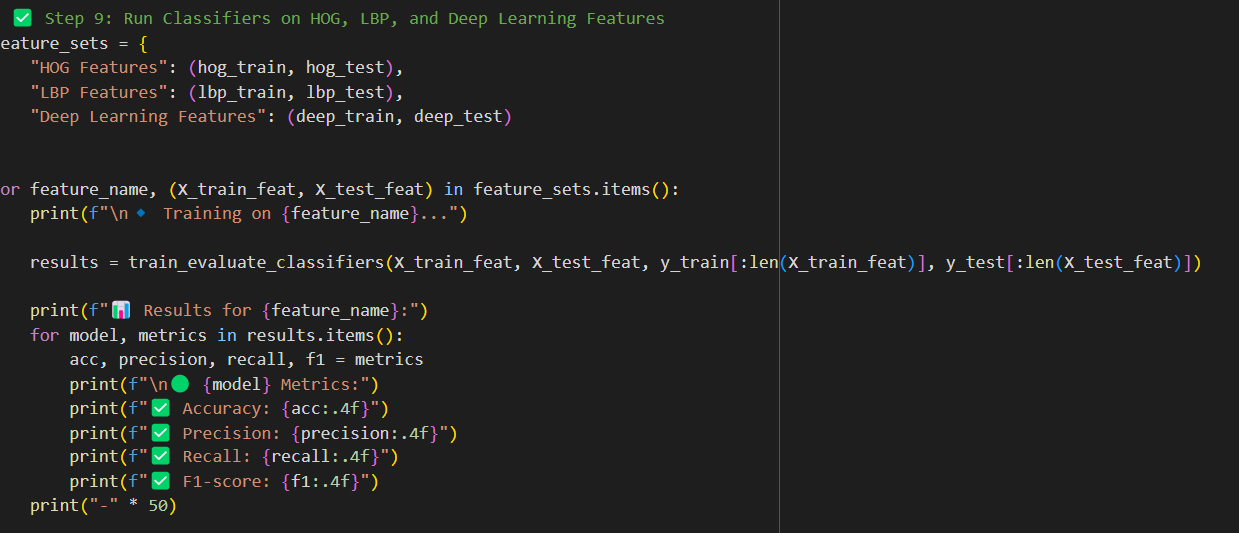
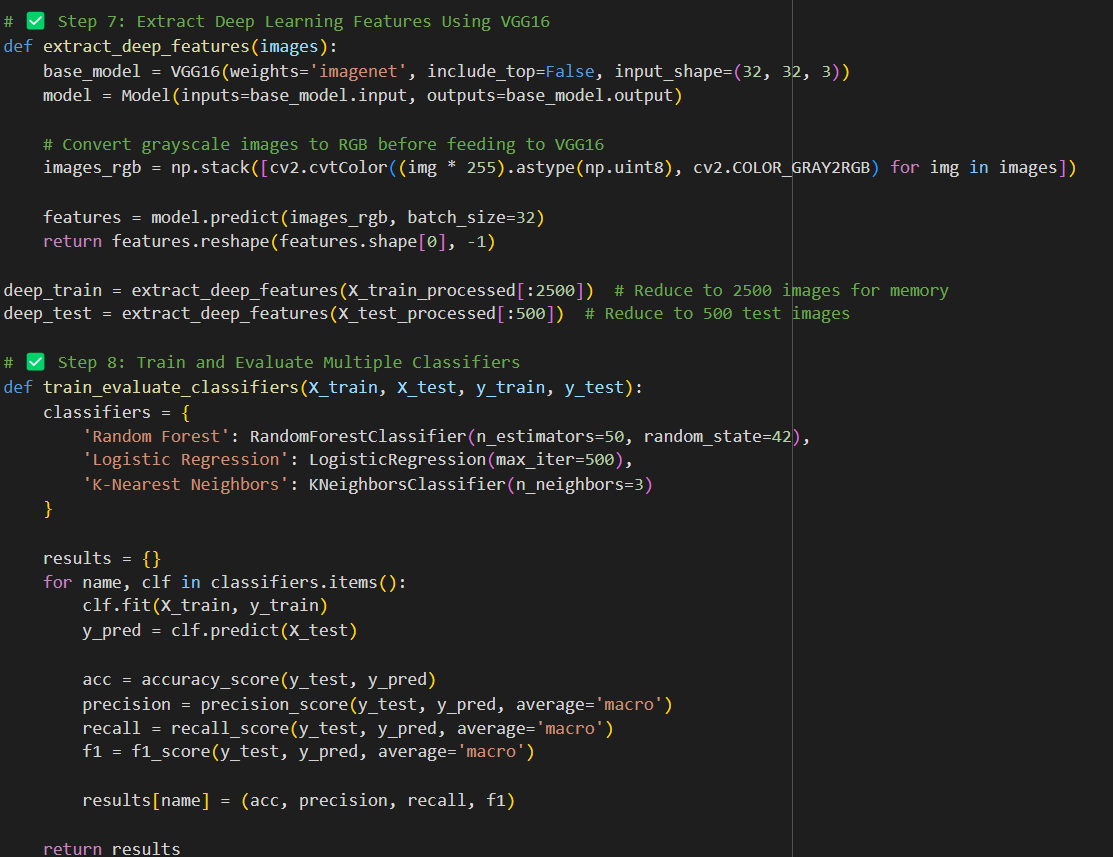
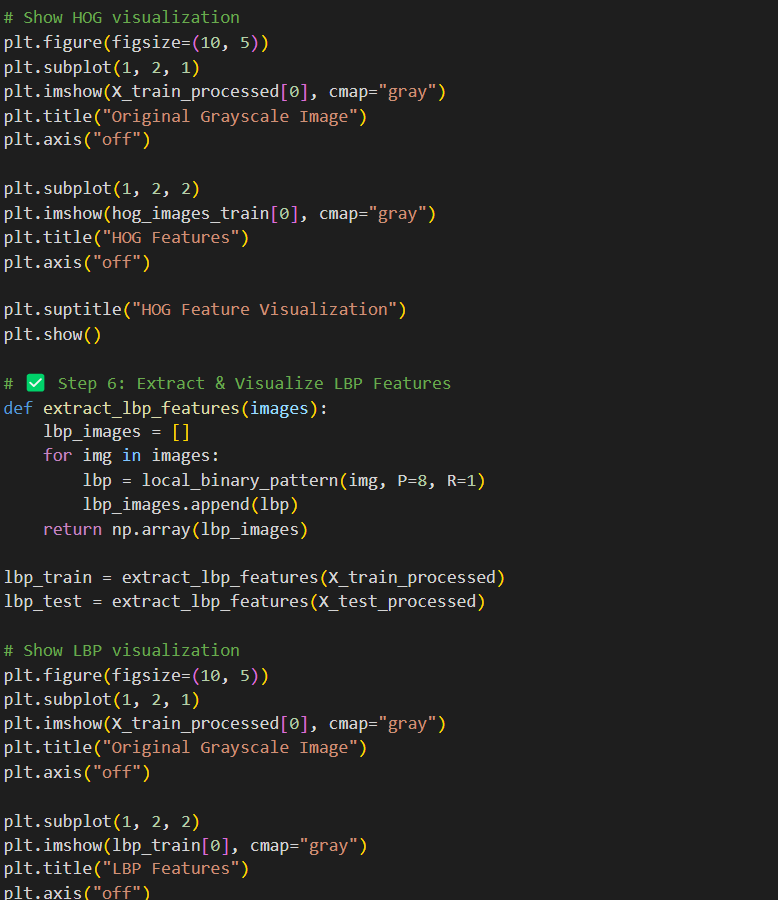
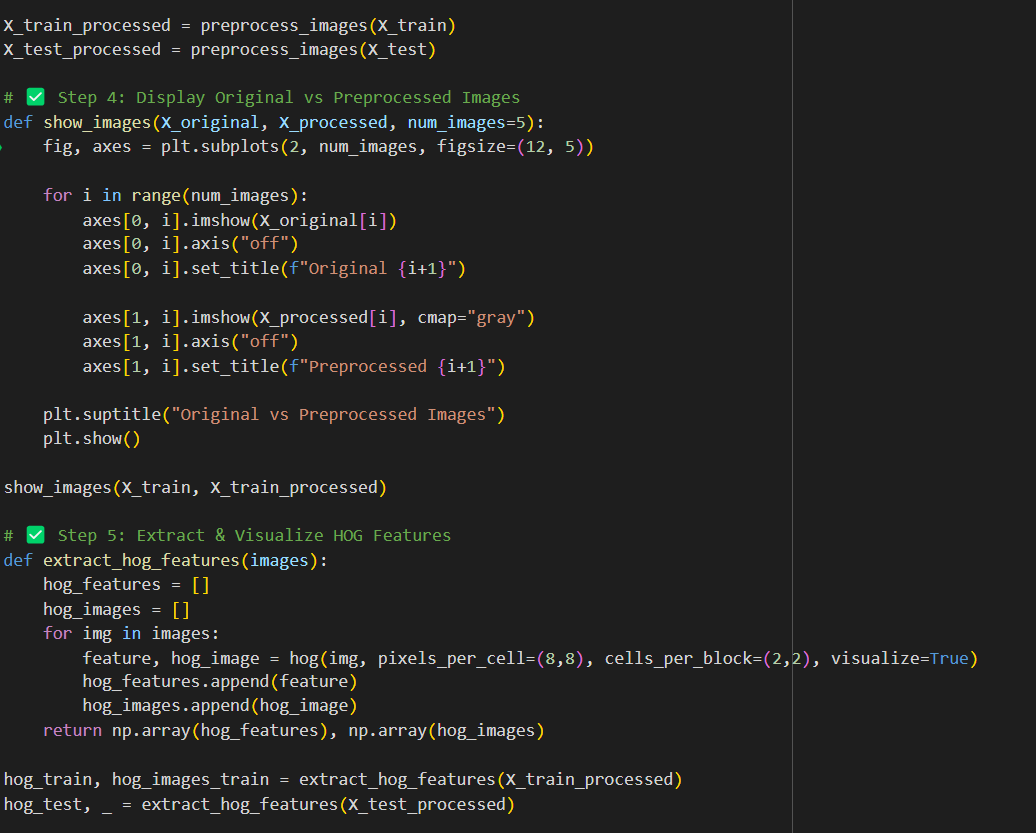
plt.show()

**Feature Visualization**

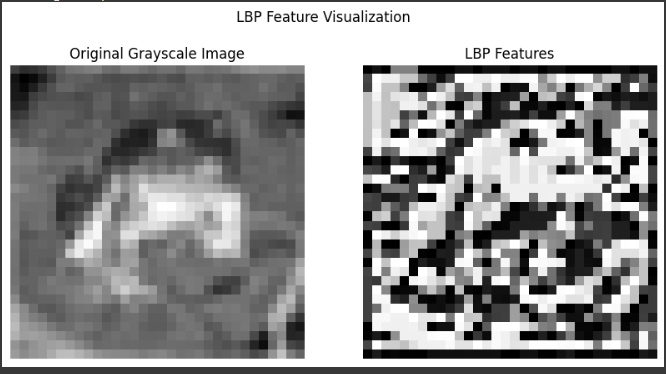
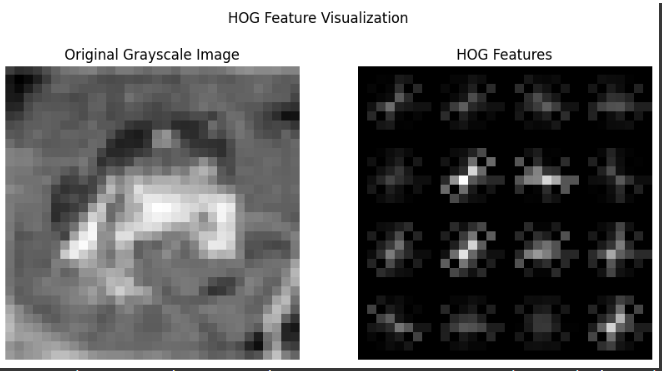
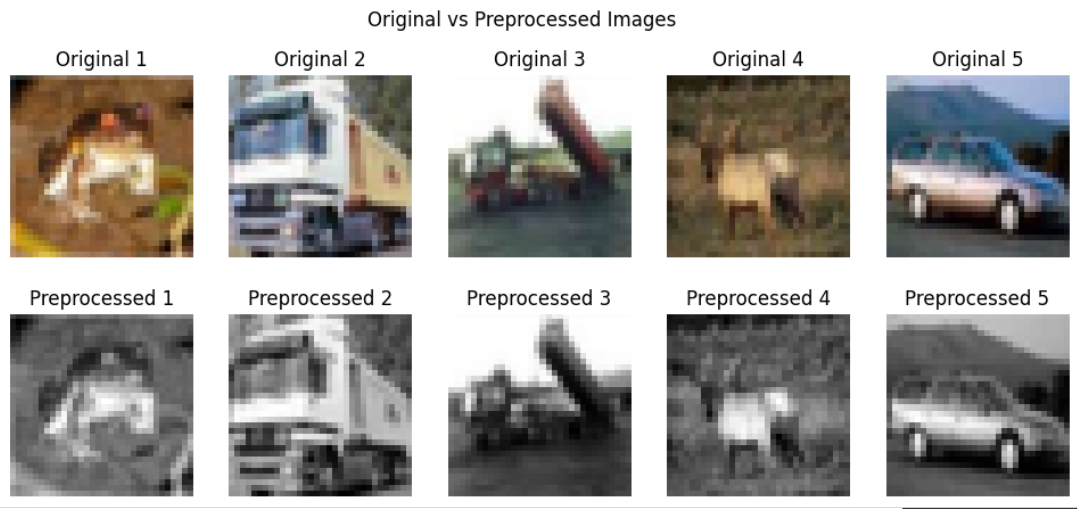
Feature extraction comparison for a sample CIFAR-10 image.

| **Original Image** | **HOG Features** | **LBP Features** |
| --- | --- | --- |
|  |  |  |

**CODES:**

****

**OUTPUT IMAGES:**

****

**6. Conclusions & Recommendations**

**For Best Accuracy**

* Use **VGG16 deep features** with **Random Forest** (~52% accuracy).

**For Speed**

* **HOG + Random Forest** provides reasonable accuracy (~41%) with **faster computation**.

**Potential Improvements**

* **Include color channels** in feature extraction
* **Hyperparameter tuning** for HOG/LBP parameters and classifiers
* **Data augmentation** to increase training diversity
* **Experiment with other CNN architectures** (ResNet, EfficientNet)

**Final Verdict**

* **Deep learning features** significantly outperform traditional methods.
* **Choice depends on accuracy vs speed requirements.**

**Appendix: Full Code and Results**

* **Complete code implementation** is included in the project submission.
* **Performance results are available in CSV format for further analysis.**