SSRG International Journal of VLSI and Signal Processing
ISSN: 2394–2584 / https://doi.org/10.14445/23942584/IJVSP-V12I2P101

Volume 12 Issue 2, 1-8, May-Aug 2025

Original Article

Leaf Identification Using SIFT Feature Extraction and SVM on PYNQZU

Mereddy Jahnavi[1], T. Satya Savithri[2]

[1,2]Department of Electronics and Communication Engineering, JNTUH University College Of Engineering, Science &Technology, Hyderabad, Telangana, India

[1]Corresponding Author : mereddyjahnavi@gmail.com

Received: 03 June 2025 Revised: 05 July 2025 Accepted: 28 July 2025 Published: 14 August 2025

Abstract - *This paper presents leaf identification utilizing an SVM classifier, based on features obtained through SIFT, VLAD, and PCA methodologies. The feature extraction process, referred to as the SVP technique, involves identifying key points using SIFT, converting these key points into a standardized feature vector with VLAD, and then reducing the dimensionality of this feature vector via PCA. A novel VLSI architecture for the SVM classifier is introduced in this paper and has been implemented on the PL portion of the PYQZU board, while the feature extraction methods are connected using the PS section of the board with a Python program. The suggested method has been evaluated using standard datasets of Apple and Cherry and custom datasets for Custard and Mango. The results obtained from the hardware implementation are comparable to those from the software, and the proposed approach yields effective outcomes even under acquisition disturbances, such as low lighting and rotation. The SVM classifier is employed for single-label classification and identification; however, there is a requirement for multilabel classification for images featuring multiple leaf types, which is addressed in this paper through the proposal of a Binary Relevance Strategy.*

Keywords - *SIFT, VLAD and PCA methodologies, Leaf Identification, SVM classifier, Support Vector Machine.*

1. Introduction

In recent years, the automation of leaf identification has gained significant importance due to the increasing demand for precision agriculture and efficient crop management. However, automating the processing of agricultural images necessitates various methods to extract the relevant information from crops over time. While several research initiatives have been undertaken in this area, there remains a need for new methodologies to enhance automation applications in precision agriculture. The primary challenges in automation arise from variations in species, leaf orientation, illumination conditions, and the presence of background noise. Selecting appropriate feature extraction techniques is crucial due to factors such as illumination, background noise, and image-dependent conditions. Therefore, capture-invariant features like leaf texture, shape, venation patterns, and boundary contours are essential for these automation techniques. This paper discusses the method to capture these features and facilitate better leaf identification.

Some relevant approaches for feature extraction have been documented in the literature. The Scale-Invariant Feature Transform (SIFT) algorithm and implementation as proposed in [1] highlights the importance of local features,[2] proposes a method for recognizing leaves based on texture features utilizes pixel relationships through GLCM and Principal Component Analysis (PCA).[3] focuses on the colour feature of the leaf using HSV color space, and [4] incorporates the Canny edge detector to extract shape-based features, alongside Support Vector Machine (SVM) for leaf classification. [5] introduces SIFT and Bag-of-Words (BOW) for detecting leaf diseases. The BOW converts keypoints into a histogram and identifies them based on their frequency of occurrence..Lastly, [6] gives a hardware implementation of SIFT and BOW methods for classification using SVM.

This paper proposes a new technique for leaf identification by utilizing various feature extraction methods. The SVP (SIFT-VLAD-PCA) technique combines SIFT, VLAD (Vector of Locally Aggregated Descriptors), and PCA. Unlike the BOW model, which loses spatial information due to histogram representation, VLAD helps retain discriminative information. For classification and identification, two methodologies are presented: an SVM classifier for single-label leaf classification and a Binary Relevance Classifier (BRC) for multilabel leaf classification. Both methodologies are proposed and compared. Additionally, novel VLSI architectures for the SVM and BRC are introduced, and a complete framework for leaf identification is implemented on the Pynq ZU board.

This paper is structured in the following manner: Section 2 addresses feature extraction techniques, Section 3 presents SVM and BRC classifiers for leaf classification and identification, and Section 4 presents the VLSI architecture.

Section 5 covers the implementation of the proposed framework along with the experimental results and performance evaluation, and Section 6 discusses the conclusion.

2. Feature Extraction

This section explores an SVP technique for local feature extraction by detecting key points from SIFT and converting these to a unified feature vector using the VLAD method. PCA is then used to find the reduced-dimensional feature vector.

2.1. SIFT

The SIFT is an effective method for extracting local features used to identify and characterize unique patterns in images. It excels in recognizing structural details in leaves that hold steady despite changes in scale, rotation, or lighting conditions. These descriptors aid in differentiating subtle structural variations in leaves, including vein patterns and serrated edges. The SIFT process is carried out in three primary stages:

2.1.1. Keypoint Detection (Scale invariance)

In order to accomplish scale invariance, SIFT creates a scale-space by gradually applying Gaussian filters to blur the image. Subsequently, the *Difference of Gaussians (D)* is utilized to identify extrema—local maxima and minima—over various scales and locations within the image. These points generally align with corners, edges, and textured areas on the leaf surface, which possess reliable and unique features.

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{1}$$

2.1.2. Orientation Assignment (Rotation Invariance)

Every keypoint is given a primary orientation determined by the local gradient directions in an area surrounding the keypoint. The orientation that occurs most often is chosen as the reference direction. This process guarantees that all further calculations are based on this orientation, enhancing the descriptor's resistance to rotation.

2.1.3. Descriptor Generation (Illumination Robustness)

Surrounding the oriented keypoint, a window of fixed size (usually 16×16 pixels) is segmented into a 4×4 grid. A histogram of local gradient orientations is generated within each grid cell using 8 orientation bins. The gradient magnitude of each pixel is adjusted according to its strength, and a Gaussian function centered on the keypoint emphasizes the significance of gradients closer to the center. The resulting descriptor vector, which is 4×4×8 = 128-dimensional, encapsulates the local gradient structure. Utilizing gradient information offers a certain level of resilience against changes in illumination.

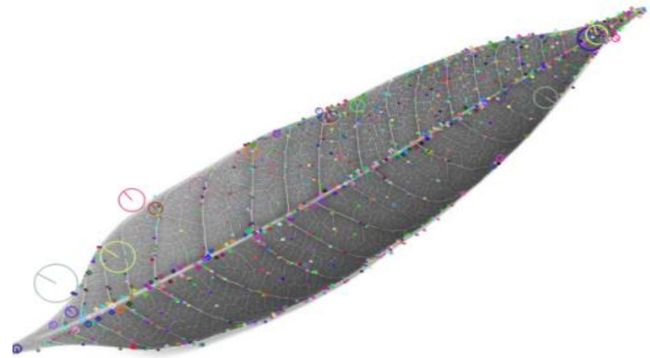

Fig. 1 SIFT key points on a test leaf

2.2. VLAD

This feature encoding method captures local pattern presence and encodes the changes of descriptors in relation to learned visual centers, resulting in a compact yet expressive overall representation of the image. A visual vocabulary is initially established by conducting k-means clustering on SIFT descriptors obtained from a training set of leaf images. Each SIFT descriptor from a specific image is then matched to its closest cluster center (visual word). VLAD calculates a residual vector for every descriptor, which is the difference between the descriptor and its associated center.

For each cluster k, all residuals $(x_i-\mu_k)$ of descriptors x_i assigned to that cluster are accumulated:

$$\vartheta_K = \sum_{x_i \in C_k} (x_i - \mu_k) \tag{2}$$

Where x_i is given vector and μ_k is the mean of the k cluster.

These accumulated vectors, v_k, capture each cluster's distribution and intensity of local feature deviations. The final feature vector for the image is formed by concatenating all v_k vectors across the clusters.

2.3. PCA

Since the concatenated VLAD vector can be high-dimensional, PCA is used to reduce the dimensionality. PCA projects the descriptor into a lower-dimensional space by retaining directions of maximum variance, thereby improving computational efficiency while preserving discriminative information essential for classification.

$$V_{PCA} = W^T (v - \mu) \tag{3}$$

$V_{PCA} \in R_d$ Is the projected (reduced) feature vector after PCA. This is the lower-dimensional representation of the original VLAD descriptor.

$V \in R_D$ Is the original high-dimensional VLAD feature vector before reduction?

$\mu \in R_D$ Is the mean vector of the training VLAD descriptors. Each component μ_j is the average value of the j-th feature across all training vectors. Subtracting μ centers the data.

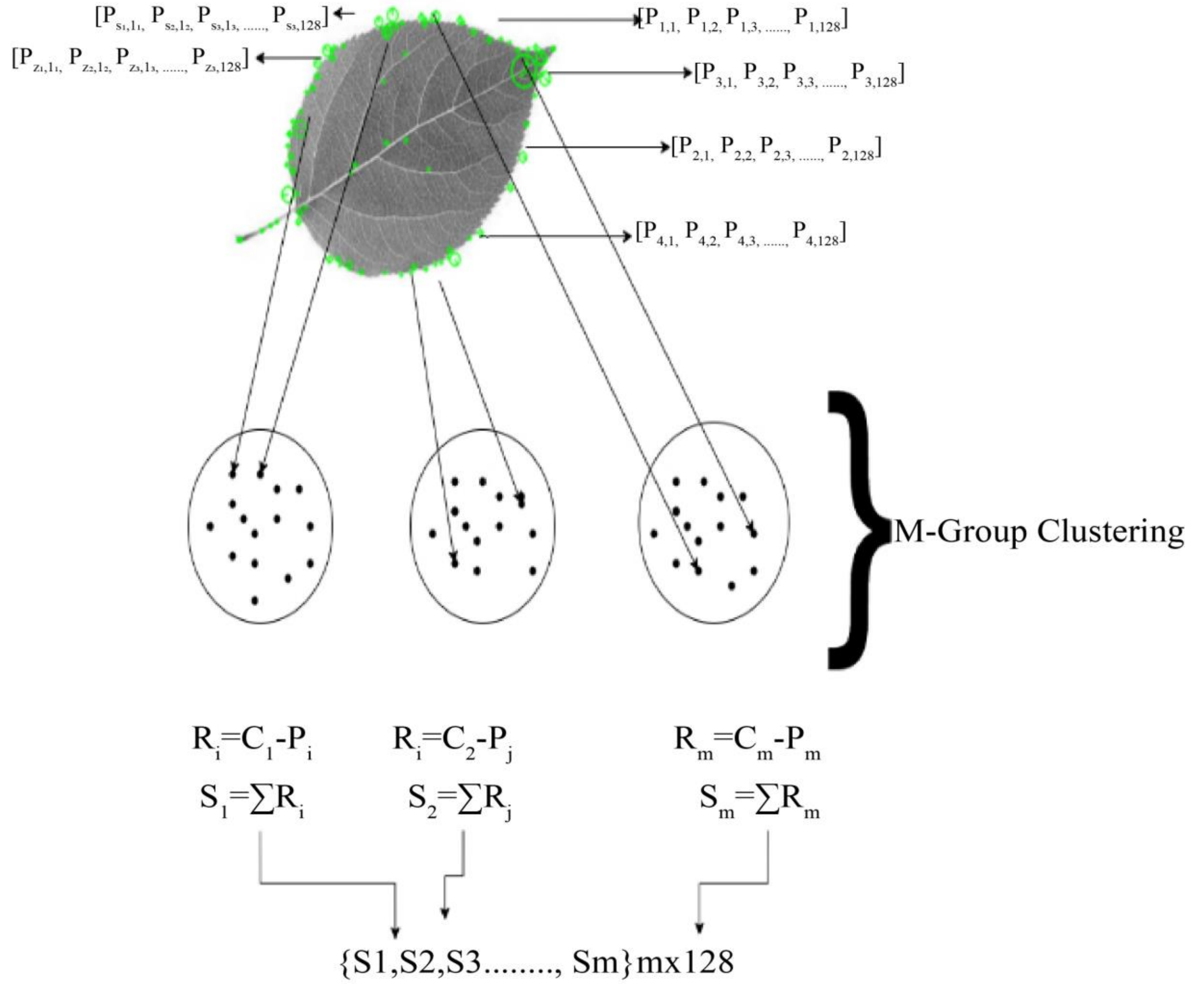

Fig. 2 SVP Feature Vector

$W \in R_{Dxd}$ The PCA projection matrix is composed of the top d eigenvectors (principal components) of the covariance matrix of the centered data. These eigenvectors correspond to the directions of maximum variance.

$W^T \in R_{Dxd}$ It is the transpose of the projection matrix, used to project the centered high-dimensional vector $v-\mu$ into a d-dimensional subspace.

This results in a fixed-length vector that preserves critical structural and textural cues useful for fine-grained recognition tasks.

3. Classification

In this paper, two approaches are used for classification and identification. Among them, one is an SVM classifier for single-label use, and the other is BRC for multilabel use.

3.1. SVM

SVM is a technique utilized for binary classification by creating an ideal hyperplane that helps to separate the data points of two independent classes. It is a supervised machine learning algorithm. In this work, a linear SVM is employed for leaf classification, where each leaf image is represented by an extracted feature vector.$x_i \in R_d$ and assigned a class label $y_i \in \{-1, +1\}$.

The aim is to identify a hyperplane specified by $w \cdot x + b = 0$, where $w \in R_d$ Is the weight vector and $b \in R_b$ Is the bias. The classifier predicts class labels based on the sign of $w \cdot x + b$. To allow for better generalization and accommodate non-separable data, an SVM with a soft margin is used. Hence, a slack variable $\xi_i \geq 0$ for misclassified or margin-violating points is introduced, leading to the following regularized optimization problem:

$$\min_{w,b,\xi} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \quad (4)$$

$$subject\ to\ y_i(w.x_i + b) \geq 1 - \xi_i, \xi_i \geq 0, \forall i \quad (5)$$

C>0 represents the regularization parameter that balances the goal of maximizing the margin with that of minimizing classification errors.

In the context of this paper, the binary classification task involves distinguishing between two leaf species based on the extracted SVP feature. A decision function for a sample x is given by:

$$f(x) = sign(w \cdot x + b) \quad (6)$$

3.2. Binary Relevance (BR) Classifier for Multilabel Classification

In scenarios where a single image may contain multiple leaf types or overlapping species, a multilabel classification approach is required. The BR strategy addresses this by decomposing the multilabel problem into independent binary classification tasks, one for each label.

In this paper, SVM is used for each label, and two SVMs are used to classify multiple leaves. The two SVMs are trained separately. During prediction, each SVM outputs a binary decision indicating the presence or absence of a specific class. The final output is constructed using one-hot or multi-hot encoding, depending on whether single or multiple labels are applicable per image.

$$f_k(x) = w_k.F(x) + b_k$$

$$y_k = sign(w_k.F(x) + b_k \quad (7)$$

$$y = [y_1, y_2, \ldots, y_N]$$

4. VLSI Architecture for SVM

For hardware acceleration, dedicated architectures for the Support Vector Machine (SVM) and the Binary Relevance (BR) classifier were designed and implemented on the PL, enabling efficient, high-speed classification suitable for live deployment.

For a linear SVM, the equation used is $y = wx + b$. The proposed architecture is shown in Figure 3, and it consists of an adder and multipliers. Each multiplier internally consists of adders and shifters.

For the 2-class BRC, two linear SVMs are used as shown in Figure 4.

5. Implementation

To realize a real-time leaf identification system on the PYNQ-ZU platform, the image acquisition and feature extraction processes are carried out on the ARM Cortex processor on board within the Processing System (PS), while the classification tasks are offloaded to the Programmable Logic (PL) for accelerated execution.

5.1. SVP

SIFT Descriptors: Images were converted to grayscale, and keypoints and descriptors were extracted using the cv2.SIFT_create().detectAndCompute() function.

VLAD Encoding: K-means clustering was performed on the training descriptors after selecting the optimal number of clusters. Residuals between descriptors and their corresponding centers were calculated for each assigned cluster.

The VLAD vector was formed by aggregating and concatenating these residuals across clusters. To enhance performance, power normalization followed by L2 normalization was applied.PCA was then used to reduce dimensionality, retaining the most discriminative components in the final feature vector.

5.2. Classification

Single-label Classification: A linear SVM classifier with a soft margin and regularization parameter C = 1 was used. Multilabel Classification: The BR approach involving two classes was adopted for multilabel classification. Two independent linear SVMs were trained using SIFT VLAD features, each predicting the presence or absence of one class.

Weight and bias vector from the trained model are sent to the SVM module via AXI DMA and AXI GPIO, respectively. For every test image, the SVP feature vector is calculated and sent using AXI DMA, the class prediction is indicated using LEDs, and the read back is using AXI GPIO.

5.3. Performance Evaluation

The classification performance was assessed using precision, recall, accuracy and F1-score. For multilabel results, weighted average scores were reported to account for class imbalance.

5.3.1. Results of single label using SVM

The simulation results for SVM are shown in Figure.6 The performance metrics for two different datasets are compared for software and hardware implementation, and the results are shown in Tables 1 and 2.
The number of clusters for VLAD is taken as 8, and 8-dimensional PCA is used.

5.3.2. Results of Multilabel using BRC

The sample Multilabel image is shown in Figure 7, which is identified using BRC. The performance metrics are shown in Table 3 for standard Apple and Cherry, and for custom Custard and Mango in Table 4.

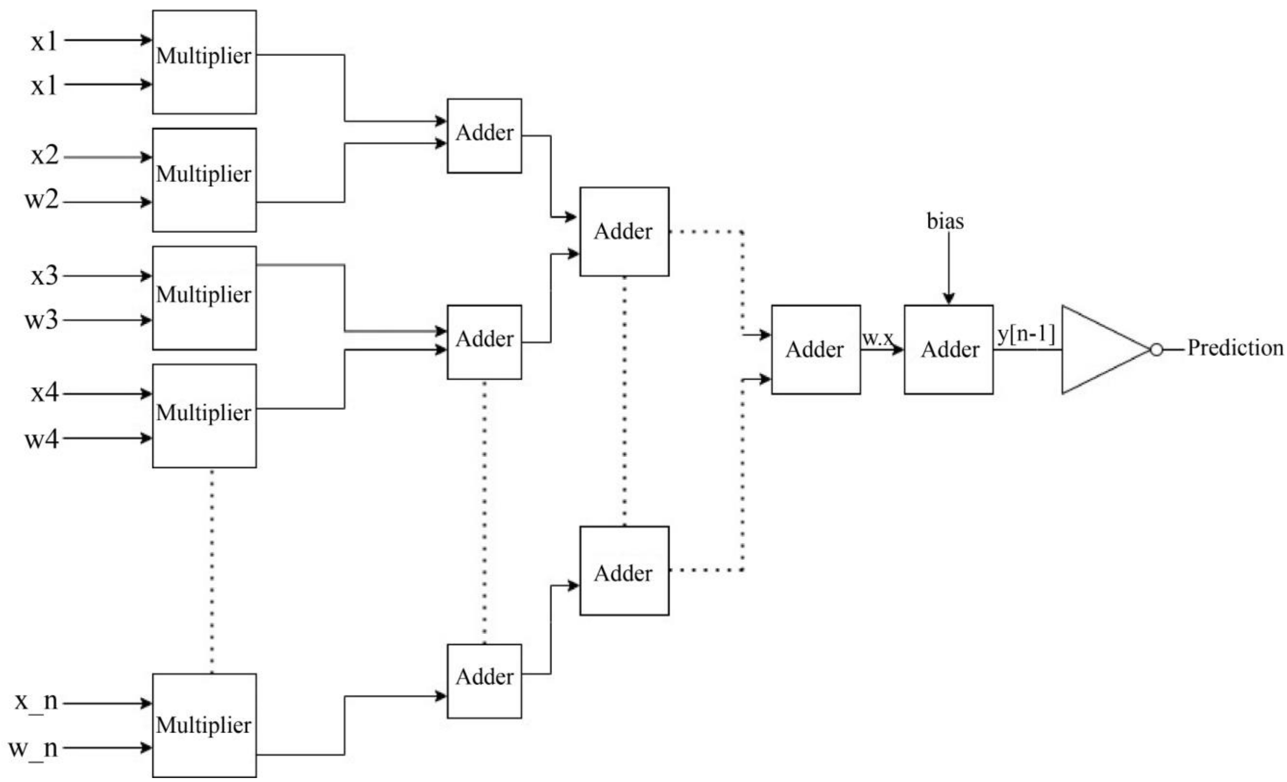

Fig. 3 Architecture of SVM

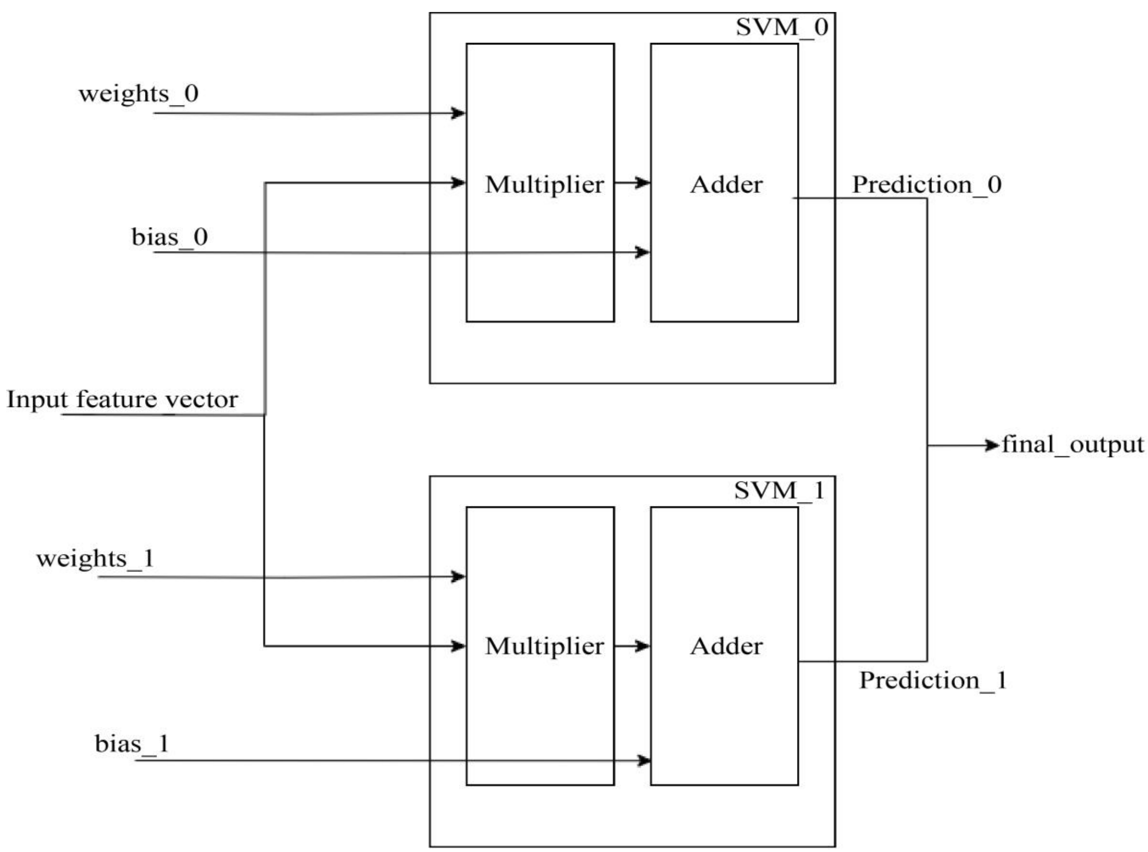

Fig. 4 Architecture of BRC

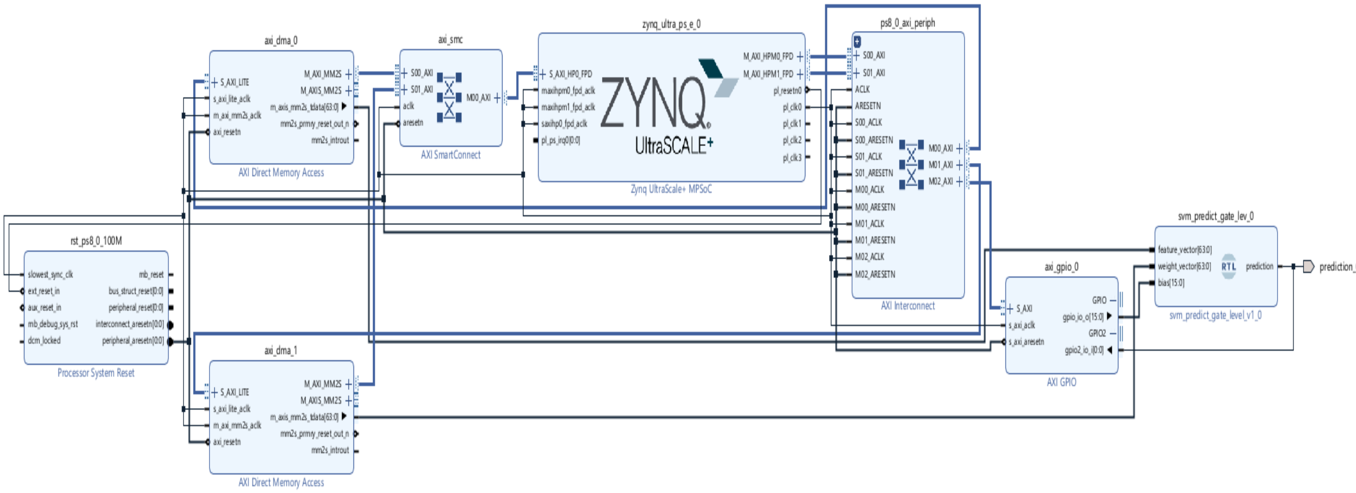

Fig. 5 Block design of proposed framework for PYNQZU board.

Table 1. SVM Classification on Standard Apple and Cherry Dataset

	Accuracy	Precision	Recall	F-1 Score
Python Software	0.90	0.92	0.90	0.90
Performance test on trained data (PYNQZU)	1.00	1.00	1.00	1.00
Test data similar to training data (PYNQZU)	0.70	0.68	0.72	0.70
Test data with deviation from trained data (PYNQZU)	0.65	0.63	0.70	0.66

Table 2. SVM Classification on Custom Custard and Mango Dataset

	Accuracy	Precision	Recall	F-1 Score
Python Software	0.70	0.70	0.70	0.69
Performance test from trained data(PYNQZU)	0.96	0.95	0.97	0.96
Test data similar to trained data (PYNQZU)	0.63	0.62	0.65	0.63
Test data with deviation from trained data (PYNQZU)	0.55	0.55	0.52	0.53

Table 3. BRC classification on Standard Apple and Cherry Dataset

	Accuracy	Precision	Recall	F-1 Score
Python software	0.82	0.82	0.82	0.82
PYNQZU	0.65	0.66	0.65	0.64

Table 4. BRC Classification on Custom Custard and Mango Dataset

	Accuracy	Precision	Recall	F-1 Score
Python software	0.71	0.71	0.71	0.71
PYNQZU	0.45	0.45	0.45	0.44

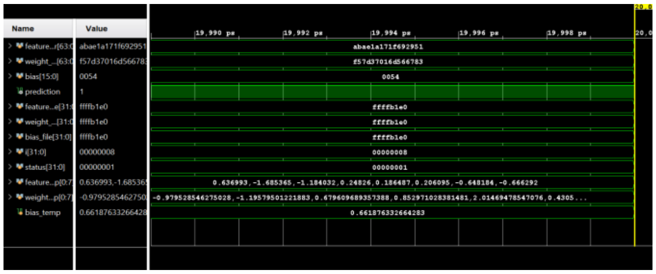

Fig. 6 SVM simulation results

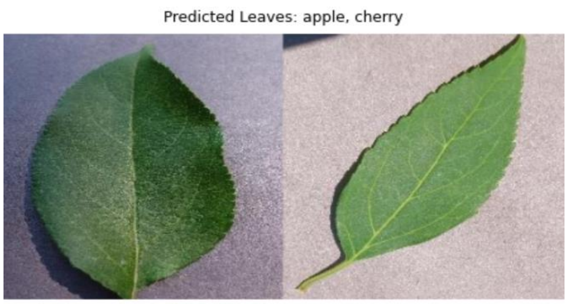

Fig. 7 Multilabel leaves images

6. Conclusion

The evaluation results indicate that the combination of SIFT, VLAD, and PCA provided the most consistent and robust performance across a range of imaging conditions. This highlights its effectiveness for real-world leaf classification tasks, particularly in environments with varying illumination, scale, and orientation. The proposed hardware results are significantly comparable to those obtained using Python software. Furthermore, the proposed technique demonstrates efficacy with multilabel data.

In testing the proposed technique on hardware, we utilized the standard Apple and Cherry dataset, which consisted of 376 training samples and 70 testing samples for performance evaluation. The results showed 100% identification accuracy, while similar data yielded a 70% identification rate and 65% accuracy despite illumination, scale, and orientation changes.

Additionally, the technique was tested on a custom dataset comprising Custard and Mango, with 352 training samples and 70 testing samples. This resulted in a 96% identification accuracy; similar data achieved a 63% identification rate and 55% accuracy in challenging scenarios involving dense backgrounds, clusters of leaves, and bushes.

The proposed BRC architecture has been adapted for multilabel leaf data and performed well, achieving an accuracy of 65% on standard data and 45% on custom data. Overall, the proposed architectures exhibit effective classification and identification of leaves.

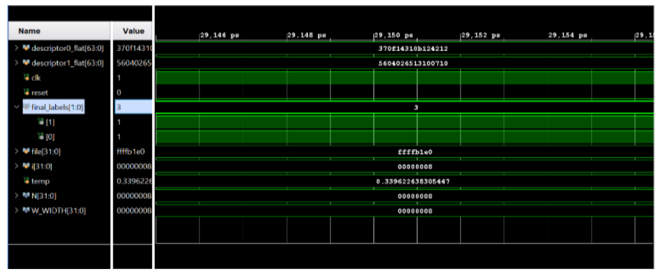

Fig. 8 BRC simulation Results

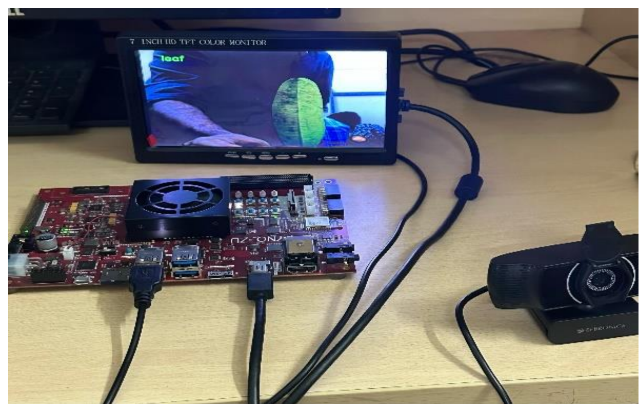

Fig. 9 Experimental setup

Acknowlegement

The Authors would like to express sincere gratitude to C2S program, Ministry of Electronics and Information Technology, Government of India, for providing required tools used during this work under the project titled "Development of SoC System with Vision-Based UAV and Remote Mobile Robotic Arm for Precision Agriculture" with CI Dr. T. Satya Savithri and Co-CI Dr. P. Chandra Sekhar Reddy, Department of Electronics and Communication Engineering, JNTUH CESTH.

References

[1] David G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004. [CrossRef] [Google Scholar] [Publisher Link]

[2] A. Ehsanirad, and S.K.Y.H, "Leaf Recognition for Plant Classification using GLCM and PCA Methods," *Oriental Journal of Computer Science & Technology*, vol. 3, no. 1, pp. 31-36, 2010. [Google Scholar]

[3] G.S. Sural, and S. Pramanik, "Segmentation and Histogram Generation using The HSV Color Space for Image Retrieval," *International Conference on Image Processing*, Rochester, Rochester, NY, USA, 2002. [CrossRef] [Google Scholar] [Publisher Link]

[4] Adil Salman et al., "Leaf Classification and Identification using Canny Edge Detector and SVM classifier," *2017 International Conference on Inventive Systems and Control (ICISC)*, Coimbatore, India, pp. 1-4, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[5] Moulay Hafid Aabidi et al., "A New Approach of Leaf Disease Detection Using Bag of Visual Words," *Journal of Theoretical and Applied Information Technology*, vol.101, no. 14, pp. 5645-5664, 2023. [Google Scholar] [Publisher Link]

[6] Murad Qasaimeh, and Assim Sagahyroon, and Tamer Shanableh, "FPGA-Based Parallel Hardware Architecture for Real-Time Image Classification," *IEEE Transactions on Computational Imaging*, vol. 1, no. 1, pp. 56-70, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[7] Qilong Li, and Xiaohong Wang, "Image Classification Based on SIFT and SVM," *IEEE/ACIS 17th International Conference on Computer and Information Science*, Singapore, pp. 762-765, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[8] Di Fu, Bo Zhou, and Jinglu Hu, "Improving SVM based Multi-Label Classification by Using Label Relationship," *International Joint Conference on Neural Networks*, Killarney, Ireland, pp. 1-6, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[9] Jesse Read et al., "Classifier Chains for Multi-Label Classification," *Machine Learning*, vol. 85, pp. 333-359, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[10] Marta Ruiz-Llata, Guillermo Guarnizo, and Mar Yébenes-Calvino, "FPGA Implementation of a Support Vector Machine for Classification and Regression," *International Joint Conference on Neural Networks*, Barcelona, Spain, pp. 1-5, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[11] Shereen Afifi, Hamid Gholam Hosseini, and Roopak Sinha, "FPGA Implementations of SVM Classifiers: A Review," *SN Computer* Science, vol. 1, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[12] C. Kyrkou and T. Theocharides, "A Parallel Hardware Architecture for Real-Time Object Detection with Support Vector Machines," *IEEE Transactions on Computers*, vol. 61, no. 6, pp. 831-842, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[13] Davood Mahmoodi et al., "FPGA Simulation of Linear and Nonlinear Support Vector Machine," *Journal of Software Engineering and Applications*, vol. 4, pp. 320-328, 2011. [CrossRef] [Google Scholar] [Publisher Link]