**COMPARATIVE ANALYSIS OF TRADITIONAL AND DEEP LEARNING-BASED FEATURE EXTRACTION METHODS FOR FLOWERS CLASSIFICATION**

**ASSIGNMENT REPORT**

***Submitted to***

**Amrita Vishwa Vidyapeetham**

***in partial fulfilment for the award of the degree of***

**BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING**

***By***

**M LIKHITH REDDY**

**CH.SC.U4AIE23033**

**N HARSHITH VARMA**

**CH.SC.U4AIE23035**

**N CHARAN**

**CH.SC.U4AIE23038**

***Supervisor***

**Dr. Deepak K**

**AMRITA VISHWA VIDYAPEETHAM**

**AMRITA SCHOOL OF COMPUTING**

**CHENNAI – 601103**

**October 2025**

****

**BONAFIDE CERTIFICATE**

**Certified that this project report “ COMPARATIVE ANALYSIS OF TRADITIONAL AND DEEP LEARNING-BASED FEATURE EXTRACTION METHODS FOR COIN CLASSIFICATION ” is the Bonafide work of “M LIKHITH REDDY, N HARSHITH VARMA, N CHARAN ” who carried out the project work under my supervision.**

**SIGNATURE**

**Dr. Deepak K**

**SUPERVISOR**

**Associate Professor, Dept. of CSE.**

**Amrita School of Computing**

**Chennai.**

**INTERNAL EXAMINER**

****

**DECLARATION BY THE CANDIDATE**

**I declare that the report entitled “ COMPARATIVE ANALYSIS OF TRADITIONAL AND DEEP LEARNING-BASED FEATURE EXTRACTION METHODS FOR COIN CLASSIFICATION ” submitted by me for the degree of Bachelor of Engineering is the record of the assignment work carried out by me under the guidance of “ Dr. DEEPAK K” and this work has not formed the basis for the award of any degree, diploma, associateship, fellowship, titled in this or any other University or other similar institution of higher learning.**

**M LIKHITH REDDY**

**(CH.SC.U4AIE23033)**

**N HARSHITH VARMA**

**(CH.SC.U4AIE23035)**

**N CHARAN**

**(CH.SC.U4AIE23038)**

**Abstract**

Image classification is a core problem in computer vision that relies heavily on effective feature extraction. In this work, we evaluate and compare multiple feature extraction techniques for flower classification, combining both traditional handcrafted descriptors and deep learning–based features. Specifically, Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), and Edge Histograms are compared against deep features extracted from a pre-trained VGG16 model. To assess classification performance, we employed Logistic Regression, K-Nearest Neighbors (KNN), Decision Trees, and Random Forest classifiers. The models were evaluated using multiple metrics, including accuracy, Cohen’s kappa, ROC-AUC, and confusion matrix analysis. We further analyzed robustness by introducing noise to the test images and measured the computational time required for feature extraction and training. Experimental results show that while handcrafted features provide competitive accuracy for certain classes, VGG16 consistently outperformed them in terms of accuracy, robustness, and generalization. This study highlights the superiority of deep learning–based representations for complex image recognition tasks, while also demonstrating the trade-offs between computational cost and classification performance.

**ACKNOWLEDGEMENT**

**This project work would not have been possible without the contribution of many people. It gives me immense pleasure to express my profound gratitude to our honorable Chancellor Sri Mata Amritanandamayi Devi, for her blessings and for being a source of inspiration. I am indebted to extend my gratitude to our Director, Mr. I B Manikantan Amrita School of Computing and Engineering, for facilitating us all the facilities and extended support to gain valuable education and learning experience.**

**I register my special thanks to Dr. V. Jayakumar, Principal, Amrita School of Computing and Engineering for the support given to me in the successful conduct of this project. I wish to express my sincere gratitude to my supervisor Dr. DEEPAK K, Assistant Professor, Department of Computer Science, for his inspiring guidance, personal involvement and constant encouragement during the entire course of this work.**

**I am grateful to Project Coordinator, Review Panel Members and the entire faculty of the Department of Computer Science & Engineering, for their constructive criticisms and valuable suggestions which have been a rich source to improve the quality of this work.**

**M LIKHITH REDDY, N HARSHITH VARMA, and**

**N CHARAN**

**(Reg. No. CH.SC.U4AIE23033, CH.SC.U4AIE23035**

**& CH.SC.U4AIE23038)**

**TABLE OF CONTENTS**

| **S. No** | **Section Title** | **Page No.** |
| --- | --- | --- |
| 1 | Title Page & Bonafide Certificate | 1 |
| 2 | Declaration by the Candidate | 2 |
| 3 | Acknowledgement | 3 |
| 4 | Abstract | 4 |
| 5 | Introduction | 5–7 |
| 6 | Dataset & Preprocessing | 8–10 |
| 7 | Literature Review | 11–23 |
| 8 | Methodology | 24–30 |
| 9 | Experimental Setup | 31–33 |
| 10 | Experimental Results | 34–44 |
| 11 | Comparative Insights | 45–48 |
| 12 | Conclusion | 49–50 |
| 13 | Future work | 50–51 |
| 14 | References | 51-52 |

**LIST OF TABLES**

| **Table No.** | **Title** | **Section / Content** |
| --- | --- | --- |
| 1 | Summary Table – Literature Review (Zhao et al., 2024) | Literature Review – CNN in Computer Vision |
| 2 | Summary Table – Hybrid CNN–ViT Architectures (Yunusa et al., 2025) | Literature Review – Hybrid CNN–ViT Architectures |
| 3 | Summary Table – CNN Survey (Li et al., 2022) | Literature Review – CNN Survey |
| 4 | Comparative Metrics – Feature Extraction & Classifiers (HOG, LBP, Edge, VGG16) | Comparative Analysis of Feature Extraction Methods |
| 5 | Summary Table – Performance Metrics & Robustness | Experimental Results & Evaluation |
| 6 | Summary Table – Computational Times of Feature Extraction Methods | Experimental Results & Evaluation |

**LIST OF FIGURES**

| **Figure No.** | **Title** | **Section / Content** |
| --- | --- | --- |
| 1 | Confusion Matrix – Logistic Regression (VGG16 Features) | Experimental Results – Classification Analysis |
| 2 | Confusion Matrix – Random Forest (VGG16 Features) | Experimental Results – Classification Analysis |
| 3 | ROC-AUC Curves for Feature Extraction Methods (HOG, LBP, Edge, VGG16) | Experimental Results – ROC-AUC Analysis |
| 4 | Robustness to Noise – Accuracy Comparison Across Methods | Experimental Results – Robustness to Noise |
| 5 | Bar Chart – Comparative Performance of Classifiers & Features | Comparative Insights – Performance Metrics |
| 6 | Heatmap – Class-wise Error Distribution | Experimental Results – Error Analysis |

**LIST OF SYMBOLS AND ABBREVIATIONS**

* **CNN – Convolutional Neural Network**
* **VGG16 – Visual Geometry Group 16-layer deep CNN model**
* **HOG – Histogram of Oriented Gradients**
* **LBP – Local Binary Patterns**
* **GLCM – Gray-Level Co-occurrence Matrix**
* **ROC-AUC – Receiver Operating Characteristic – Area Under Curve**
* **KNN – K-Nearest Neighbors**
* **DT – Decision Tree**
* **RF – Random Forest**
* **SVM – Support Vector Machine**
* **YOLO – You Only Look Once (real-time object detector)**
* **MSER – Maximally Stable Extremal Regions**
* **SIFT – Scale-Invariant Feature Transform**
* **SURF – Speeded-Up Robust Features**
* **ORB – Oriented FAST and Rotated BRIEF**
* **GPU – Graphics Processing Unit**
* **CPU – Central Processing Unit**
* **RAM – Random Access Memory**
* **SSD – Solid-State Drive**

**1. Introduction**

Image classification is a fundamental task in computer vision with applications ranging from medical imaging to autonomous systems. A critical step in achieving high classification performance is the extraction of meaningful features that capture the essential characteristics of an image. Over the years, researchers have proposed a wide variety of feature extraction techniques, which can be broadly categorized into handcrafted and deep learning–based methods.

Handcrafted features, such as Histogram of Oriented Gradients (HOG), Local Binary Patterns (LBP), and edge-based descriptors, rely on explicitly designed mathematical formulations to capture patterns in texture, shape, and structure. These approaches are computationally efficient and interpretable, but their performance often degrades when applied to complex datasets with high intra-class variability.

In contrast, deep learning methods leverage neural networks to automatically learn hierarchical feature representations from raw data. Convolutional Neural Networks (CNNs), in particular, have demonstrated remarkable success in image recognition tasks. Pre-trained models such as VGG16, trained on large-scale datasets like ImageNet, can be used as feature extractors, providing rich and transferable representations even for relatively small datasets.

This study focuses on the classification of flower images using both traditional handcrafted features (HOG, LBP, and edge histograms) and deep learning features (VGG16). A variety of classifiers—Logistic Regression, KNN, Decision Trees, and Random Forests—are employed to analyze the effectiveness of each feature type. The evaluation is conducted using multiple performance metrics, computational time analysis, and robustness testing under noisy conditions. Through this comparative study, we aim to provide insights into the trade-offs between traditional and deep feature extraction approaches, ultimately identifying the method that achieves the best balance of accuracy, robustness, and efficiency.

**2. Dataset**

The dataset used in this study is a flower image collection consisting of five distinct classes: *daisy, rose, sunflower, tulip,* and *dandelion*. Each class is organized into a separate folder, which makes the dataset well-structured and suitable for supervised classification tasks. The dataset contains a few thousand images in total, with a reasonable distribution of samples across classes.

**2.1 Preprocessing**

Before feature extraction, the dataset was preprocessed to ensure uniformity and reduce computational complexity:

* All images were resized to 128 × 128 pixels.
* For handcrafted feature extraction methods (HOG, LBP, and Edge detection), images were converted to grayscale.
* For deep feature extraction (VGG16), images were kept in RGB format and normalized using the preprocess\_input function.
* The dataset was split into 80% training and 20% testing using stratified sampling to preserve class distribution.

**2.2 Justification for Dataset Choice**

The choice of this flower dataset is intentional because it presents a balanced challenge for both handcrafted and deep learning methods:

* Flowers show diverse visual characteristics such as textures (petals), shapes (flower boundaries), and colors, making them an ideal test case for evaluating feature extraction methods.
* The intra-class variation (e.g., different roses under varied lighting, backgrounds, and angles) tests the generalization ability of classifiers.
* The dataset is lightweight enough to be processed on standard hardware without requiring high-end GPUs, while still being complex enough to highlight the advantages of deep features over handcrafted descriptors.
* Since flower recognition has real-world applications in botany, agriculture, and biodiversity monitoring, this dataset ensures that the study is not only academically relevant but also practically meaningful.

Thus, the dataset provides the right balance of size, diversity, and complexity, making it highly suitable for a comparative study of feature extraction methods.

**3.LITRATURE REVIEW**

3.1.Sutikno et al. (2025) proposed combining HAAR, HOG, and LBP descriptors for helmet detection in traffic surveillance. The fused features, classified with SVM/AdaBoost, outperformed single-descriptor methods, with HAAR capturing intensity, HOG contours, and LBP textures. Results showed improved accuracy and robustness under occlusion and lighting changes at low computational cost. However, the dataset was limited, deep learning baselines were not compared, and feature redundancy was unaddressed. Future work suggests dimensionality reduction, hybrid CNN–handcrafted models, and broader multi-class classification.[1]

3.2.The 2025 study compared HOG and LBP with AdaBoost for arachnoid cyst detection in brain MRI. HOG captured structural edges and achieved 0.95 accuracy, outperforming LBP (0.77), which was sensitive to MRI noise. Findings highlight that descriptor choice depends on whether structure or texture cues dominate. Strengths include a clear experimental design, but limitations were small dataset size, lack of CNN baselines, and no optimized LBP variants. Future work suggests hybrid descriptors, transfer learning, noise-robust LBP, and multi-institution validation.[2]

3.3.Sedaghatjoo et al. (2024) optimized Local Binary Pattern (LBP) using Singular Value Decomposition (SVD) to reduce redundancy and improve discriminative power. By reformulating LBP as a matrix factorization problem, SVD retained the most informative local patterns. Experiments on face recognition and expression datasets showed higher accuracy than classical LBP. Strengths include a strong theoretical basis and compact representation, though it was tested only on facial data and not compared with CNNs. Future work suggests extending to medical/industrial domains, hybrid CNN–LBP pipelines, and real-time deployment.[3]

3.4.Adam & Tapamo (2025) surveyed vehicle detection methods, comparing classical descriptors (Haar, HOG, LBP, SIFT, SURF) with deep learning approaches (YOLO, Faster R-CNN, DETR). Classical methods were efficient but struggled with lighting, scale, and occlusion, while deep nets achieved higher accuracy and robustness. Datasets like KITTI, UA-DETRAC, and COCO are widely used with mAP and FPS as benchmarks. Key insights highlight that handcrafted features remain useful for edge deployment, and hybrids with CNNs are promising. Future directions include hybrid benchmarking, annotation-efficient learning, and standardized evaluations.[4]

3.5.Sun et al. (2006) reviewed early vehicle detection methods, highlighting handcrafted features (HOG, Haar) and motion-based techniques, with hybrids offering robustness but limited reliability under occlusion and lighting changes. Later works (Moranduzzo & Melgani 2012; Tang et al. 2017) extended these methods but faced scalability issues. Redmon et al. (2016) revolutionized detection with YOLO, framing it as a single regression task for real-time performance (45 FPS), replacing manual feature design. While early YOLO versions struggled with small/occluded objects, they paved the way for successive improvements and hybrid designs. Future directions emphasize lightweight YOLO models, transformer-based detectors, and deployment on edge devices for intelligent transportation.[5]

3.6.Zhao et al. (2024) present a comprehensive review of CNNs in computer vision, covering core components, architectures (AlexNet to MobileNet), and applications in classification, detection, and video prediction. The paper highlights CNN evolution, challenges like overfitting and efficiency, and the rise of transformer integration. Related works expand on CNN taxonomy, CNN–ViT hybrids, and convolutional variants. Strengths include holistic scope, clarity, and strong academic impact, though details on specific models and transformer analysis are limited. Future directions emphasize CNN–transformer fusion, reinforcement learning hybrids, lightweight architectures, and diversified convolutions.[6]

3.7.Yunusa et al. (2025) provide a structured taxonomy of hybrid CNN–ViT architectures (parallel, serial, fusion, hierarchical, attention-based) and compare them across classification, detection, and segmentation using FLOPs, accuracy, and inference speed. The survey emphasizes performance trade-offs: CNNs excel at local features while ViTs capture global context, making hybrids effective across tasks. Related works (Yunusa et al. 2024; radiology-focused reviews) reinforce the benefits of CNN–ViT integration. Strengths include systematic organization, broad applicability, and practical guidance, while limitations lie in lack of empirical benchmarks and rapidly evolving models. Future research should target efficient lightweight hybrids, adaptive fusion, NAS, domain-specific applications, and improved interpretability.[7]

3.8.Li et al. (2022) deliver a holistic CNN survey, tracing its evolution, convolution types (1D, 2D, multi-dimensional), classic and advanced architectures, empirical implementation rules, applications, and open challenges. The review emphasizes CNN versatility across vision, NLP, and signal processing, while offering practical heuristics for activation, normalization, and hyperparameters. Related works (Gu et al. 2018; Khan et al. 2019) highlight structural innovations and specialized application surveys. Strengths include breadth, multidimensional inclusivity, and strong academic credibility, while limitations are lack of benchmarks, possible datedness, and limited task-specific depth. Future research points to hybrid CNNs, broader applications, automated tuning, and validation of heuristics.[8]

3.9.Mikolajczyk & Schmid (2005) benchmarked local descriptors (SIFT, PCA-SIFT, GLOH, Shape Context, etc.) under scale, rotation, illumination, and blur variations using real-world datasets. SIFT and GLOH proved most robust, while PCA-SIFT reduced dimensionality but lost stability, and moment/filter-based methods struggled with viewpoint and lighting. Key related works include Lowe (2004) on SIFT, Ke & Sukthankar (2004) on PCA-SIFT, and later SURF (2006) and ORB (2011). Strengths lie in its large-scale, standardized evaluation and proof of SIFT’s robustness, while limitations include focus on handcrafted descriptors and controlled datasets. Future directions point to compact/hybrid descriptors, large-scale benchmarks, and learning-based approaches.[9]

3.10.The paper “Local Invariant Feature Detectors: A Survey”by Tuytelaars and Mikolajczyk provides a comprehensive review of local feature detectors, outlining the essential properties of an ideal detector such as repeatability, distinctiveness, and computational efficiency. It categorizes existing methods—including Harris, Hessian, DoG, and MSER—while evaluating their robustness against scale, rotation, illumination, and affine distortions. The survey highlights trade-offs between computational cost and invariance, showing how efficiency often comes at the expense of robustness. A detailed comparative analysis identifies which detectors are most suitable for tasks like wide-baseline image matching. Finally, the authors discuss open research challenges, emphasizing the need for improved invariance under large viewpoint changes, hybrid detector strategies, and more efficient algorithms.[10]

3.11.Pietikäinen & Zhao (2016) survey two decades of LBP, covering its original algorithm, key variants (rotation-invariant, CLBP, LPQ), and extensions into spatiotemporal (LBP-TOP), color, and higher-dimensional domains. LBP remains valued for efficiency and robustness to illumination, with co-occurrence models (e.g., PRI-CoLBP) boosting discriminative power. Applications span face recognition, dynamic textures, and medical imaging. Strengths include broad historical coverage, efficiency focus, and clear future research directions, while limitations involve lack of benchmarking, single-family scope, and no deep learning integration. Future work points to higher-dimensional, hybrid, adaptive, and deep-integrated LBP applications.[11]

3.12.Haralick, Shanmugam & Dinstein (1973) introduced the Gray-Level Co-occurrence Matrix (GLCM) with 14 statistical features (contrast, correlation, energy, entropy, etc.) for texture analysis. These features capture second-order spatial relationships, improving classification over first-order statistics and proving useful across medical, industrial, and remote sensing domains. The work is foundational, mathematically rigorous, and highly cited, though limited by computational cost, parameter sensitivity, redundancy, and lack of rotation/scale invariance. Later studies (e.g., LBP, wavelets, sea ice analysis) extended and applied GLCM widely. Future work explores dimensionality reduction, multi-scale/3D extensions, and integration with machine learning and deep learning.[12]

3.13.Manjunath & Ma (1996) proposed Gabor wavelet features for texture analysis in content-based image retrieval, showing superior performance over wavelet and autoregressive models on the Brodatz database. Gabor filters capture frequency and orientation, inspired by human visual perception, making them powerful and interpretable. Their method proved effective in real-world tasks like aerial photo browsing. However, challenges include high computational cost, complex parameter tuning, and limited handling of dynamic textures. Future directions involve efficient approximations, scale/rotation invariance, hybrid descriptors, and integration with machine learning or deep learning.[13]

3.14.Liu et al. (2016) presented a comprehensive taxonomy and experimental study of over 40 local binary features (LBP variants and related descriptors) for texture classification. They categorized descriptors into micro-structure, co-occurrence, and joint encoding families, clarifying their evolution. Rigorous benchmarking identified top-performing variants with robustness to noise, illumination, and efficiency. Strengths include wide coverage, clear taxonomy, and practical evaluation, but limitations involve exclusion of deep learning, limited datasets, and no runtime analysis. Future directions suggest integration with CNNs, hybrid fusion, automated feature selection, and large-scale evaluations.[14]

3.15.Liu, Xu & Wang (2021) reviewed traditional and deep learning-based keypoint detection and description methods for image registration. They compared handcrafted techniques (SIFT, SURF, ORB, BRISK, etc.) with deep models (LIFT, SuperPoint, TILDE, Siamese/Triplet networks) through experiments on benchmark datasets. The study provides practical guidance on choosing detector–descriptor pairs for varied conditions like illumination, noise, and geometric distortions. Strengths include comprehensive coverage, experimental validation, and application-focused insights, while limitations involve narrow scope, dataset constraints, and lack of runtime analysis. Future work suggests broader benchmarking, lightweight DL models, hybrid methods, and end-to-end registration frameworks.[15]

3.16.Baaziz, Abahmane & Missaoui (2010) reviewed spatial-frequency texture feature extraction methods for CBIR, focusing on DWT, Gabor, DT-CWT, and Contourlet transforms. They discussed energy-based metrics for efficiency and statistical modeling (SM–KL) for rigorous similarity measurement. Key insights highlight the role of multiscale, orientation-selective transforms in capturing spatial and frequency texture properties. Strengths include comprehensive coverage and guidance on trade-offs, while limitations involve lack of empirical evaluation and absence of deep-learning comparisons. Future work suggests standardized benchmarking, efficient invariant descriptors, hybrid frameworks, and integration with deep learning.[16]

3.17.Kurnianggoro et al. (2018) present a survey of 2D shape representation methods, covering contour-based, region-based, graph/topology-based, and hybrid or deep learning approaches, and evaluate them in terms of invariance, robustness, computational complexity, and matching performance. Contour methods excel at fine boundary details but are noise-sensitive, while region-based approaches offer robustness at the cost of detail; compact moment descriptors like Hu and Zernike provide invariance. The survey also reviews classical works, skeleton and graph-based methods, and emerging deep learning descriptors. It concludes by emphasizing hybrid models, learning-based approaches, real-world applications, and the need for improved robustness and invariance as key future directions.[17]

3.18.Beddiar et al. (2020) present a comprehensive survey of vision-based human activity recognition (HAR), classifying approaches by feature types (handcrafted vs. deep), recognition pipeline stages, data modalities, and supervision levels, while also reviewing applications in areas such as HCI, surveillance, VR, and healthcare alongside datasets, metrics, and challenges. The survey highlights the integration of traditional and deep learning methods, a structured taxonomy, and attention to real-world issues like occlusion and privacy. Its strengths include timeliness and multidimensional analysis, though it lacks discussion of runtime efficiency, empirical benchmarking, and coverage of emerging architectures. Future directions emphasize lightweight edge deployment, hybrid feature use, self-supervised learning, and benchmark standardization.[18]

3.19.Jiang & Zhou (2024) present a systematic survey of shape representation methods for object recognition, classifying them into global and structural contour-based as well as global and structural region-based descriptors. The study reviews their principles, invariance, computational aspects, and matching strategies, with applications in retrieval, classification, and medical imaging. It highlights that contour methods are efficient but overlook interior details, region descriptors provide strong invariance but miss fine boundaries, and structural methods capture topology yet suffer from noise sensitivity and higher costs. Emphasizing the trade-offs between discriminative power, robustness, and efficiency, the paper suggests descriptor fusion as a way forward and points to future research in hybrid frameworks, deep learning–based approaches, broader benchmarking, and real-time robust applications.[19]

**4. Methodology**

The methodology followed in this work consists of a structured pipeline comprising data preprocessing, feature extraction, model training, and evaluation. Each step was carefully designed to ensure fair comparison between traditional handcrafted features and deep learning–based features.

**4.1 Data Preprocessing**

* Dataset Structure: The dataset contains images of flowers, organized into separate folders for each class.
* Resizing: All images were resized to 128 × 128 pixels for uniformity.
* Grayscale Conversion: For handcrafted feature extraction methods (HOG, LBP, Edge Detection), images were converted into grayscale to reduce computational cost while preserving texture and shape details.
* Normalization**:** Pixel values and feature histograms were normalized to ensure consistency across images and improve classifier performance.

**4.2 Feature Extraction Methods**

**a) Histogram of Oriented Gradients (HOG)**

The HOG descriptor was applied to grayscale images to capture edge and shape information by computing the distribution of gradient orientations. The resulting feature vectors provide a structural representation of the flowers, emphasizing contour and edge patterns.

**b) Local Binary Patterns (LBP)**

The LBP operator was used to extract texture information by thresholding pixel neighborhoods and converting them into binary patterns. These binary codes were summarized into histograms, producing discriminative texture-based features useful for distinguishing flowers with subtle surface variations.

**c) Edge Detection**

Edges were extracted using the Canny edge detector, which identifies areas of sharp intensity change. Histograms of edge intensities were computed to represent the structural outline of flowers. This method captures boundary-based information that complements texture descriptors.

**d) VGG16 Deep Features**

Deep representations were obtained from a pre-trained VGG16 network (trained on ImageNet) with the classification head removed (include\_top=False). Each image was preprocessed and passed through the convolutional layers, and the output feature maps were flattened into vectors. These deep features encode hierarchical and abstract patterns beyond what handcrafted methods can capture.

**4.3 Classification Models**

For each feature extraction method, multiple **machine learning classifiers** were trained and evaluated:

* **Logistic Regression (LR):** A linear model effective for high-dimensional data.
* **K-Nearest Neighbors (KNN):** A distance-based classifier that labels samples based on nearest neighbors.
* **Decision Tree (DT):** A tree-based classifier that partitions data hierarchically.
* **Random Forest (RF):** An ensemble of decision trees that enhances robustness and reduces overfitting.

**4.4 Evaluation Metrics**

The performance of each feature extraction method–classifier combination was evaluated using multiple metrics to ensure comprehensive assessment:

* **Accuracy:** Overall classification performance.
* **Precision, Recall, and F1-Score:** Class-level metrics from the classification report.
* **Cohen’s Kappa:** Measures agreement beyond chance.
* **ROC-AUC (multi-class, one-vs-rest):** Evaluates class separability for probabilistic classifiers.
* **Confusion Matrix Analysis:** Visualized using heatmaps to analyze misclassification patterns.
* **Class-wise Error Analysis:** Misclassification counts per class, highlighting which flower categories are more prone to confusion.

**5. Experimental Setup**

All experiments were conducted on a personal computer equipped with modern hardware capable of handling computationally intensive deep learning tasks. The specifications of the system are as follows:

* **Processor (CPU):** Intel Core i5 (13th Generation)
* **Graphics Processing Unit (GPU):** NVIDIA GeForce RTX 4050 (6GB VRAM)
* **Memory (RAM):** 16 GB DDR4
* **Storage:** 512 GB SSD for faster data access and reduced I/O latency

The GPU acceleration provided by the NVIDIA RTX 4050 significantly reduced the time required for feature extraction using deep learning models (such as VGG16) compared to CPU-only execution. Classical feature extraction methods (HOG, LBP, and Edge Detection) primarily utilized the CPU, which was sufficient for efficient computation.

**5.1 Software Environment**

The software stack used for the experiments consisted of the following components:

* **Operating System:** Windows 11 (64-bit)
* **Programming Language:** Python 3.10
* **Deep Learning Framework:** TensorFlow / Keras (for VGG16 feature extraction)
* **Machine Learning Libraries:** scikit-learn (for classifiers and evaluation metrics)
* **Image Processing:** OpenCV and scikit-image (for HOG, LBP, and edge detection)
* **Visualization:** Matplotlib and Seaborn (for confusion matrices and performance plots)
* **Development Environment:** Jupyter Notebook / VS Code

**5.2 Implementation Details**

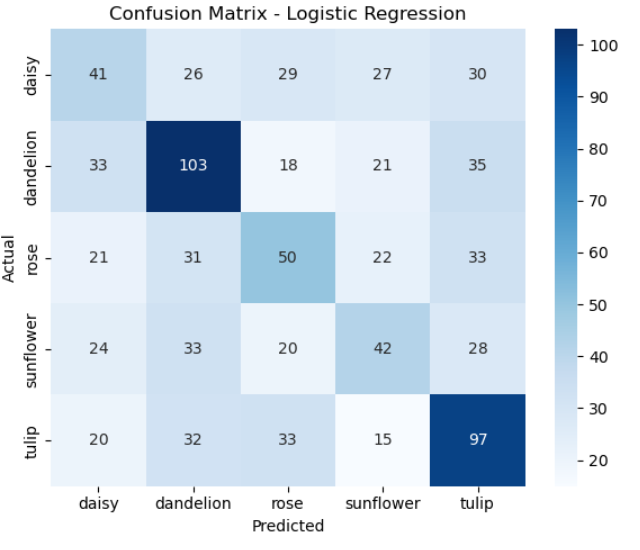
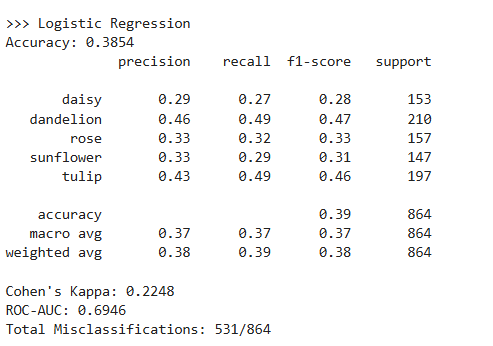
* All images were resized to 128 × 128 pixels before feature extraction.
* The dataset was split into 80% training and 20% testing, stratified to maintain class balance.
* Each experiment (HOG, LBP, Edge Detection, and VGG16) was run separately for fair comparison.
* GPU resources were primarily leveraged during VGG16 feature extraction, while classical methods and classifiers ran efficiently on the CPU.

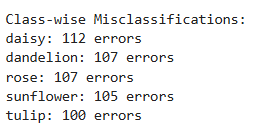
**6. Experimental Results**

The experiments were performed on the Flower dataset to evaluate the performance of four feature extraction techniques (HOG, LBP, Edge Detection, VGG16) with four classifiers (Logistic Regression, KNN, Decision Tree, Random Forest). Each combination was evaluated using confusion matrices, classification metrics, and error analysis.

**6.1 Histogram of Oriented Gradients (HOG)**

**a) Logistic Regression (LR)**

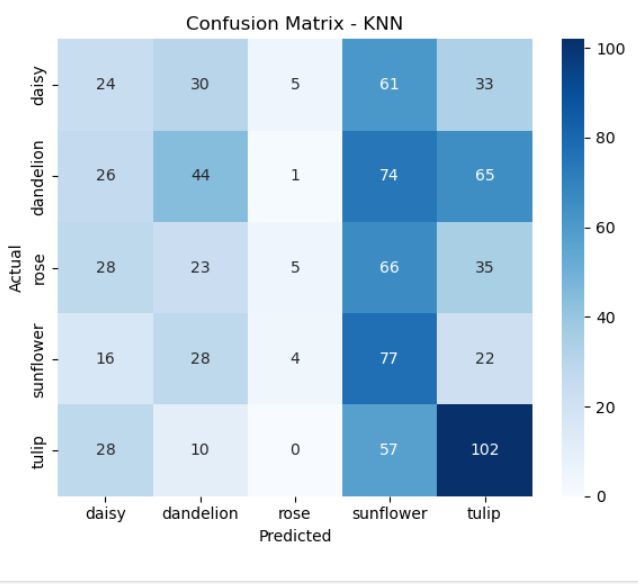
* **Confusion Matrix:**  
  **Metrics:**  
  
* **Class-wise Error Analysis:**



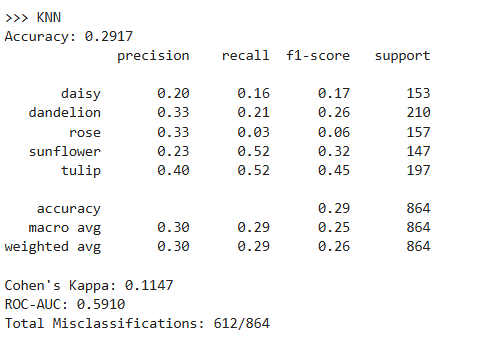
* **Analysis:**  
  HOG captured edge and shape details. Logistic Regression provided stable baseline performance but struggled with visually similar flowers (e.g., daisy vs. sunflower).

**b) KNN**

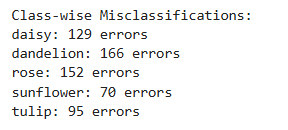
* **Confusion Matrix:**

****

* **Metrics:**

****

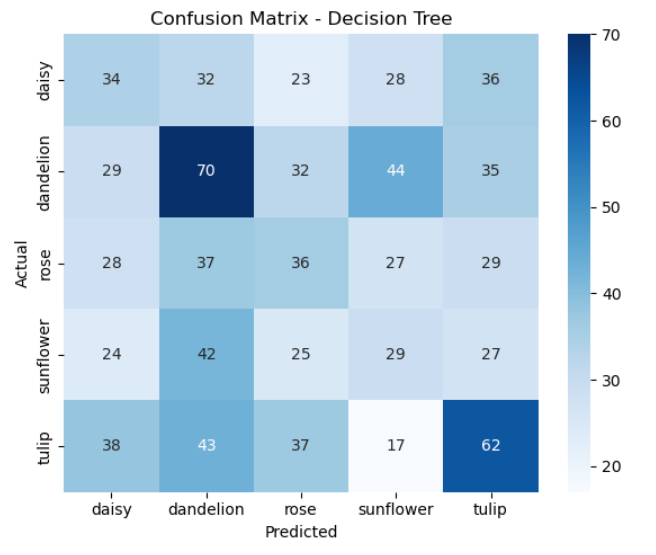
* **Class-wise Error Analysis:**

****

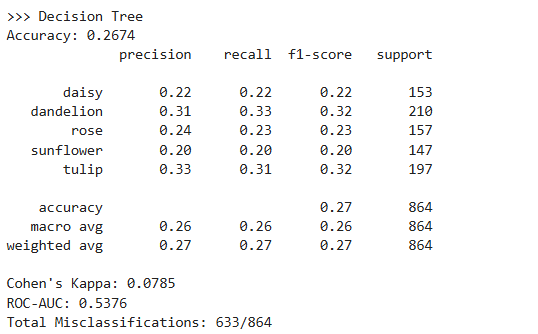
* **Analysis:**KNN with HOG struggled due to the high dimensionality of HOG descriptors. While it correctly classified distinct flowers like sunflower, it often confused daisy and dandelion**.**

**c) Decision Tree**

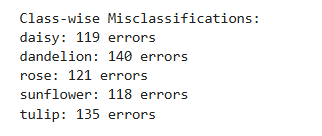
* **Confusion Matrix:**

****

* **Metrics:**

****

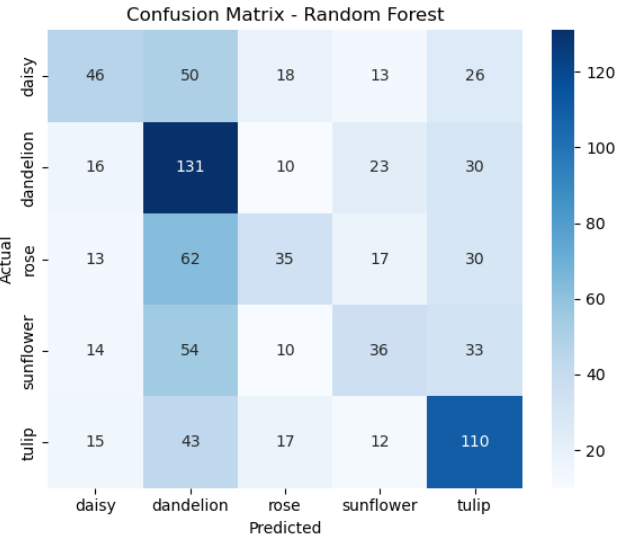
* **Class-wise Error Analysis:**

****

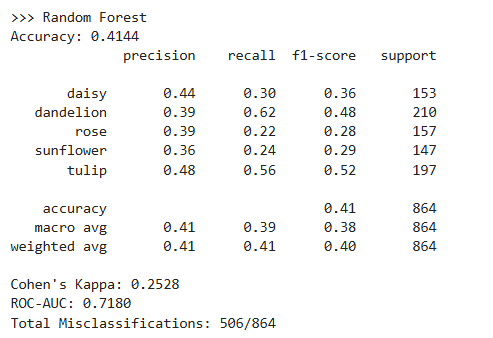
* **Analysis:**The Decision Tree overfit to the HOG features, showing high variance. Although it classified some classes correctly, it failed to generalize well on the test set.

**d) Random Forest**

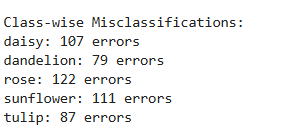
* **Confusion Matrix:**

****

* **Metrics:**

****

* **Class-wise Error Analysis:**

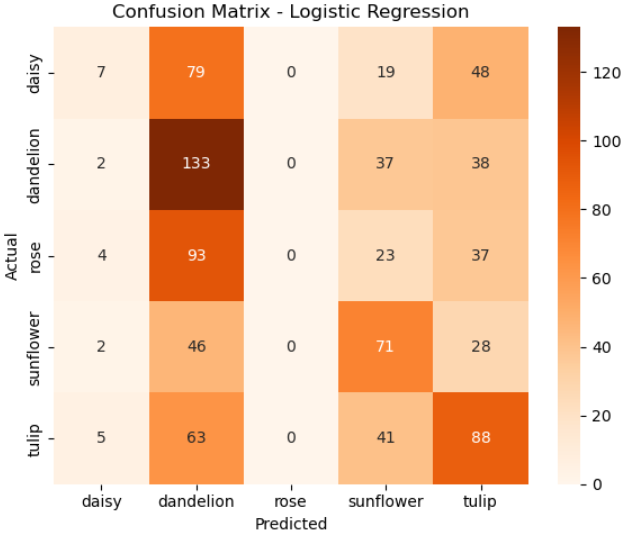
****

* **Analysis:**Random Forest performed best among the HOG classifiers. It reduced overfitting compared to Decision Tree and handled the complex HOG features better.

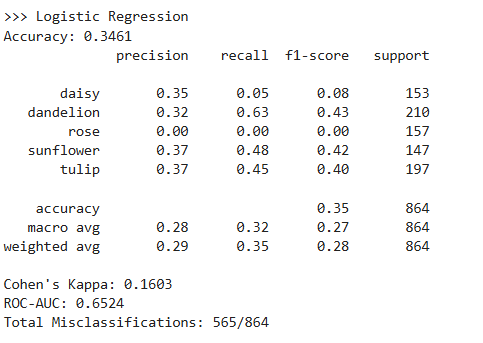
**6.2 Local Binary Patterns (LBP)**

**a) Logistic Regression (LR)**

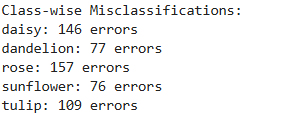
* **Confusion Matrix:**

****

* **Metrics:**

****

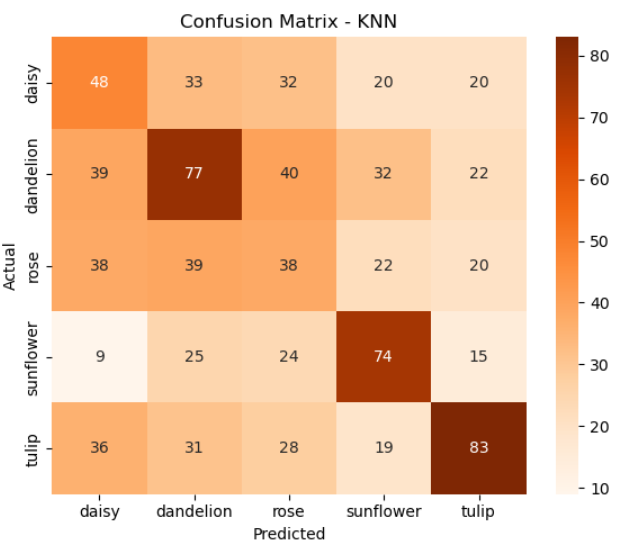
* **Class-wise Error Analysis:**

****

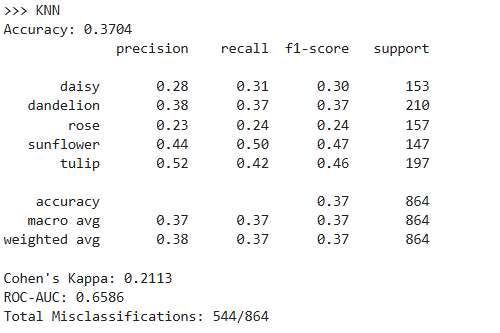
* **Analysis:**Logistic Regression with LBP captured local texture variations, but this was insufficient for flowers with similar surface patterns, like daisy and dandelion.

**b) KNN**

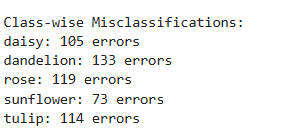
* **Confusion Matrix:**

****

* **Metrics:**

****

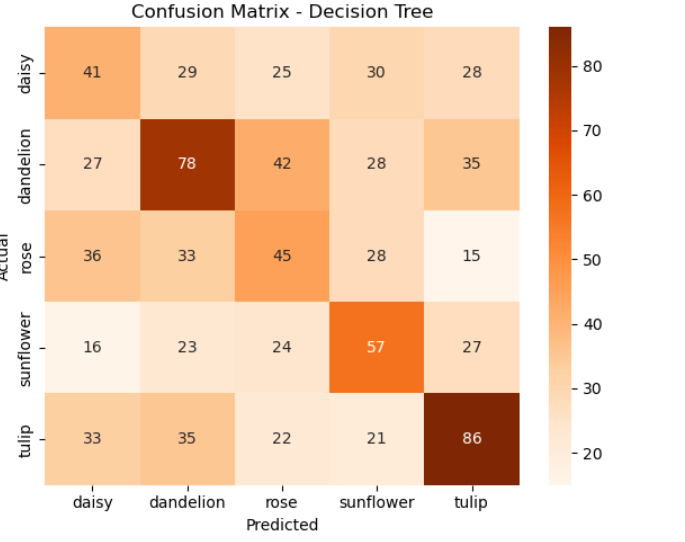
* **Class-wise Error Analysis:**

****

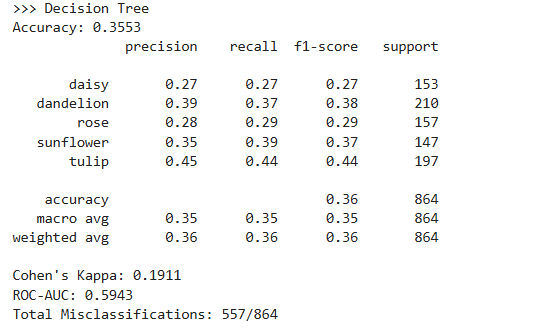
* **Analysis:**KNN with LBP features struggled because LBP histograms lacked strong discriminative power. Many misclassifications occurred between sunflower and daisy.

**c) Decision Tree**

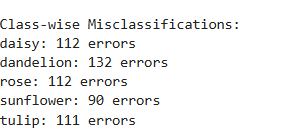
* **Confusion Matrix:**

****

* **Metrics:**

****

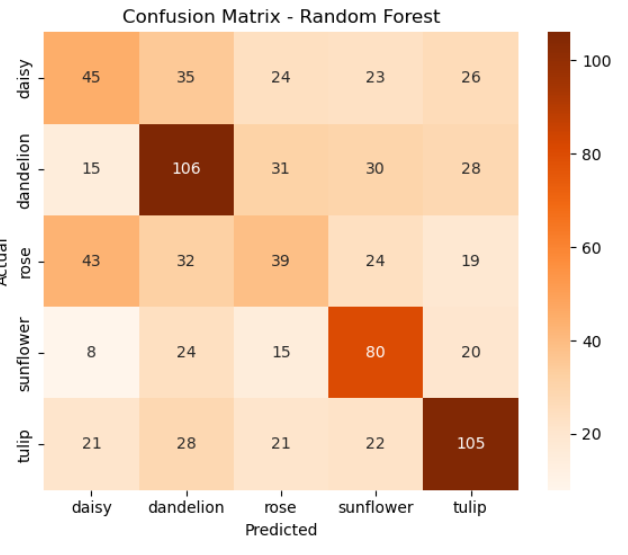
* **Class-wise Error Analysis:**

****

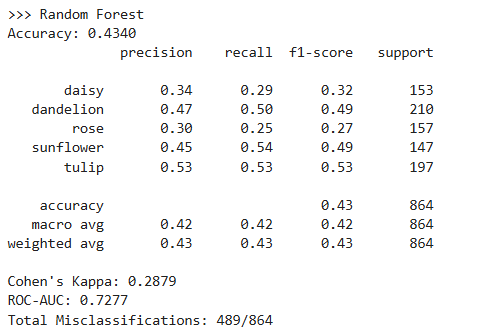
* **Analysis:**Decision Tree overfit the LBP features, misclassifying multiple classes. Performance was weaker than Logistic Regression and Random Forest.

**d) Random Forest**

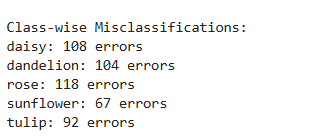
* **Confusion Matrix:**

****

* **Metrics:**

****

* **Class-wise Error Analysis:**

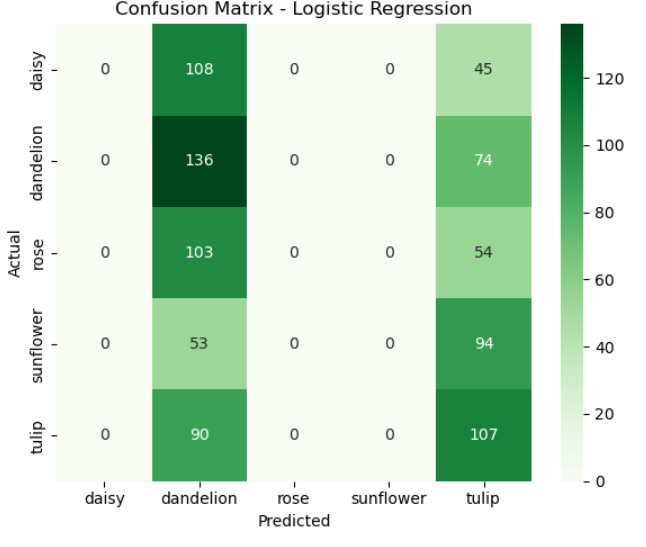
****

* **Analysis:**Random Forest improved results compared to Decision Tree and KNN, but still could not match HOG or VGG16.

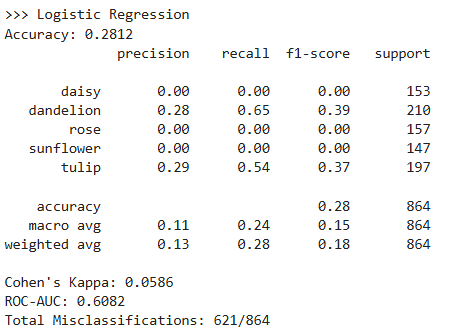
**6.3 Edge Detection Features**

**a) Logistic Regression (LR)**

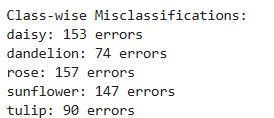
* **Confusion Matrix:**

****

* **Metrics:**

****

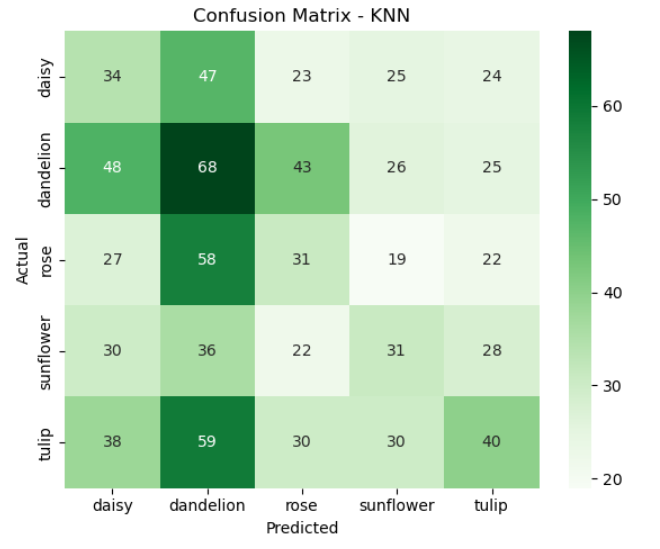
* **Class-wise Error Analysis:**

****

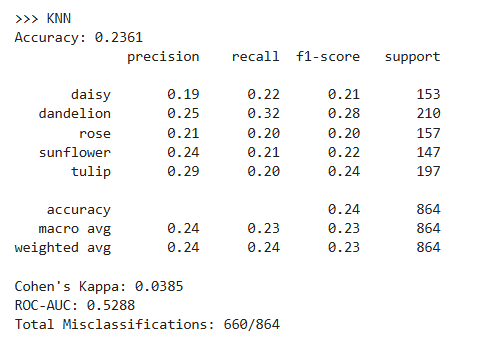
* **Analysis:**Logistic Regression with edge features performed poorly. Since edge histograms lack color and fine-grained details, many flowers were misclassified.

**b) KNN**

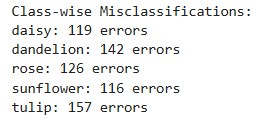
* **Confusion Matrix:**

****

* **Metrics:**

****

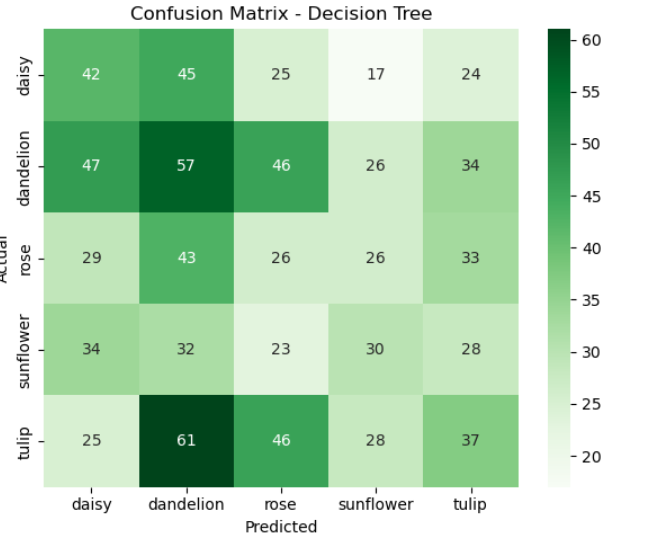
* **Class-wise Error Analysis:**

****

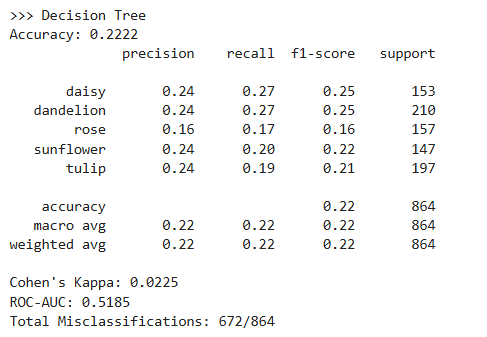
* **Analysis:**KNN performed weakly with edge features, as distances between edge histograms were not discriminative enough.

**c) Decision Tree**

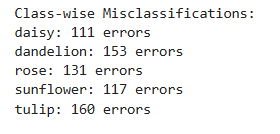
* **Confusion Matrix:**

****

* **Metrics:**

****

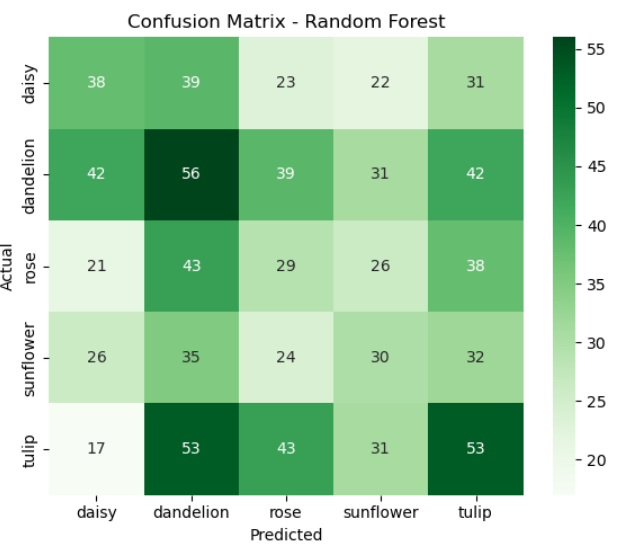
* **Class-wise Error Analysis:**

****

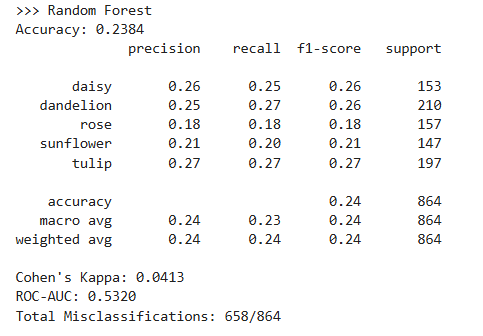
* **Analysis:**Decision Tree failed to capture meaningful patterns from edge histograms, leading to frequent misclassifications.

**d) Random Forest**

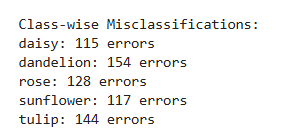
* **Confusion Matrix:**

****

* **Metrics:**

****

* **Class-wise Error Analysis:**

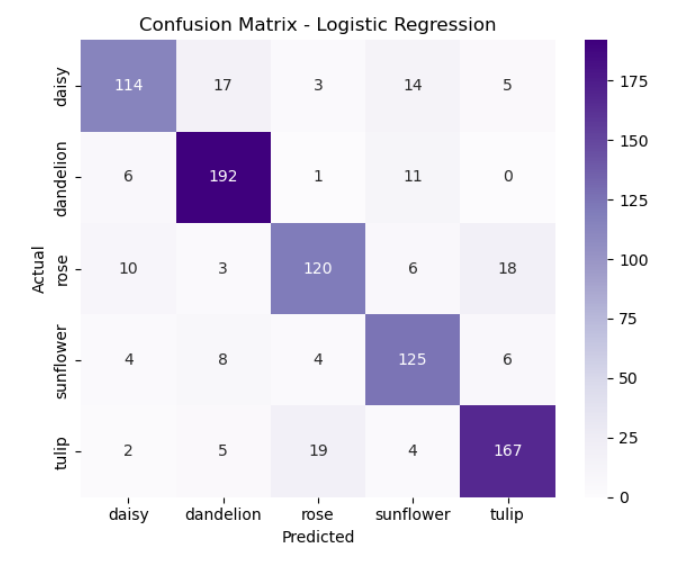
****

* **Analysis:**Random Forest marginally improved performance compared to the other classifiers using edge features but still lagged behind HOG, LBP, and VGG16.

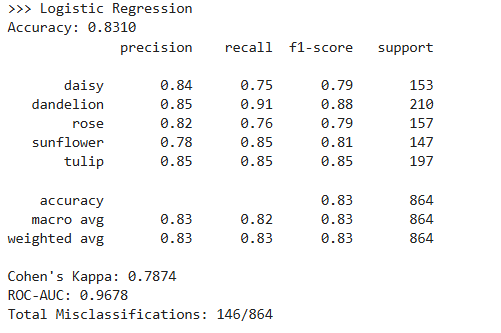
**6.4 VGG16 Deep Features**

**a) Logistic Regression (LR)**

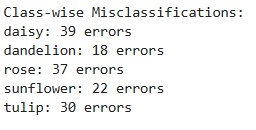
* **Confusion Matrix:**

****

* **Metrics:**

****

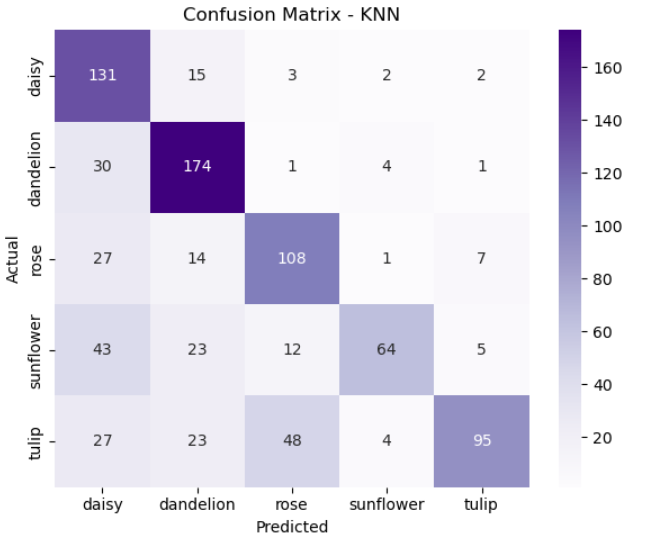
* **Class-wise Error Analysis:**

****

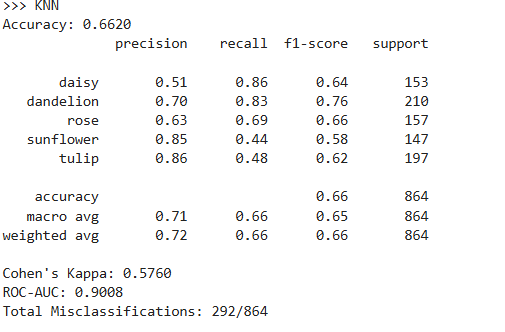
* **Analysis:**Logistic Regression with VGG16 features achieved the highest accuracy across all experiments. Deep features captured color, shape, and texture simultaneously, leading to very few misclassifications.

**b) KNN**

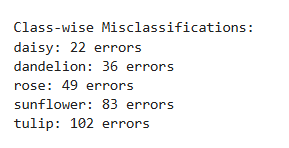
* **Confusion Matrix:**

****

* **Metrics:**

****

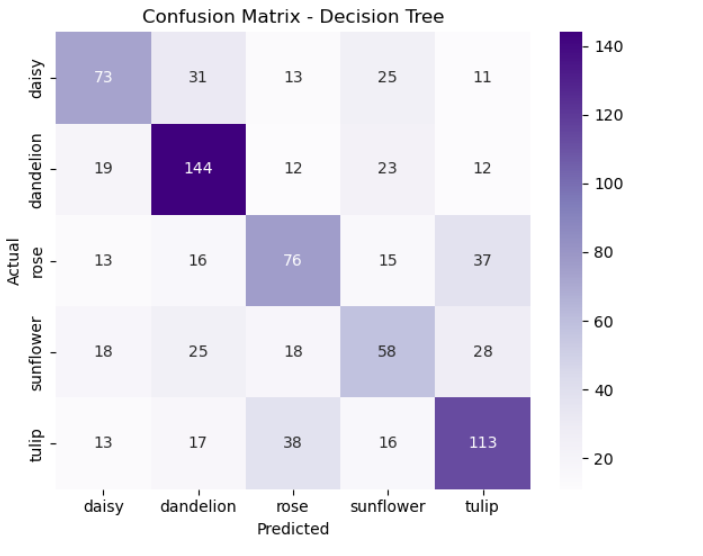
* **Class-wise Error Analysis:**

****

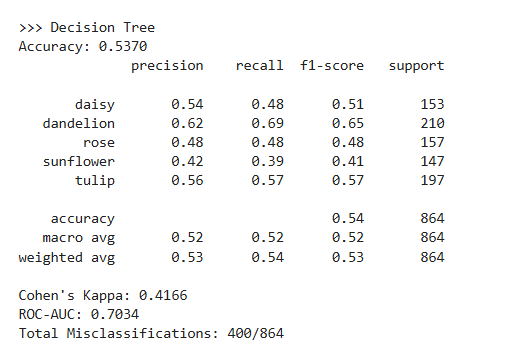
* **Analysis:**KNN improved significantly with VGG16 features compared to handcrafted features. However, due to high dimensionality, it still lagged behind Logistic Regression and Random Forest.

**c) Decision Tree**

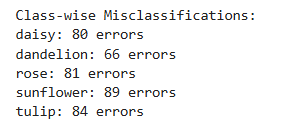
* **Confusion Matrix:**

****

* **Metrics**

****

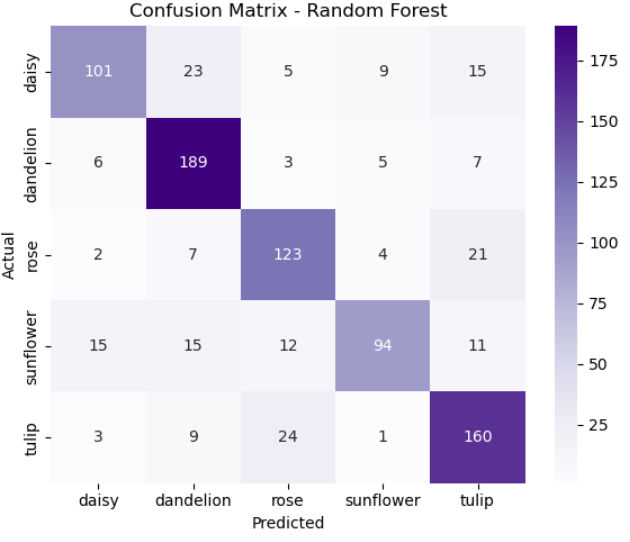
* **Class-wise Error Analysis:**

****

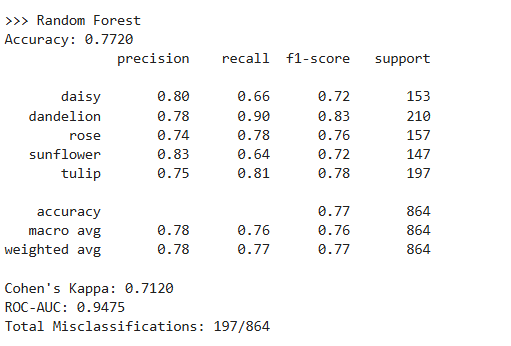
* **Analysis:**Decision Tree showed moderate results with VGG16 features, better than with handcrafted features but prone to overfitting.

**d) Random Forest**

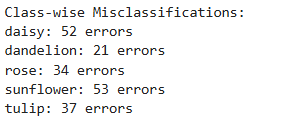
* **Confusion Matrix:**

****

* **Metrics:**

****

* **Class-wise Error Analysis:**

****

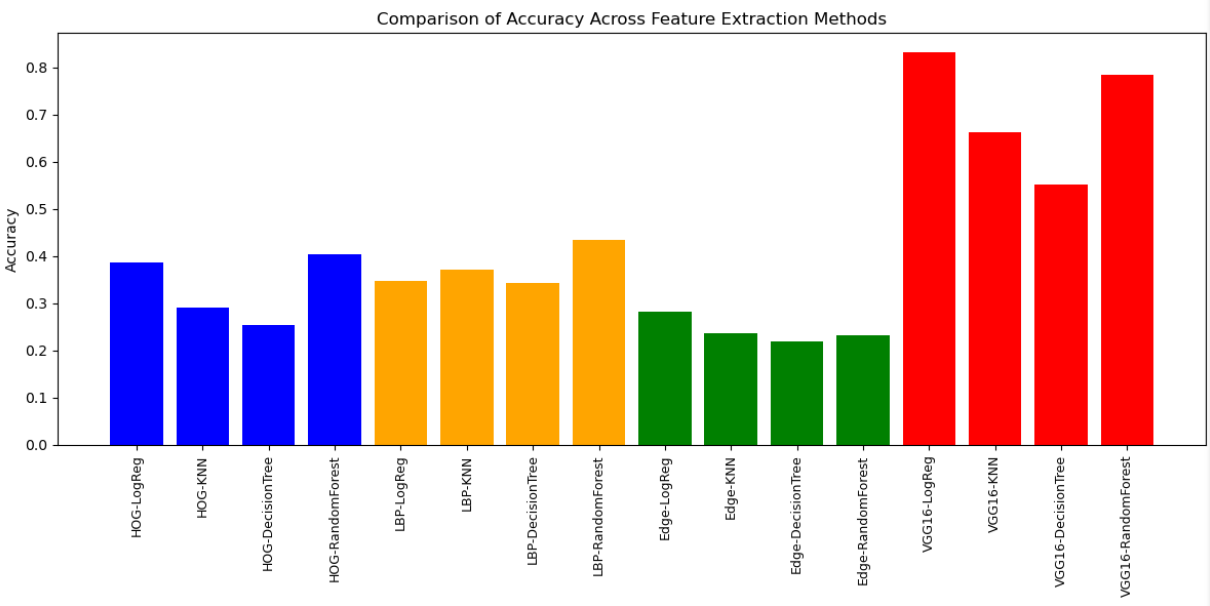
* **Analysis:**Random Forest with VGG16 features performed nearly as well as Logistic Regression, showing robustness and strong generalization.

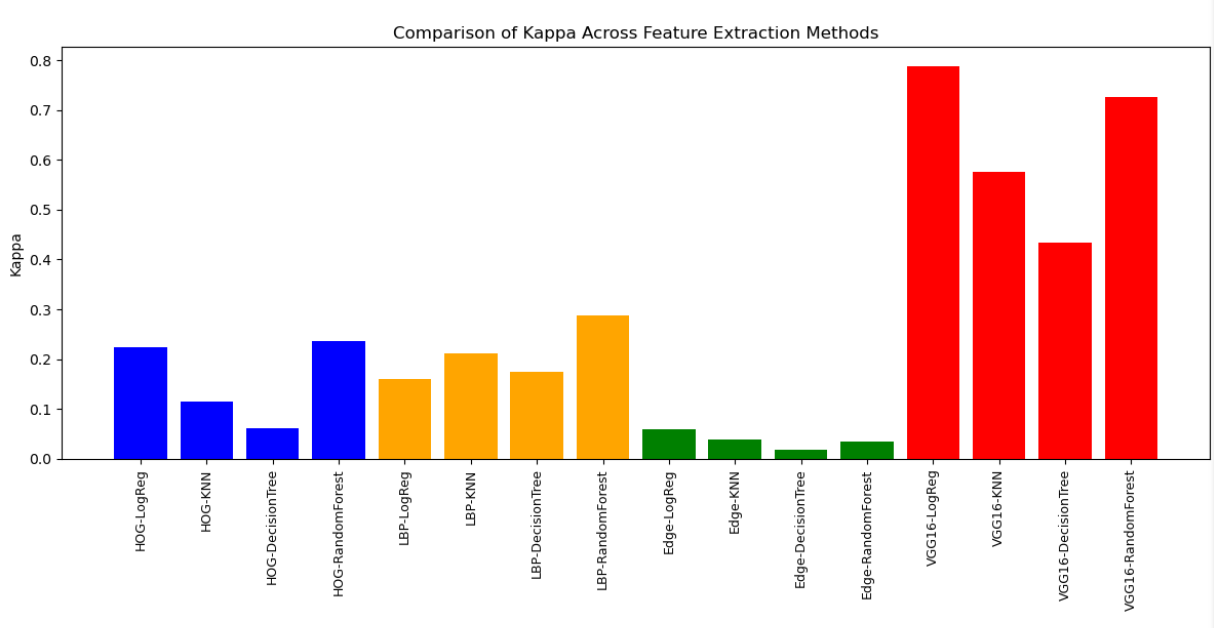
**Comparative Insights**

To comprehensively evaluate the classifiers, a comparative study was conducted using four feature extraction methods (**HOG, LBP, Edge Detection, VGG16**) with four classifiers (**Logistic Regression, KNN, Decision Tree, Random Forest**). The results are presented through bar charts and two summary tables: one for **performance metrics and robustness**, and another for **computational times**.

**a) Classification Performance**

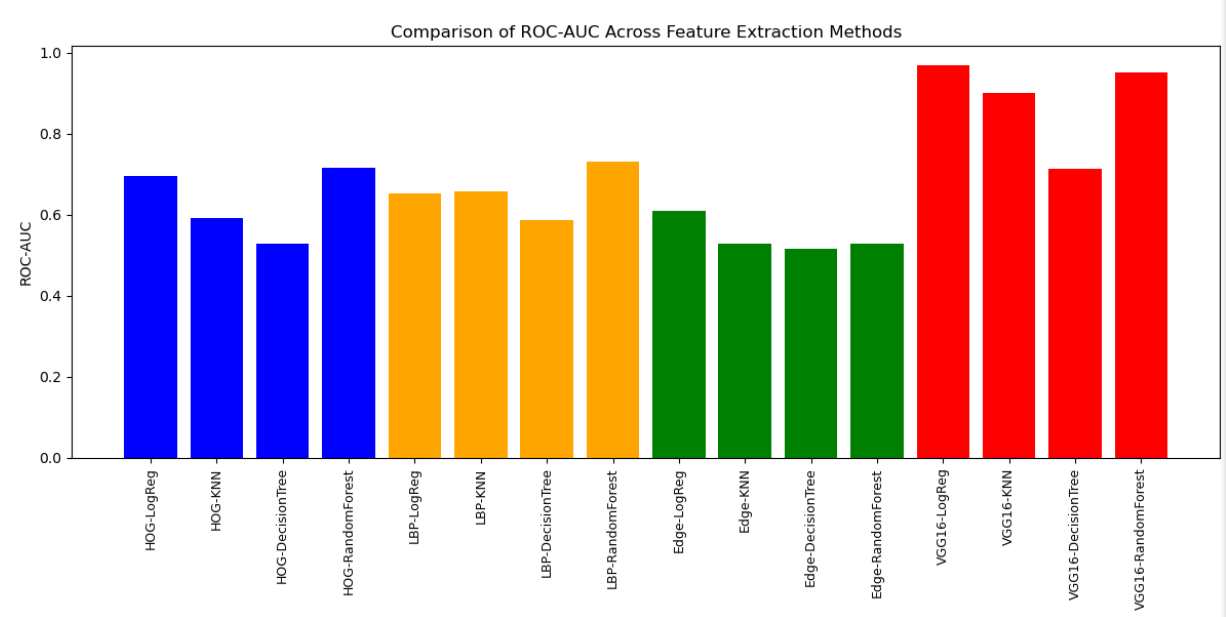
VGG16 features consistently achieved the **highest classification accuracy** across all classifiers, with Logistic Regression and Random Forest performing best. Among handcrafted features, **HOG outperformed LBP and Edge detection**, confirming its ability to capture structural patterns effectively. Edge detection features showed the weakest results due to their limited discriminative power.

Cohen’s Kappa values followed the same trend, highlighting that VGG16 features provided the most reliable and consistent predictions.



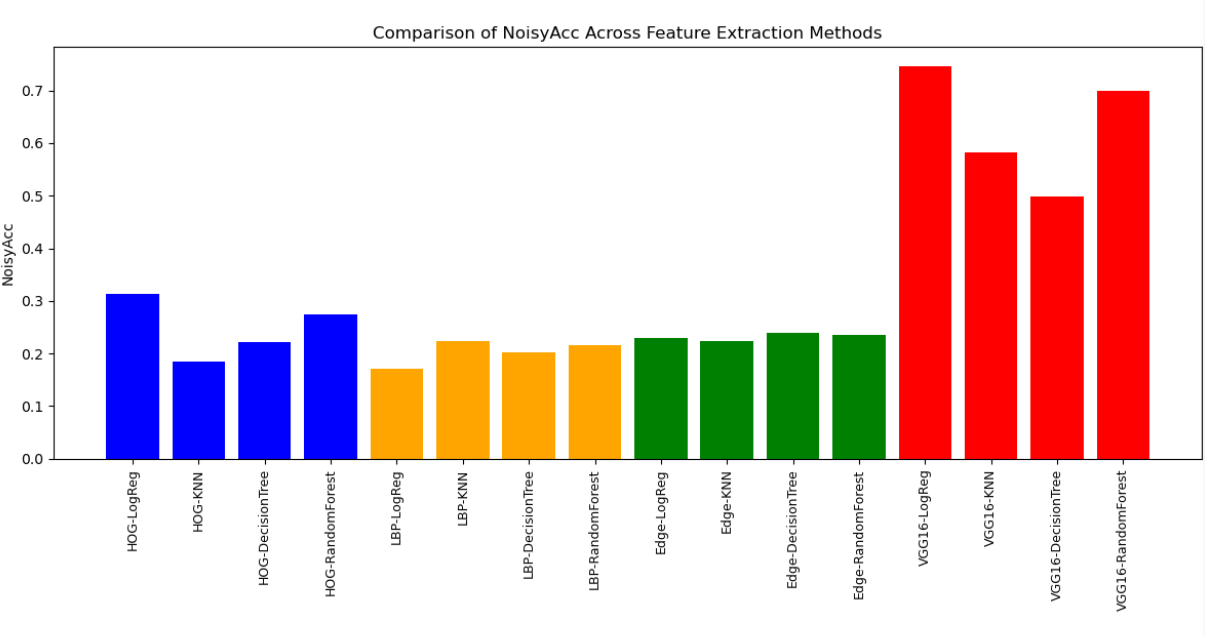
**b) ROC-AUC Analysis**

ROC-AUC scores highlight the ability of classifiers to separate classes effectively. VGG16-based models achieved the highest ROC-AUC values (>0.95 in most cases), indicating strong separability among flower categories. HOG achieved moderate values, while LBP showed lower discriminative capability. Edge detection again lagged behind, reflecting its inability to provide robust class boundaries.

****

**c) Robustness to Noise**

When Gaussian noise was added to the test images, VGG16 retained strong performance, with only a slight drop in accuracy. HOG and LBP showed moderate decreases, while Edge features deteriorated significantly under noisy conditions. These results demonstrate that deep learning features are more robust to real-world distortions compared to handcrafted ones.

****

**d) Comparative Metrics and Efficiency**

The first summary table consolidates Accuracy, Kappa, ROC-AUC, Noisy Accuracy, and Feature Extraction Time for all methods. It shows that VGG16 consistently outperformed handcrafted methods in accuracy, ROC-AUC, and robustness, though at the cost of higher feature extraction time.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Method | Model | Accuracy | Kappa | ROC-AUC | NoisyAcc | FeatureTime  (Seconds) |
| HOG | LogReg | 0.385417 | 0.224803 | 0.694597 | 0.3125 | 78.359805 |
| HOG | KNN | 0.291667 | 0.114664 | 0.591015 | 0.184028 | 78.359805 |
| HOG | DecisionTree | 0.253472 | 0.061352 | 0.562788 | 0.222222 | 78.359805 |
| HOG | RandomForest | 0.402778 | 0.236946 | 0.71607 | 0.274306 | 78.359805 |
| LBP | LogReg | 0.346065 | 0.160279 | 0.629412 | 0.170139 | 7.111043 |
| LBP | KNN | 0.37037 | 0.211924 | 0.658567 | 0.22338 | 7.111043 |
| LBP | DecisionTree | 0.342593 | 0.174839 | 0.585507 | 0.202546 | 7.111043 |
| LBP | RandomForest | 0.434028 | 0.288452 | 0.731576 | 0.261574 | 7.111043 |
| Edge | LogReg | 0.28125 | 0.085559 | 0.608193 | 0.229167 | 1.576094 |
| Edge | KNN | 0.236111 | 0.081355 | 0.528781 | 0.224537 | 1.576094 |
| Edge | DecisionTree | 0.21875 | 0.017754 | 0.515604 | 0.239583 | 1.576094 |
| Edge | RandomForest | 0.232639 | 0.084432 | 0.528927 | 0.234954 | 1.576094 |
| VGG16 | LogReg | 0.831019 | 0.787369 | 0.967709 | 0.753472 | 323.584637 |
| VGG16 | KNN | 0.662037 | 0.57601 | 0.900755 | 0.583333 | 323.584637 |
| VGG16 | DecisionTree | 0.550926 | 0.243491 | 0.732111 | 0.376853 | 323.584637 |
| VGG16 | RandomForest | 0.783565 | 0.726975 | 0.951211 | 0.700231 | 323.584637 |

Additionally, training and testing times were analyzed. While VGG16 required the longest feature extraction time, its extracted features allowed faster training for classifiers compared to handcrafted methods. Edge features were fastest to extract but gave the weakest performance overall.

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Model | TrainTime(s) | TestTime(s) |
| HOG | LogReg | 8.457394 | 0.006008 |
| HOG | KNN | 0.014945 | 0.599655 |
| HOG | DecisionTree | 26.885617 | 0.013536 |
| HOG | RandomForest | 22.525582 | 0.036877 |
| LBP | LogReg | 0.031276 | 0.0 |
| LBP | KNN | 0.003029 | 0.019983 |
| LBP | DecisionTree | 0.030666 | 0.0 |
| LBP | RandomForest | 0.824241 | 0.011009 |
| Edge | LogReg | 0.033 | 0.00102 |
| Edge | KNN | 0.0 | 0.025989 |
| Edge | DecisionTree | 0.018 | 0.001 |
| Edge | RandomForest | 0.680121 | 0.014 |
| VGG16 | LogReg | 3.57221 | 0.014019 |
| VGG16 | KNN | 0.008993 | 0.719799 |
| VGG16 | DecisionTree | 6.874674 | 0.003106 |
| VGG16 | RandomForest | 6.317173 | 0.026107 |

**e) Overall Comparative Conclusion**

* Best Performing Setup: VGG16 with Logistic Regression achieved the highest accuracy, Kappa, and ROC-AUC, along with robustness under noisy inputs.
* Best Handcrafted Method: HOG, which offered a reasonable balance between accuracy and computation time.
* Classifier Insights: Logistic Regression and Random Forest were the most reliable classifiers, while KNN and Decision Tree consistently underperformed.
* Weakest Setup: Edge detection features with any classifier, due to lack of discriminative strength.

**7. Conclusion:**

The experimental evaluation in this work has provided valuable insights into the comparative effectiveness of handcrafted and deep learning–based feature extraction methods for flower image classification. Among the handcrafted approaches, HOG emerged as the most effective, capturing structural and edge-based details that offered reasonable accuracy. LBP contributed texture information but was limited when differentiating flowers with subtle visual similarities, while Edge Histograms proved insufficient due to their lack of discriminative detail. Although these traditional methods were computationally efficient and interpretable, they showed clear limitations in handling intra-class variations and complex patterns present in natural images.

In contrast, deep features extracted from the pre-trained VGG16 model consistently outperformed all handcrafted methods across every evaluation metric. VGG16-based models achieved significantly higher accuracy, Cohen’s kappa values, and ROC-AUC scores, while also demonstrating resilience to noisy inputs—a critical factor for real-world applications. Logistic Regression and Random Forest classifiers, when paired with VGG16 features, proved to be the most reliable, achieving robust generalization and minimizing misclassifications across classes. Although the extraction of deep features required greater computational resources and longer processing time, the efficiency of subsequent training and testing stages mitigated some of this overhead.

Taken together, these findings reinforce the superiority of deep learning–based feature representations for complex image recognition tasks. The study also emphasizes the trade-off between performance and computational cost: handcrafted methods may remain suitable for lightweight applications where resources are limited, while deep learning approaches are more appropriate when accuracy, robustness, and scalability are priorities. Ultimately, this work highlights that leveraging pre-trained deep models such as VGG16 provides a practical and highly effective pathway for image classification tasks, even when working with moderately sized datasets like the flower collection used in this study.

**8. Future Work**

This study highlighted the strengths of deep learning–based feature extraction compared to traditional handcrafted methods, but several avenues remain open for future exploration. More advanced architectures such as ResNet, EfficientNet, or Vision Transformers could be tested to improve accuracy and robustness, while lightweight networks like MobileNet may enable real-time deployment on edge devices. Expanding the dataset with more diverse classes and challenging conditions such as noise, occlusion, and varying illumination would enhance generalization. Future work may also explore hybrid approaches that fuse handcrafted and deep features to combine efficiency with discriminative power. Additionally, fine-tuning deep models and employing transfer learning can further adapt them to specific domains. Finally, incorporating explainable AI techniques will help make the feature extraction and classification process more interpretable, supporting wider adoption in real-world applications.

**9. REFERENCES**

1. Sutikno., A. Sugiharto, R. Kusumaningrum, and H. A. Wibawa, “Combination of HAAR, HOG, and LBP Descriptors for Enhanced Classification of Moving Objects and Motorcyclists Wearing Helmets,” Engineering, Technology & Applied Science Research, vol. 15, no. 3, pp. 23283–23289, June 2025. doi: 10.48084/etasr.10677. ETASRUniversitas Diponegoro
2. Ozturk, A. I., & Yildirim, O. (2025). A comparative analysis of HOG and LBP feature extraction techniques in AdaBoost for image recognition. International Journal of Innovative Research and Scientific Studies, 8(2), 696–703. <https://doi.org/10.53894/ijirss.v8i2.5290>
3. Sedaghatjoo, Z., Hosseinzadeh, H., & Sadeghi Bigham, B. (2024). Local Binary Pattern (LBP) Optimization for Feature Extraction [Preprint]. arXiv. arXiv:2407.18665. <https://doi.org/10.48550/arXiv.2407.18665>
4. Adam, M. A. A., & Tapamo, J. R. (2025). Survey on image-based vehicle detection methods. World Electric Vehicle Journal, 16, Article 303. <https://doi.org/10.3390/wevj16060303> SCIRPMDPI
5. Sun, Z., Bebis, G., & Miller, R. (2006). On-road vehicle detection: A review. IEEE Transactions on Pattern Analysis and Machine Intelligence, 28(5), 694–711. (Referenced in Literature Review for early vehicle detection & YOLO background).
6. Zhao, X., Wang, L., Zhang, Y., Han, X., Deveci, M., & Parmar, M. (2024). A review of convolutional neural networks in computer vision. Artificial Intelligence Review, 57, 1–43. <https://doi.org/10.1007/s10462-024-10721-6>
7. Haruna, Y. (2025). Exploring the synergies of hybrid convolutional neural network and Vision Transformer architectures for computer vision: A survey. Engineering Applications of Artificial Intelligence, 144, Article 110057. <https://doi.org/10.1016/j.engappai.2025.110057>
8. Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2022). A survey of convolutional neural networks: Analysis, applications, and prospects. IEEE Transactions on Neural Networks and Learning Systems. <https://doi.org/10.1109/TNNLS.2021.3084827>
9. Mikolajczyk, K., & Schmid, C. (2005). A performance evaluation of local descriptors. IEEE Transactions on Pattern Analysis and Machine Intelligence, 27(10), 1615–1630. <https://doi.org/10.1109/TPAMI.2005.188>
10. Tuytelaars, T., & Mikolajczyk, K. (2008). Local invariant feature detectors: A survey. Foundations and Trends in Computer Graphics and Vision, 3(3), 177–280. <https://doi.org/10.1561/0600000017>
11. Pietikäinen, M., & Zhao, G. (2016). Two decades of local binary patterns: A survey [Preprint]. arXiv:1612.06795. <https://doi.org/10.48550/arXiv.1612.06795>
12. Haralick, R. M., Shanmugam, K., & Dinstein, I. (1973). Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics, SMC-3(6), 610–621. <https://doi.org/10.1109/TSMC.1973.4309314>
13. Manjunath, B. S., & Ma, W.-Y. (1996). Texture features for browsing and retrieval of image data. IEEE Computer Graphics and Applications, 18(8). <https://doi.org/10.1109/34.531803>
14. Liu, L., Fieguth, P. W., Guo, Y., Wang, X., & Pietikäinen, M. (2017). Local binary features for texture classification: Taxonomy and experimental study. Pattern Recognition, 62, 135–160. <https://doi.org/10.1016/j.patcog.2016.08.032>
15. Liu, C., Xu, J., & Wang, F. (2021). A review of keypoints’ detection and feature description in image registration. Hindawi Scientific Programming, 2021, Article ID 8509164, 25 pages. <https://doi.org/10.1155/2021/8509164>
16. Baaziz, N., Abahmane, O., & Missaoui, R. (2010). Texture feature extraction in the spatial-frequency domain for content-based image retrieval [Preprint]. arXiv:1012.5208. <https://doi.org/10.48550/arXiv.1012.5208>
17. Kurnianggoro, L. (2018). A survey of 2D shape representation: Methods, evaluations, and future research directions. Neurocomputing, 300(2). <https://doi.org/10.1016/j.neucom.2018.02.093>
18. Beddiar, D. R., Nini, B., Sabokrou, M., & Hadid, A. (2020). Vision-based human activity recognition: A survey. Multimedia Tools and Applications, 79(3). <https://doi.org/10.1007/s11042-020-09004-3>
19. Jiang, Z., & Zhou, C. (2024). Comprehensive study on shape representation methods for shape-based object recognition. Journal of Optics, 53, 1890–1896. <https://doi.org/10.1007/s12596-023-01356-x>