International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(1s) 2026 pp. 1-19
© Cerebration Science Publishing
https://doi.org/10.70917/ijcisim-2026-1981
Article

COMPARATIVE ANALYSIS OF FEATURE DESCRIPTORS FOR OBJECT RECOGNITION UNDER DIVERSE ENVIRONMENT

Lakshmi Sharma K. M.[1], Aradhana D.[2]

[1] Department of Computer Science and Engineering (CSE), Ballari Institute of Technology & Management, Ballari – 583104, Karnataka, India; Affiliated to Visvesvaraya Technological University (VTU), Belagavi – 590018, Karnataka, India.
lakshmisharmakm@gmail.com
ORCID: 0009-0003-0464-9587
[2] Department of Computer Science and Engineering (Data Science), Ballari Institute of Technology & Management, Ballari – 583104, Karnataka, India; Affiliated to Visvesvaraya Technological University (VTU), Belagavi – 590018, Karnataka, India.
aradhanabm@gmail.com
ORCID: 0000-0002-7292-5657

Abstract: Object detection is performed in several applications, especially in self-driving cars, where the exact detection of objects could reduce human error-related accidents to a large extent. Most traditional approaches to object detection involve the use of predefined features, and their practical use can be limited by low adaptability and robustness in various conditions. On the other hand, deep learning (DL) methods collect a considerable number of datasets to establish their personally productive, broad feature sets for deeper detection performance. The research presents a rigorous comparative analysis of feature descriptors for the object detection task across different environments, examining the domain of both traditional and DL based methods. Thus, the paper conducts an assessment of numerous feature descriptors, such as Scale-Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), Oriented FAST and Rotated BRIEF (ORB), as well as current DL models like Convolutional Neural Networks (CNN), YOLO, and Vision Transformers (ViTs). Utilizing a Support Vector Machine (SVM) for classification, the study rigorously evaluates the performance of each method employing the Pascal VOC 2007 dataset. The experimental analysis of the proposed model is conducted utilizing a Python tool. The results demonstrate that the ViTs + SVM combination significantly outperforms other models, achieving an impressive accuracy of 95.04%. This finding underscores the advantages of deep learning in effectively capturing complex patterns and enhancing the robustness of object detection systems.
Keywords: Object detection, Feature descriptors, handcrafted features, Convolutional Neural Networks, You Only Look Once, Vision Transformers, and Support Vector Machine.

1. Introduction

Object detection plays a major role in the computer vision because it allows machines to perceive and identify visual information to some degree, which is the simplest precondition to make decisions. This feature does not only have a significant contribution to many achievements and applications, but is also gaining popularity, as observed in intelligent surveillance, industrial automation, medical diagnostics, artificial reality, and, specifically, autonomous and assisted driving systems [1]-[2]. The urgency of the hour with regards to accuracy in object detection (OD) has been critically defined by the U.S. National Highway Traffic Safety Administration (NHTSA) that has revealed alarming statistics that over 88% of road accidents are human induced [3]. These are mainly caused by poor reaction times, distractions, and inappropriate choices are the key elements that cause such occurrences.

Copyright: © 2026 by the authors

Object detection is the process of identifying the instance of objects with various object classes and their spatial positions in an image [4]. Conventional object detection systems also require humans to be involved in extracting features [5]. A typical list of detection aspects comprises of the SIFT [6], the Histogram of Oriented Gradients (HOG) [7], and task-specific representations. These handcrafted features are beneficial in the detection task, but they have to be built on prior domain knowledge, which makes their construction costly and restricts their applicability to different environments besides decreasing their robustness.

In order to address the drawbacks of the conventional approach, over the years several researchers have come up with a variety of feature descriptors that are based on the dissimilar techniques to describe the image properties [8], such as SIFT [9], SURF [10], and ORB [11]. SIFT is scale, rotation, and affine invariant, whereas SURF has better performance rates and computation speed. ORB offers real-time capability since it offers speed advantage with minimum memory consumption relative to SIFT [12]. These features are beneficial for certain applications but struggle when confronted with conditions such as rapid lighting changes or blurry motions, along with background clutter [13]. Changeable settings decrease the reliability of detection systems because they make the combination of extracted features and matched results less precise during mobile operations that require quick responses.

Additionally, the deep learning-based approaches address the problems of manual feature design by automatically learning effective feature representations from large-scale datasets [14]. This allows OD to be separated into anchor-based as well as non-anchor-based approaches according to whether an anchor is present [15]. Two-stage detectors, like the R-CNN family of techniques, and one-stage detectors, such as YOLOv2 (You Only Look Once v2) as well as SSD (Single Shot MultiBox Detector), are examples of anchor-based techniques [16]. Following the identification of potential regions using the two-stage R-CNN approach, the regions are subjected to a CNN for feature extraction, classification, and localization [17]. In contrast to R-CNN, which employs a two-stage feature design, fast R-CNN takes a step to improve performance by adopting a Region Proposal Network (RPN) as well as enhancing the depth of the networks that comprise the detection pipeline [18]. The first persistence and integration of the RPN, along with the depth of the networks, increase the computational efficiency of the one-stage detection. This is achieved through a single forward pass, which enables the localization and classification of object categories, as well as bounding box regression, all within a single pass [19].

Non-anchor detection methods utilize bounding box locations directly, as they do not require predetermined anchor boxes. Non-anchor methods used in object detection include YOLOv1, alongside CornerNet [20] and ExtremeNet [21], as well as the Fully Convolutional One-Stage Object Detector (FCOS). The YOLOv1 approach treats detection as a grid-based regression system in input space. CornerNet realizes detection by identifying corner keypoint coordinates, which express the top-left as well as bottom-right corners of bounding boxes. ExtremeNet describes the boundaries of an object using extreme keypoints and groups these keypoints based on geometric cues. FCOS serves as a detection algorithm that makes object location and category predictions at the pixel level without requiring anchor boxes for improved feature selection.

Therefore, an accurate evaluation of handcrafted and deep learning-based feature descriptors is urgently required to assess their effectiveness under various scenarios. The study introduced a comparative analysis of feature descriptors for OD in diverse environments. This research conducts a comprehensive evaluation of features to choose optimal descriptor methods that enhance algorithm robustness for developing secure automated systems. The major objective of the proposed technique is described as follows:

- To conduct a comprehensive comparative analysis of Scale-Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), Oriented FAST, Rotated BRIEF (ORB), and DL-based feature descriptors for object detection.
- To enhance data quality, a pre-processing step is performed, which includes various filtering techniques.
- To extract and evaluate both handcrafted and deep features to assess their impact on detection accuracy, robustness, and computational efficiency.
- To utilize a Support Vector Machine (SVM) for object classification based on the extracted features.

The remaining content of the research is structured as follows: Section 2 presents a survey of current methods. Section 3, the suggested methodology's process and workflow are briefly described. The results analysis and comparison are evaluated in Section 4. Finally, the paper's overall conclusion is presented in Section 5.

2. Related works

A survey of current techniques for feature-based object detection was analyzed and described as follows.

Zhou et al. [22] developed an accident recognition technique based on spatio-temporal feature encoding and a multilayer neural network. This network architecture allows the spatial-temporal video features to be encoded and then frames to be clustered, which yields efficient and effective accident detection capabilities from driving videos. The developed methodology demonstrates both excellent traffic accident detection capabilities and real-time readiness for VANET environments, as confirmed by an exhaustive experimental analysis. However, the developed method faced a significant limitation due to a high degree of overfitting.

Mahaur et al. [23] developed a few architectural updates on the popular YOLOv5 to improve its capability to detect small objects without degrading its ability to detect large objects, and with a particular reference to autonomous driving. Although the updates to the model increased the computational load, they significantly improved both detection accuracy and speed. The method developed (iS-YOLOv5 model) yields a 3.35% increase in mean Average Precision (mAP) on the BDD100K dataset compared to the standard YOLOv5. However, the limitation of the developed method is computational complexity.

Singh et al. [24] presented an OD and recognition algorithm for indoor service robots by making adjustments to the automatically learning YOLO framework. The model's computer-vision-based algorithm is associated with similar conventional OD as well as recognition algorithms on aspects such as mAP score, mean inference period, weight extent, as well as false positive percentage, among others. Regarding object recognition, the model was able to decrease computational complexity and model weights while maintaining high detection accuracy. However, this method was limited to indoor use and lacked adaptability to dynamic environments.

Jiang et al. [25] developed a multi-task semantic segmentation model by enhancing the Faster R-CNN algorithm. This model not only performs indoor scene segmentation but also performs object classification and detection using RGBD imaging. The improvements in model training due to optimal RGB and depth image fusion, as well as the non-maximum suppression, which was updated for multi-scale detection, were the main entities. Additionally, the model's performance was compromised under extreme lighting variations and high-speed object motion, which in turn restricted its real-time usability.

Zhao et al. [26] improved the Fire-YOLO deep learning algorithm to detect fires under different natural lighting conditions. The Fire-YOLO detection model expands features to three dimensions, which enhances its propagation capabilities of features to detect fire, as well as improving performance and reducing parameters. The Fire-YOLO recognition model is capable of analyzing the objects that are similar to fire and smoke and small targets of fire. The technology can identify forest fires in real-time with images of 416 pixels x 416 pixels and an average of 0.04 seconds per frame. The system experienced challenges in identifying the presence of hidden fires and the good ability to distinguish between the fire and non-fire objects which undermined the reliability of operations when complex scenarios were to be analyzed.

The small object detection framework created by Shao et al. [27] was object-specific to the maritime industry and was, therefore, about to resolve issues such as the background clutter and scale variation. The flow has shown to be quite efficient with an oceanic surveillance application. The implemented approach, however, is challenged by the fact that it fails to be effective in non-maritime situations.

Hussain et al. [28] created a hybrid object recognition scheme and implemented both traditional and deep learning techniques. Such a combination of the techniques was an attempt to enhance the quality of recognition and its power. Nonetheless, the created method is based on the features developed by hands, and this is why it is not adaptable in the complicated environment.

The object detection procedure had been limited with some restrictions on the basis of the performance of this method [22] to [28]. This requires a good object detector. This paper provides a comparative study of the feature descriptors of the object detection models in various settings.

3. Proposed methodology

This study presents a comparative analysis of feature descriptors based on object detection models in various environments, as depicted in Figure 1.

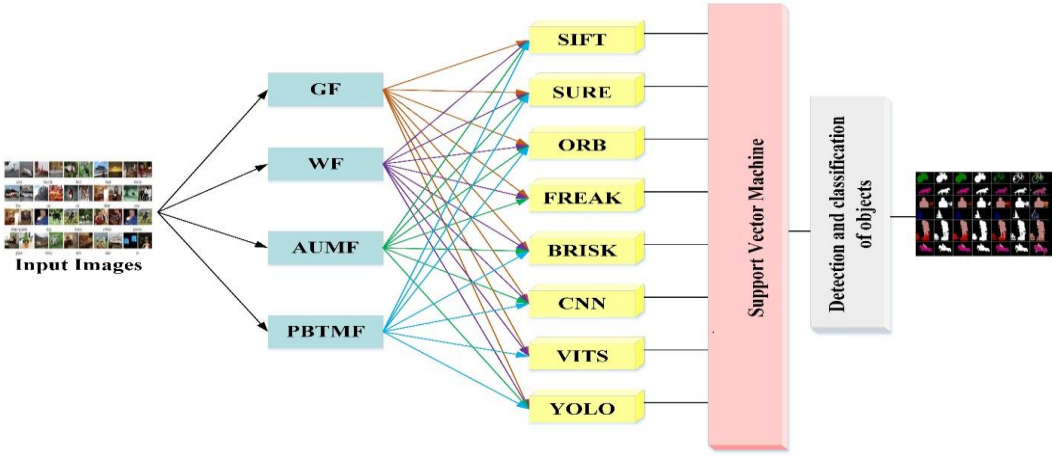


Figure 1: Block diagram of an object recognition system

Initially, the input image was collected from the publicly available Pascal VOC 2007 dataset. Then, to improve image quality and reduce unwanted effects, various filtering methods are employed, including the Gaussian filter (GF), Wiener filter (WF), adaptive unsharp mask filters (AUMF), and pixel density-based trimmed median filter (PBTMF). Then, the pre-processed image is fed into the feature extraction method. The feature extraction method can be categorized into handcrafted and DL based feature descriptors. Handcrafted features were extracted using Scale-Invariant Feature Transform (SIFT), Speeded-Up Robust Features (SURF), Oriented FAST and Rotated BRIEF (ORB), Fast Retina Key-point (FREAK), and Binary Robust Invariant Scalable Key Point (BRISK) technique. Deep learning methods (CNNs, YOLO, ViTs) learn features automatically from data, providing superior accuracy over handcrafted descriptors. Then the extracted features are fed into the SVM, which classifies the object in an input image. The method mentioned above is described in the following sections.

3.1 Filtering Methods

To increase the accuracy, a filtering technique is required. It reduces the unwanted noise in the input image. Finally, it enhances the equality of the input image. The different filtering methods employed in this study were a GF, a WF, an adaptive unsharp mask filter (AUMF), and a pixel density-based trimmed median filter (PBTMF).

3.1.1 Gaussian filter

The GF is utilized to increase the efficacy of picture smoothing [29]. The GF initial stage involves detecting noise, which is ineffective at eliminating salt as well as pepper noise. The analysis was achieved employing the Gaussian distribution. The Gaussian distribution's Probability Density Function $(G(a))$ is designated in the equation below.

$$G(a) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-(a-\mu)^2} \Big/ \left(2\sigma^2\right) \tag{1}$$

here, the gray level image is signified as a ,mean value signified as a and standard deviation is denoted as σ . The quantity of smoothing is determined by the standard deviation (σ). The GF output is then fed into the WF.

3.1.2 Wiener filter

The WF [30] is a linear filter that is intended to decrease the mean square error among the unique as well as filtered signals. It is intended to improve photographs that have been damaged by added noise. The WF adjusts its restrictions to achieve a compromise among noise reduction as well as image detail retention. The WF equation is shown by the following equations:

$$K(x, y) = G(x, y)A(x, y) \tag{2}$$

$$G(x,y) = \frac{S^*(x,y)F_m(x,y)}{\left|S^*(x,y)\right|^2 F_m(x,y) + F_n(x,y)} \tag{3}$$

$$G(x,y) = \frac{S^*(x,y)}{\left|S^*(x,y)\right|^2 \frac{F_n(x,y)}{F_m(x,y)}} \tag{4}$$

here, $F_m(x,y)$ is denoted as the power spectrum of the image process $F_n(x,y)$ is denoted as the spectrum equation (4) that can be attained by separating through F_m in equation 3.

3.1.3 AUMF

The AUMF [31] is employed to eliminate the disturbance from the input image. The AUMF preserves the edge and intensity while preventing the usage of additional procedures. Furthermore, an adaptive gain factor (AGF) is employed to modify the repetition process through image improvement. Finally, the denoised image can be mathematically stated as,

$$P = O + \gamma_{AG} X \tag{5}$$

here, O signifies the original image, γ_{AG} denotes the AGF, and X employs the edge-preserved image. The AGF γ_{AG} highly depends upon the common gain factor γ_G, colour enhancements γ_C and edge-preserved image γ_E. Generally, the AGF series γ_{AG} from a minimum rate of 0 to a maximum of 2.

3.1.4 PDBTMF

A PDBTMF [32] proposes tasks for each pixel at two levels. In the first step, this filter determines if the test pixel is deteriorated by salt and pepper noise (SPN). This filter determines whether a detected corrupted pixel is loud or not. It does this by examining the selected pixels mask. In particular, when the pixel rate of the test pixel equals 255 and the maximum value within the selected 3x3 window includes other pixels with values of 255, then the flutter assumes that the present pixel (255) is noisy.

3.2 Feature extraction technique

An object can be recognized by its color, texture, blob, shape, or any other characteristic. The performance of an OD system is primarily dependent on the significant characteristics derived from the image database. The experiment included eight feature extraction algorithms: SIFT, ORB, SURF, FREAK, BRISK, CNN, YoLo, as well as Vision Transformer (VisT).

3.2.1 Scale Invariant Feature Transform (SIFT)

The SIFT [33] is an image processing technique established by David Lowe for recognizing and characterizing local features in images. Though it is robust against affine distortion, size, rotation, and lighting variations, it works effectively in object detection, image stitching, and 3D reconstruction. The procedure begins with the construction of a scale-space through the application of Gaussian blurring to the input image, surveyed by the recognition of keypoints based on the Difference of Gaussians (DoG) method. These keypoints are candidate interest points that are invariant to a range of image scales. Once detected, keypoint localization is performed to refine the positions by removing low-contrast and edge responses. Every keypoint receives an assigned orientation through the analysis of local image gradient directions, thereby achieving rotation independence. The gradient magnitude $m(i,j)$ as well as orientation $\phi(i,j)$ are calculated as:

$$m(i,j)=\sqrt{\left(G(i+1,j)-G(i-1,j)\right)^2+\left(G(i,j+1)-G(i,j-1)\right)^2} \tag{6}$$

$$\phi(i,j)=\arctan\left(\frac{G(i,j+1)-G(i,j-1)}{G(i+1,j)-G(i-1,j)}\right) \tag{7}$$

here, $G(i,j)$ is the strength of the Gaussian-blurred image at point (i,j). Finally, a keypoint descriptor is generated by considering a region around the keypoint, which is separated into 4×4 sub-regions. Every sub-region produces an 8-bin orientation histogram, subsequent by a 128-dimensional descriptor vector. This vector effectively captures local gradient distributions, offering high distinctiveness and invariance.

3.2.2 Speeded-Up-Robust Features (SURF)

SURF [34] is a fast as well as robust algorithm for scale- and rotation-invariant keypoint recognition as well as description. The SURF considers features using the Hessian matrix (HM), which is obtained from second-order Gaussian results of image intensity information. The matrix is defined as:

$$HIP=\begin{bmatrix}L_{xx} & L_{xy}\\ L_{xy} & L_{yy}\end{bmatrix} \tag{8}$$

where, L_{xx}, L_{xy}, L_{yy} represent second-order partial results of the image after applying Gaussian smoothing. To determine the strength of each detected feature, SURF calculates the basis of the approximated HM as:

$$Det\left(HIP_{approx}\right)=D_{xx}D_{yy}-\left(wD_{xy}\right)^2 \tag{9}$$

here, D_{xx}, D_{yy}, D_{xy} are the approximated second-order derivatives employing box filters, and w is a weight factor that balances the response. Keypoints are extracted using a $3\times3\times3$ non-maximal dominance on a scale-space pyramid, and their location is refined via interpolation. Each keypoint is then allocated an orientation based on Haar wavelet answers in the x and y orders within a circular neighborhood. These responses help compute the trace of the HM as:

$$T=L_{xx}+L_{yy} \tag{10}$$

This trace is used for selecting keypoints with strong intensity changes. The SURF descriptors are obtained by calculating wavelet responses against a square region that aligns with the current prevailing orientation. Image distortions and scaling, along with rotation, do not influence the SURF descriptors. The fast operation and high stability of SURF make it the ideal tool for time-sensitive identification systems such as facial and object detection.

3.2.3 Oriented Fast and Rotated BRIEF (ORB)

ORB [35] is a fast as well as reliable feature extraction algorithm. ORB is significantly faster than conventional approaches, such as SIFT and SURF, and therefore it can be utilized in real-time applications. It is a fusion of the FAST (Feature form Accelerated Segment Test) keypoint descriptor as well as the BRIEF (Binary Robust Independent Elementary Features) descriptor, with some adjustments that improve scale and rotational invariance.

ORB detects the keypoints employing the FAST detector, and then it refines them with the Harris corner measure. Finally, it assigns orientation to a keypoint using the intensity centroid method. The resulting descriptors are binary and highly efficient to compute and match.

The mass center of the patch is calculated using the image moments:

$$X=\frac{e_{10}}{e_{100}},\qquad Y=\frac{e_{01}}{e_{00}} \tag{11}$$

The orientation θ of the keypoint is then computed as:

$$\theta = \tan^{-1}\left(\frac{e_{01}}{e_{10}}\right) \tag{12}$$

here, t_{mn} denotes the image moments, and u, and v represent the pixel intensities at position j and k, respectively.

3.2.4 Fast Retina Key-Point

A novel binary descriptor called Fast Retina Key-point (FREAK) [36] was developed, stimulated by the retina and the human visual scheme. It employs a circular sampling pattern known as the retinal sampling grid, which concentrates more sampling points near the center region. From this grid, 512 selected pairs are used to generate binary features, while the rest are discarded.

Each binary descriptor is formed by relating the strength values of two facts in a pair $P_a = (p_a^1, p_a^2)$. The comparison is expressed by the following thresholding function:

$$T(P_a) = \begin{cases} 1 & if\left(I\left(p_a^1\right) - I\left(p_a^2\right)\right) > 0 \\ 0 & otherwise \end{cases} \tag{13}$$

where $I\left(p_a^1\right)$ and $I\left(p_a^2\right)$ are the smoothed intensity standards at the selected point locations. The final FREAK descriptor is computed by combining these binary outcomes across all pairs:

$$F = \sum_{a=0}^{N-1} 2^a T(P_a) \tag{14}$$

here, N denotes the number of point pairs. The descriptor is variation invariant and resistant because of the summation of local gradients over the chosen point pairs. FREAK is computationally light and works reliably in keypoint description and matching operations, hence it can be utilized in real-time vision applications.

3.2.5 Binary Robust Invariant Scalable Key Point (BRISK)

BRISK [37] is a keypoint indicator and descriptor that selects important points in an image employing a saliency condition. A circular sampling design is then functional around each keypoint to collect intensity standards, which are employed to determine the keypoint's orientation.

To find the orientation, BRISK uses long pairs of sample points and calculates the gradient between them using:

$$g(p_i, p_j) = \frac{p_i - p_j}{\left\|p_i - p_j\right\|} \times \frac{I\left(p_i, \sigma_i\right) - I\left(p_j, \sigma_j\right)}{\left\|p_i - p_j\right\|} \tag{15}$$

where p_i and p_j are the coordinates of sample points in each pair, and $I\left(p_i, \sigma_i\right)$ is the intensity of an image flattened by a Gaussian kernel with a alteration of σ_i in position p_i. After calculating the gradient for every pair, the summation of all the gradients of extensive couples is computed in horizontal and vertical directions together, as shown in the following:

$$G = \begin{pmatrix} g(x) \\ g(y) \end{pmatrix} = \frac{1}{L} \sum_{p_i, p_j \in patch} g(p_i, p_j) \tag{16}$$

Finally, the patch positioning is determined by calculating the arctangent of G as shown in the following:

$$\theta = \arctan\left(\frac{g(y)}{g(x)}\right) \tag{17}$$

where θ is the positioning of the patch. Every patch has a key location, representing its dominant direction.

3.2.6 Convolutional neural network (CNN)

In DL-CNNs [38], the first-order feature extractor is the convolutional layer, which generates 2D feature maps by convolving the input image with filters of various extents. Each learns spatial hierarchies of features (corners, edges, etc.) in training as:

$$Y_{i,j} = (X * W)_{i,j} + b \tag{18}$$

where X is designated as the input, W is indicated as the filter, as well as b is signified as the bias term. The pooling layer, mostly max pooling, decreases spatial dimensions by choosing the maximum value in 2×2 non-overlapping regions, effectively downsampling feature maps by 4 and minimizing overfitting and computational cost. In average pooling, the mean is used instead. The batch normalization layer (NL) regularizes inputs to a layer by:

$$\hat{x} = \frac{x - \mu}{\sqrt{\sigma^2 + \varepsilon}} \tag{19}$$

here, x is denoted as the input value, μ is denoted as the mean of the batch, σ^2 is meant as the variance of the batch, $\hat{x}$ is indicated as the normalized output, and $\in$ is designated as the small constant added for numerical stability. Giving faster convergence and preventing vanishing gradients. The dropout layer drops neurons randomly at training time, controlled by a dropout rate p, to prevent overfitting. Finally, the fully connected layer outputs the classification.

3.2.7 Vision Transformer

The original ViT is presented in [39]. The pre-processed images $R \in H^{X \times Y \times Z}$ is separated into fixed-size patches to be converted into a consecutive representation of compressed 2D patches $R_c \in H^{P \times (c^2 Z)}$, where X signifies the image height, Y signifies the image width, Z is the amount of channels, and (C, C) denotes the resolution of every image patch. The amount of patches P can be intended as

$$P = \frac{X \times Y}{c^2} \tag{20}$$

Before feeding the sequence of patches into a transformer, a linear projection (LP) is functional to the patches. During this LP, the patches are mapped to a T dimensional vector space by increasing them with an embedding matrix M. The output of this LP is stated to as patch embedding. To permit the model to seize positional data within the image, positional embedding M_{pos} are added to the patch embedding. Additionally, the embedded image patches are concatenated with a learnable class token R_{class}, which is needed for the classification process. The early patch embedding s_0, containing the order of image patches beside the class token, is intended as follows:

$$s_0 = \left[R_{class}; R_c^1 M; R_c^2 M; .....; R_c^N M\right] + M_{pos}, \quad M \in H^{(c^2 Z) \times T}, M_{pos} \in H^{(N+1) \times T} \tag{21}$$

here, R_c^n represents the n^{th} image patch, where $n \in 1,2,.....,N$. The subsequent embedded image patches are then accepted to the transformer encoder. The transformer encoder is employed by totalling a multihead self attention (MSA) and a multilayer perceptron (MLP) block after an NL. A residual connection is comprised among each NL input as well as the output of the MSA and MLP blocks, as it is correspondingly designated by (22) and (23):

$$Z_l' = MSA(LN(Z_{l-1})) + Z_{l-1}, \quad l = 1.....L \tag{22}$$

$$Z_l = MLP(LN(Z_l^{'})) + Z_l^{'}, \quad l = 1.....L \tag{23}$$

where 1 is the layer ViT index as well as LN a normalization layer. More details about the MSA layer are described in [39]. Finally, $\frac{1}{z_0}$ is the yield of the ViT is normalized according to equation (23) and altered into an ultimate decision class label by an MLP block.

$$y = LN(z_L^0) \tag{24}$$

where z_L^0 is the identification decision and the output of the ViT in the preceding layer. This decision head is employed by an MLP for ViT pre-training as well as a linear layer for fine-tuning. In the hybrid ViT version, rather than separating the input image into numerous patches that pass through the transformer, the output of a CNN (such as a ResNet50) serves as a feature extractor, feeding the ViT.

3.2.8 YoLo

The face tracking method is used to provide efficient face detection for temperature evaluation. In this research, the Yolov5-Face model is used to track faces based on their facial features, and it is also fine-tuned for thermal images. A highly efficient solution for applications such as face tracking in thermographic images, YOLOv5-Face is a continuation of the YOLO-OD architecture, designed for detecting and tracking faces in real-time [40]. It can track several faces simultaneously in an image sequence. The detection of each face with bounding boxes around the face is described in the following equation.

$$D = \left(\hat{a}, \hat{b}, \hat{w}, \hat{h}, \hat{d}_{Face}\right) \tag{25}$$

here, the center coordinate of the box is denoted as (a, b), the width as well as height of the image are signified as W, and h. The predicted probability of detection of a face is denoted as $\hat{d}_{Face}$

3.3 Classification using SVM

The retrieved structures are fed into the SVM classification model [41]. The classifier, trained employing data-handcrafted features and deep features, is utilized to identify the various items. The SVM data only distinguishes two classes. The input restrictions for SVM are as follows:

i. Kernel: Linear

ii. Gamma: 0.0001

iii. Test Size: 0.2 (20%)

SVM is primarily a classification approach, although it may also be employed to solve regression as well as classification difficulties. SVM simplifies the processing of many categorical as well as continuous variables. Figure 2 shows the SVM model.

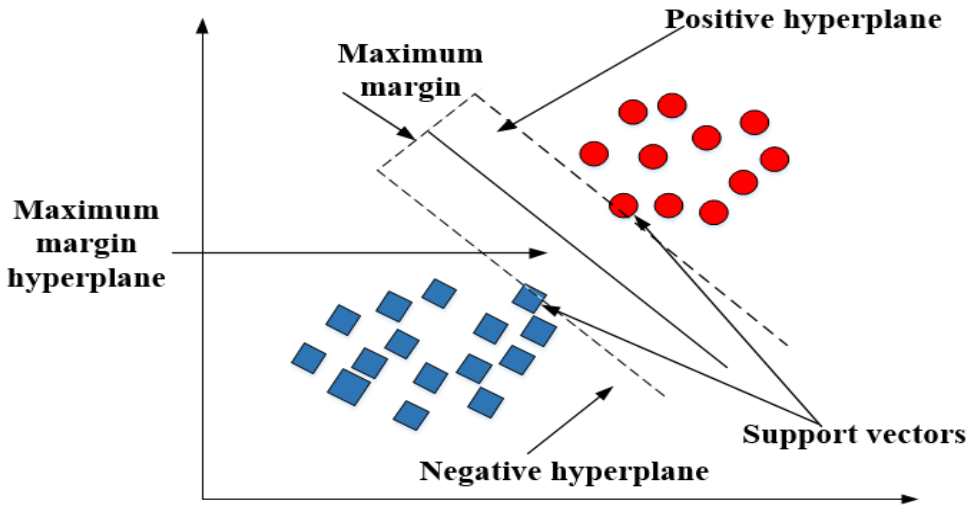


Figure 2: SVM model

The SVM model is used for crack detection by classifying the structures and the obtained value of the membership role to stratify the hyperplane $(H(P))$ or not determined by optimal hyperplanes.

Satisfaction Criteria: $H(P) > 1$ (26)

The optimal separating hyperplanes $H_j(p) = 0$ determined by defining a one-dimensional membership function $f_{ij}(p,q)$ as follows.

(i) The values of the diagonals are equal $(D = j)$

$$f_{ij}(p) = \begin{cases} 1 & for \quad H_i(P) > 1 \\ H_i(P) & for \quad H_i(P) < 1 \end{cases} \tag{27}$$

These rules determine the correct class as 1, 2, or 3

(ii) The values of the diagonal are not equal $(i \neq j)$

$$f_{ij}(p) = \begin{cases} 1 & for \quad H_i(P) < 1 \\ -H_i(p) & for \quad H_i(P) > 1 \end{cases} \tag{28}$$

These rules determine the correct class as 1, 2, or 3.

Some of the procedures are given below for detection.

(i) If the descriptor value p is $H_i(p) > 0$, then it would satisfy only that class.

(ii) If $H_i(p) > 0$, when p placed in several classes, the data are categorized into classes using maximum $H_i(p)$.

If $H_i(p) \leq 0$, when p placed in several classes, using the minimum $H_i(p)$. Finally, the SVM model is employed to accurately categorize the different objects.

4. Results and Discussion

This section undertakes a thorough experimental analysis and comparison of various feature descriptor methodologies to evaluate their performance on the Pascal VOC 2007 testing dataset. The research evaluates various feature extraction approaches by combining ViTS + SVM, YOLO + SVM, CNN + SVM, SURF + SVM, BRISK +

SVM, ORB + SVM, SIFT + SVM, and FREAK + SVM. Evaluation methods based on various performance metrics measure the success rates of detection and classification functionality for each framework. The comparative analysis examines the performance characteristics and shortcomings of various feature descriptors in conjunction with SVM classifiers to identify their practical applications in visual recognition systems. Table 1 presents the system configuration for the suggested approach.

Table 1: System configuration

Processor	Intel(R) Core(TM) i7-4770 CPU@ 3.40 GHz 3.40 GHz
Installed RAM	1.60 GB (15.9 GB usable)
System type	64-bit operating system, x64-based processor
Pen and touch	No pen and touch input is available for this display
Tool	Python

4.1 Dataset Description

The Pascal VOC 2007 dataset (https://www.kaggle.com/datasets/zaraks/pascal-voc-2007) is a popular benchmark for testing OD as well as classification algorithms. It has a total of 9,963 images, divided into 5,011 (80%) for training and 4,952 (20%) for testing. Every image is labeled to show the location of item labels and bounding boxes. The dataset comprises 20 object classes, which are categorized into four classes: Animal, Indoor, Person, and Vehicle. The images represent a varied series of object classes, making them ideal for evaluating the model's robustness in real-world circumstances. Each image from the Pascal VOC 2007 dataset contains numerous objects. Figure 3 exhibits a variety of image samples from the Pascal VOC 2007 dataset.

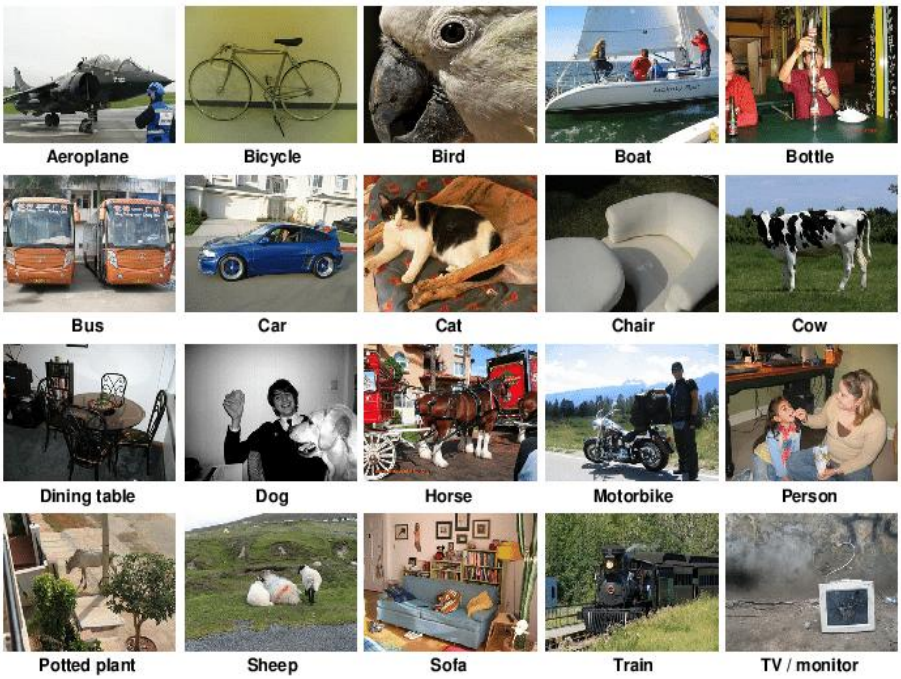


Figure 3: Sample images of the Pascal VOC 2007 dataset

4.2 Performance metrics and their formulation

This section outlines the several performance metrics utilized to evaluate the efficacy of the comparative model. The metrics contain accuracy, precision, recall, F1-score, TPR, and FPR. These metrics are detailed below:

$$Accuracy = \frac{TN + TP}{FP + TN + TP + FN} \tag{29}$$

$$\Pr ecision = \frac{TP}{TP + FP} \tag{30}$$

$$\mathrm{Re}\, call = \frac{TP}{TP + FN} \tag{31}$$

$$F1_{score} = 2 \times \left(\frac{\Pr ecision \times \mathrm{Re}\, call}{\Pr ecision + \mathrm{Re}\, call} \right) \tag{32}$$

$$FPR = \frac{FP}{FP + TN} \tag{33}$$

$$FNR = \frac{FN}{TP + FN} \tag{34}$$

here, TP - true positive, TN -true negative, FP - false positive, FN - false negative.

4.3 *Performance analysis*

This section provides a complete analysis of the outcomes attained from the different feature descriptors with the SVM classifier. The model's effectiveness was validated using the Pascal VOC 2007 dataset. Figures 4(a)-(d) visually depict the performance of the different feature descriptors with the SVM classifier.

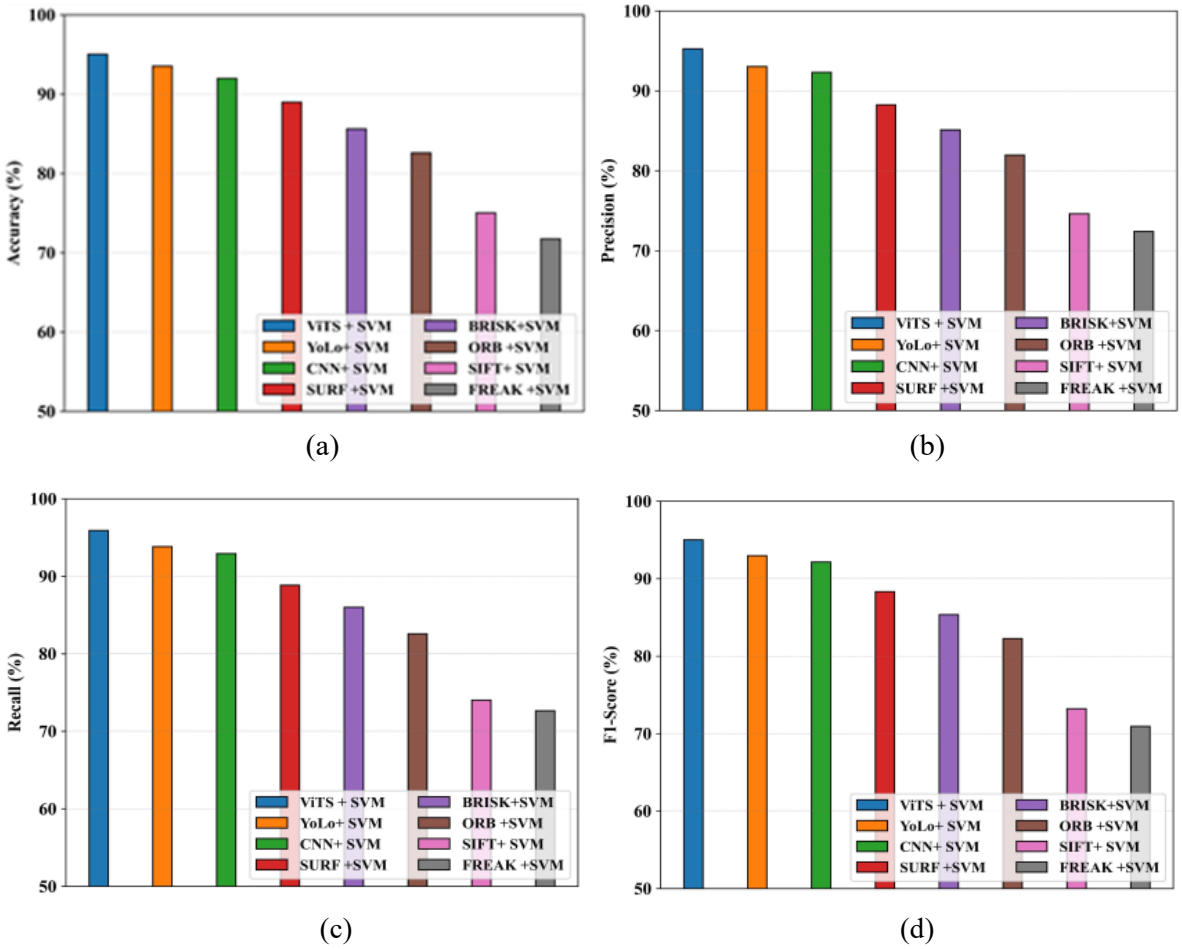


Figure 4(a-d): Analysis of the proposed and other models' performance

Figures 4(a-d) display a highly informative comparison of several feature descriptors with an SVM classifier. Among the numerous combinations, the ViTs + SVM model did exceedingly well, being the most successful. The ViTs + SVM model has a high overall classification accuracy of 95.04%, precision of 95.24%, F1-score of 95.01%, and recall of 95.92%. Therefore, it's quite evident that this model is competent and has a precise nature that allows it to identify and categorize the features of the image data effectively for classification. On the other hand, YoLo+ SVM, CNN+SVM, SURF+SVM, BRISK+SVM, ORB+SVM, SIFT+SVM, and FREAK+SVM have demonstrated lower performance due to several limitations. One of the limitations is the restricted feature learning capability. Many of these common descriptors are either human-made features or tend to one of the most basic image characteristics, namely, texture, edges, and shape, but they don't extract more complex patterns from the data. However, the ViTs + SVM model addresses this problem by utilizing a deep learning architecture to automatically identify patterns in the input images. It is capable of extracting both local and contextual information, as well as global information, and semantic or conceptual representations that surpass those of traditional methods. A greater variety of features, ranging from subtle differences to profound understanding, is learned by the model, which leads to a more discriminative and robust representation that results in improved classification performance. Figure 5 presents a comparison of the precision–recall curve between the in-depth comparative analyses of multiple feature descriptors integrated with an SVM classifier.

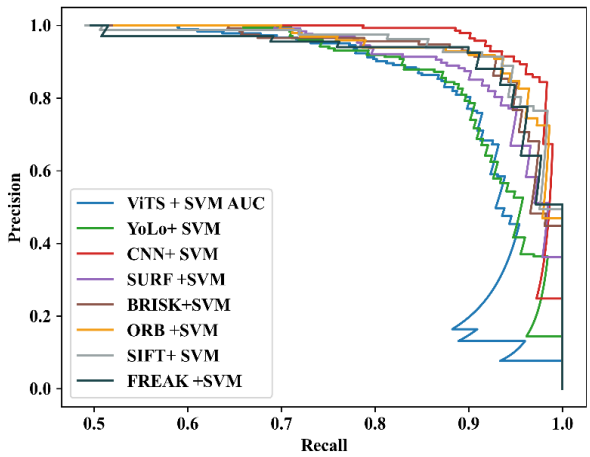
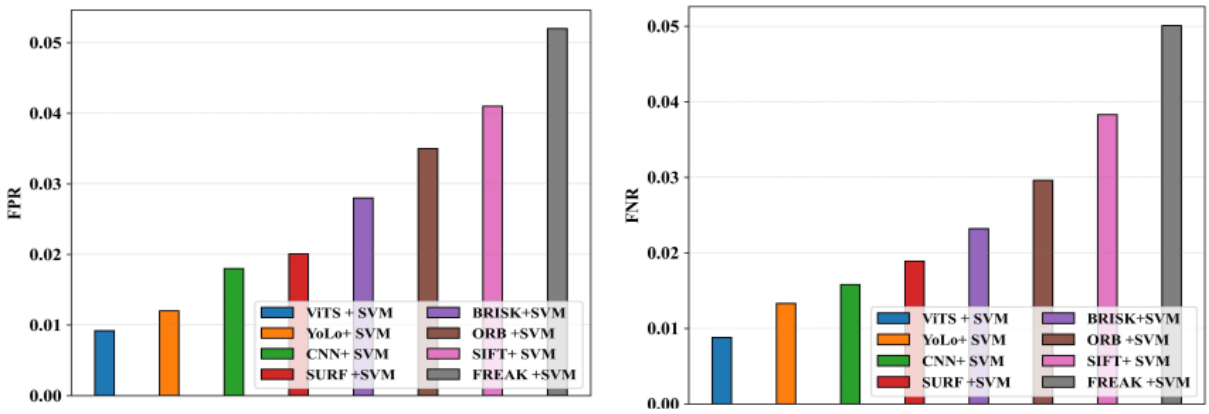


Figure 5: Analysis of Precision–Recall curve

The combined model of ViTs and SVM achieved the top performance in the comparison with other methods, as illustrated in Figure 5. To begin with, the self-attention mechanism of the Transformer's architectures (ViT) not only benefits from the robust feature representations captured by the model's long-range dependency abilities but also allows better feature extraction than conventional methods. Through the utilization of a large and diverse dataset, the ViTs have a better chance of generalization and thus a higher probability of performance improvement in terms of accuracy. Additionally, when the ViT is coupled with the SVM classifier, performance improves because SVM is designed to handle high-dimensional data more effectively than many conventional classifiers. Furthermore, the hyperparameters used with ViTs could have significantly impacted classifier effectiveness, as a hyperparameter optimally chosen can substantially influence model performance. Together, these elements contribute to the higher precision and recall values observed in the graph, distinguishing ViTs + SVM from other techniques such as YOLO, CNN, and SIFT. Figures 6 (a) and (b) display the FPR and FNR performance of the comparison between different feature descriptors with the SVM classifier.



(a) (b)

Figure 6: Analysis of FPR and FNR

Figure 6 demonstrates FPR and FNR of each feature descriptors model of an SVM classifier. ViTs + SVM yield the best model, as the rates of the FPR and FNR of 0.0092 and 0.9504 denote the capacity to provide accurate predictions with low error rates. The other models including the FREAK + SVM, SIFT + SVM and ORB + SVM have higher FPR and lower TPR meaning that they are performing poorly. The ViTs + SVM was working because the Vision Transformer allows learning deep features, and the SVM allows classifying complicated data more efficiently than conventional analysis.

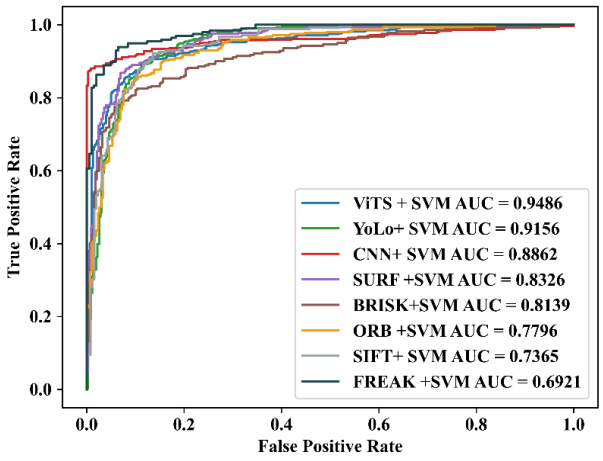


Figure 7: Analysis of ROC curve

Figure 7 indicates the ROC curves of each feature descriptor with the use of an SVM classifier. The best performing model is the Vision Transformer plus SVM which had an area under the curve of 0.9486. In comparison, YOLO + SVM had a value of 0.9156, CNN + SVM had a value of 0.8862, SURF + SVM had a value of 0.8326, BRISK + SVM had a value of 0.8139, ORB + SVM had a value of 0.7796, SIFT + SVM had a value of 0.7365, and FREAK + SVM had a value of 0.6921. Table 2 offers a more detailed analysis of the different feature descriptors with the SVM model.

Table 2:In-depth analysis of different feature descriptors with the SVM model

Performance (%)	ViTS+ SVM	YoLo+ SVM	CNN+ SVM	SURF+ SVM	BRISK+ SVM	ORB + SVM	SIFT+ SVM	FREAK + SVM
Accuracy	**95.04**	93.53	91.98	88.98	85.6	82.6	75.02	71.75
Precision	**95.24**	93.03	92.31	88.24	85.1	81.95	74.62	72.38

Recall	**95.92**	93.82	92.92	88.85	86.01	82.58	74.031	72.65
F1_score	**95.01**	92.96	92.15	88.32	85.36	82.27	73.21	70.93
FPR	**0.0092**	0.012	0.018	0.0201	0.028	0.035	0.041	0.052
FNR	**0.9504**	0.9353	0.9198	0.8898	0.856	0.826	0.7502	0.7175

The performance evaluation in Table 2 assesses key metrics, comprising Accuracy, Recall, Precision, F1 Score, Computational Time, False Positive Rate (FPR), as well as False Negative Rate (FNR), based on feature descriptors as well as an SVM classifier. The results show that ViTS+ SVM outperforms other methods, producing better performance in all evaluated metrics for object detection. Figure 8 displays the output of the different features extracted with the SVM classifier.

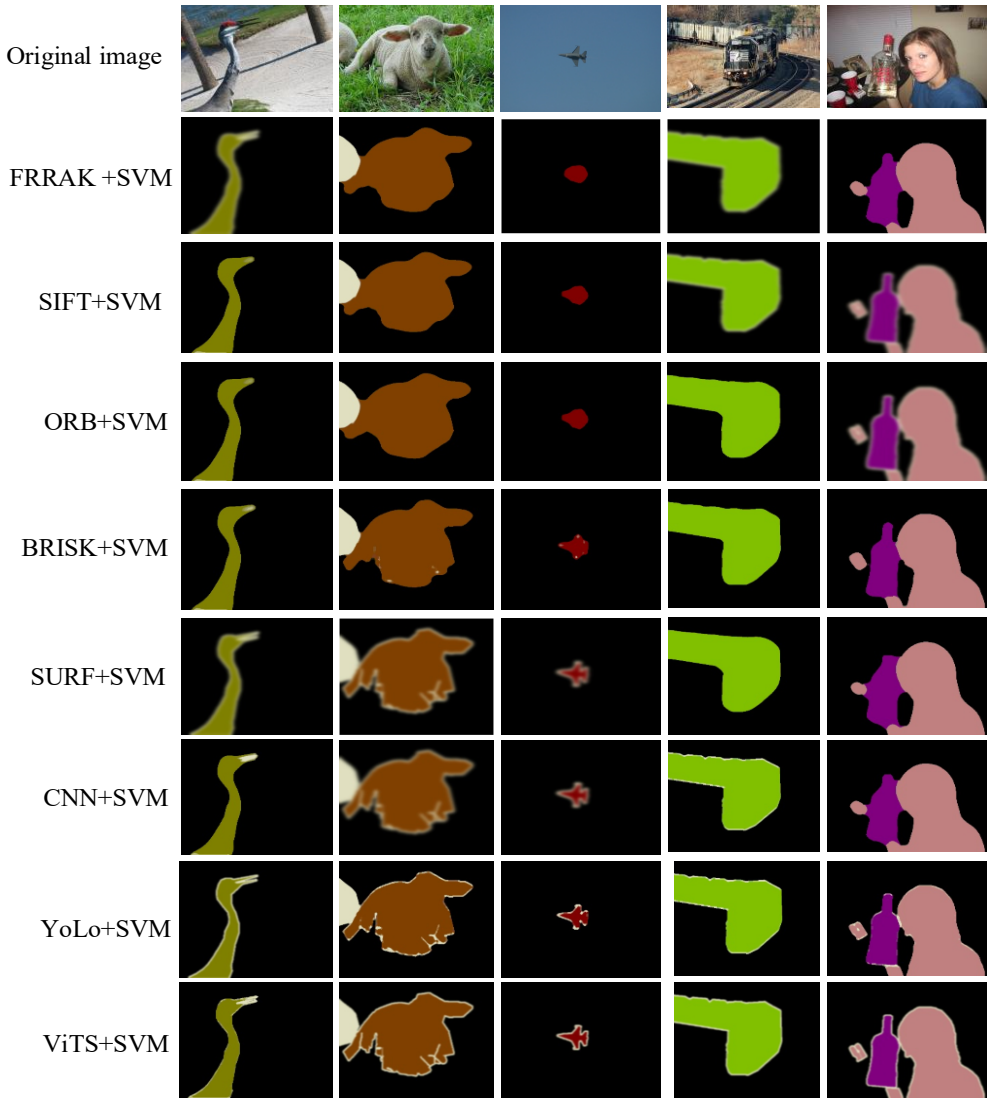


Figure 8: Outcome of the analysis method

4.4 Ablation study

The ablation study examines the interaction between various feature descriptors and an SVM classifier for object detection, as illustrated in Table 3. The ablation study is divided into several modules to consider their impact on accuracy, precision, and recall. Module 1 considers the GF, WF, AUMF, and without PBTMF. Module 2 considers PBTMF, AUMF, and those without GF and WF. Module 3 considers that GF and WF, without PBTMF and

AUMF. Module 4 is considered to be Only GF. Module 5 is considered to be only WF. Only AUMF is evaluated in Module 6, only PBTMF is studied in Module 7, and the Full Model is considered in Module 8.

Table 3: Ablation study result

Model	Module	Accuracy (%)	Precision (%)	Recall (%)
SIFT + SVM	Module 1	75.02	74.62	74.03
	Module 2	71.75	72.38	72.65
	Module 3	82.60	81.95	82.58
	Module 4	76.84	76.12	75.75
	Module 5	77.33	76.91	76.85
	Module 6	73.45	73.02	72.94
	Module 7	70.02	70.60	71.10
	Module 8	84.12	83.84	84.55
SURF + SVM	Module 1	88.98	88.24	88.85
	Module 2	85.60	85.10	86.01
	Module 3	91.98	92.31	92.92
	Module 4	87.23	87.55	87.04
	Module 5	88.10	87.68	88.25
	Module 6	84.70	84.20	84.95
	Module 7	82.11	81.74	82.42
	Module 8	93.01	93.00	93.00
ORB + SVM	Module 1	82.60	81.95	82.58
	Module 2	75.02	74.62	74.03
	Module 3	91.98	92.31	92.92
	Module 4	80.05	79.80	80.02
	Module 5	80.66	80.25	80.40
	Module 6	78.25	77.83	77.65
	Module 7	74.10	74.20	73.88
	Module 8	93.50	93.14	93.00
FREAK + SVM	Module 1	80.50	79.80	80.00
	Module 2	76.00	75.50	75.80
	Module 3	90.00	89.50	89.80
	Module 4	79.21	78.84	79.10
	Module 5	80.23	80.01	80.18
	Module 6	75.90	75.45	75.62
	Module 7	72.88	72.34	72.60
	Module 8	91.75	91.25	91.90
BRISK + SVM	Module 1	84.00	83.50	83.80
	Module 2	78.50	77.80	78.00
	Module 3	89.50	88.90	89.20
	Module 4	82.35	82.00	82.44
	Module 5	83.44	83.00	83.32
	Module 6	79.68	79.20	79.45
	Module 7	75.10	74.85	74.62
	Module 8	91.00	90.65	90.78
CNN + SVM	Module 1	92.00	91.50	91.80

	Module 2	89.00	88.50	88.80
	Module 3	94.50	94.00	94.20
	Module 4	91.45	91.10	91.00
	Module 5	91.95	91.45	91.60
	Module 6	88.10	87.85	87.95
	Module 7	86.02	85.90	86.15
	Module 8	93.00	94.90	94.90
YOLO + SVM	Module 1	93.50	93.00	93.30
	Module 2	90.00	89.50	89.80
	Module 3	91.00	94.50	94.80
	Module 4	92.65	92.25	92.40
	Module 5	93.10	92.85	92.95
	Module 6	89.40	89.00	88.95
	Module 7	87.00	86.75	86.85
	Module 8	93.00	94.80	94.90
ViTs + SVM	Module 1	95.00	95.00	95.00
	Module 2	93.53	93.03	93.82
	Module 3	91.98	92.31	92.92
	Module 4	94.12	94.00	94.08
	Module 5	94.45	94.28	94.30
	Module 6	92.10	91.80	92.00
	Module 7	90.22	89.95	90.15
	Module 8	95.00	95.24	95.12

The results of the experiments indicate that Module 8 consistently provided the best overall performance, suggesting that the combined performance of ViTs + SVM is at least 95% accurate, with maximum and minimum measurements. Generally, the experiment demonstrated that the combination of filtering methods proposed in the previous sections enhanced feature extraction, resulting in improved classification accuracy overall. Module 8, compared to the other configurations, clearly outperforms the others offered in this study.

5. Conclusion

In conclusion, the comparative analysis of SVM-based object detection with different feature descriptors using traditional and advanced feature descriptors. Investigated traditional descriptors like SIFT, SURF, FREAK, BRISK, ORB, and advanced models like CNNs, YOLO, and ViTs. This study also analyzes the image quality techniques, such as GF, WF, AUMF, and PBTMF. The experimental outcomes indicated that the ViTs+SVM model attained the best classification accuracy of 95.04%, surpassing the other models. The work also demonstrated the inadequacy of handcrafted features that mostly face problems in dealing with variations in environmental conditions and complicated image patterns. However, deep learning methods, particularly ViTs, tend to learn a powerful feature representation from large-scale data in a data-driven manner and consequently preserve significant details in the colorimetric space for accurate object detection.

In future work, improve the performance of the hybrid Vision Transformer by adding deep learning-based feature descriptors into the pipeline and designing new filtering models in object detection. Furthermore, investigations into other machine learning classifiers and feature fusion techniques are explored. Further explore adaptive algorithms for better robustness across the board and develop training techniques to improve model generalization.

Reference

1. Pathak, Ajeet Ram, Manjusha Pandey, and Siddharth Rautaray. "Application of deep learning for object detection." Procedia computer science 132 (2018): 1706-1717.

2. Wu, Xiongwei, Doyen Sahoo, and Steven CH Hoi. "Recent advances in deep learning for object detection." Neurocomputing 396 (2020): 39-64.
3. Kamel, Mohamed M., Sherif I. Hussein, Gouda I. Salama, and Yehia Z. Elhalwagy. "Efficient Target Detection Technique Using Image Matching Via Hybrid Feature Descriptors." In 2020 12th International Conference on Electrical Engineering (ICEENG), pp. 102-107. IEEE, 2020.
4. Joshi, Khushbu, and Manish I. Patel. "Recent advances in local feature detector and descriptor: a literature survey." International Journal of Multimedia Information Retrieval 9, no. 4 (2020): 231-247.
5. Kloster, Michael, Gerhard Kauer, and Bánk Beszteri. "SHERPA: an image segmentation and outline feature extraction tool for diatoms and other objects." BMC bioinformatics 15 (2014): 1-17.
6. Srikar, M., and K. Malathi. "A Supervised Stable Object Detection with Image Feature Extraction using Image Segmentation by Comparing Histogram of Oriented Gradients (HOG) Algorithm over Scale Invariant Feature Transform (SIFT) Algorithm Model." Journal of Pharmaceutical Negative Results 13 (2022).
7. Lee, K. L., and M. M. Mokji. "Automatic target detection in GPR images using Histogram of Oriented Gradients (HOG)." In 2014 2nd International Conference on Electronic Design (ICED), pp. 181-186. IEEE, 2014.
8. Bansal, Monika, Munish Kumar, and Manish Kumar. "2D object recognition: a comparative analysis of SIFT, SURF and ORB feature descriptors." Multimedia Tools and Applications 80, no. 12 (2021): 18839-18857.
9. Azeem, A., M. Sharif, J. H. Shah, and M. Raza. "Hexagonal scale invariant feature transform (H-SIFT) for facial feature extraction." Journal of applied research and technology 13, no. 3 (2015): 402-408.
10. Setiawan, A., R. A. Yunmar, and H. Tantriawan. "Comparison of speeded-up robust feature (SURF) and oriented FAST and rotated BRIEF (ORB) methods in identifying museum objects using Low light intensity images." In IOP conference series: earth and environmental science, vol. 537, no. 1, p. 012025. IOP Publishing, 2020.
11. Awaludin, M., & Yasin, V. (2020). Application Of Oriented Fast And Rotated Brief (Orb) And Bruteforce Hamming In Library Opencv For Classification Of Plants. Journal of Information System, Applied, Management, Accounting and Research, 4(3), 51-59.
12. Yang, Lihong, Shunqin Xu, Zhiqiang Yang, Jia He, Lei Gong, Wanjun Wang, Yao Li, Liguo Wang, and Zhili Chen. "Fast Registration Algorithm for Laser Point Cloud Based on 3D-SIFT Features." Sensors 25, no. 3 (2025): 628.
13. Nandeshwar, Vikas J., Sarvadnya Bhatlawande, Anjali Solanke, Harsh Sathe, Shivanand Satao, Safalya Satpute, and Atharv Saste. "Comparative analysis of feature descriptors and classifiers for real-time object detection." Int J Reconfigurable & Embedded Syst 14, no. 1 (2025): 89-99.
14. Yadav, Satya Prakash, Muskan Jindal, Preeti Rani, Victor Hugo C. de Albuquerque, Caio dos Santos Nascimento, and Manoj Kumar. "An improved deep learning-based optimal object detection system from images." Multimedia Tools and Applications 83, no. 10 (2024): 30045-30072.
15. Yao, Hui, Yaning Fan, Yanhao Liu, Dandan Cao, Ning Chen, Tiancheng Luo, Jingyu Yang, Xueyi Hu, Jie Ji, and Zhanping You. "Development and optimization of object detection technology in civil engineering: A literature review." Journal of Road Engineering (2024).
16. Parupalli, SriPadma, Siddi Akhsitha, Diksha Naval, Prathyusha Kasam, and Suprajareddy Yavagiri. "Performance evaluation of YOLOv2 and modified YOLOv2 using face mask detection." Multimedia Tools and Applications 83, no. 10 (2024): 30167-30180.
17. Chai, Bosong, Xuan Nie, Qifan Zhou, and Xingyu Zhou. "Enhanced cascade R-CNN for multi-scale object detection in dense scenes from SAR images." IEEE Sensors Journal (2024).
18. Qing, Chen, Tong Xiao, Shuzhuang Zhang, and Peng Li. "Region Proposal Networks (RPN) Enhanced Slicing for Improved Multi-Scale Object Detection." In 2024 7th International Conference on Communication Engineering and Technology (ICCET), pp. 66-70. IEEE, 2024.
19. Mao, Makara, and Min Hong. "YOLO Object Detection for Real-Time Fabric Defect Inspection in the Textile Industry: A Review of YOLOv1 to YOLOv11." Sensors (Basel, Switzerland) 25, no. 7 (2025): 2270.
20. Law, Hei, Yun Teng, Olga Russakovsky, and Jia Deng. "Cornernet-lite: Efficient keypoint based object detection." arXiv preprint arXiv:1904.08900 (2019).
21. Pang, Shuyang, Xuan Liu, Shangwei Mao, Hongsheng Jia, and Bin Liu. "Advanced-ExtremeNet: Combined with Depthwise Separable Convolution for the Detection of Steel Bars." In 2021 2nd International Conference on Artificial Intelligence and Information Systems, pp. 1-6. 2021.
22. Zhou, Zhili, Xiaohua Dong, Zhetao Li, Keping Yu, Chun Ding, and Yimin Yang. "Spatio-temporal feature encoding for traffic accident detection in VANET environment." IEEE Transactions on Intelligent Transportation Systems 23, no. 10 (2022): 19772-19781.
23. Mahaur, Bharat, and K. K. Mishra. "Small-object detection based on YOLOv5 in autonomous driving systems." Pattern Recognition Letters 168 (2023): 115-122.
24. Singh, Kiran Jot, Divneet Singh Kapoor, Khushal Thakur, and Anshul Sharma. "Computer-vision based object detection and recognition for service robots in indoor environments." Computers, Materials & Continua 72, no. 1 (2022).
25. Jiang, Du, Gongfa Li, Chong Tan, Li Huang, Ying Sun, and Jianyi Kong. "Semantic segmentation for multi-scale target based on object recognition using the improved Faster-RCNN model." Future Generation Computer Systems 123 (2021): 94-104.

26. Zhao, Lei, Luqian Zhi, Cai Zhao, and Wen Zheng. "Fire-YOLO: a small target object detection method for fire inspection." Sustainability 14, no. 9 (2022): 4930.
27. Shao, Zeyuan, Yong Yin, Hongguang Lyu, C. Guedes Soares, Tao Cheng, Qianfeng Jing, and Zhilin Yang. "An efficient model for small object detection in the maritime environment." Applied Ocean Research 152 (2024): 104194.
28. Hussain, Nazar, Muhammad Attique Khan, Muhammad Sharif, Sajid Ali Khan, Abdulaziz A. Albesher, Tanzila Saba, and Ammar Armaghan. "A deep neural network and classical features-based scheme for object recognition: an application for machine inspection." Multimedia Tools and Applications (2024): 1-23.
29. Lew, Kai Liang, Chung Yang Kew, Kok Swee Sim, and Shing Chiang Tan. "Adaptive Gaussian Wiener Filter for CT-Scan Images with Gaussian Noise Variance." Journal of Informatics and Web Engineering 3, no. 1 (2024): 169-181.
30. Göreke, Volkan. "A novel method based on Wiener filter for denoising Poisson noise from medical X-Ray images." Biomedical Signal Processing and Control 79 (2023): 104031.
31. Yan, Yaping, Hongjuan Zhang, Songlin Du, and Yide Ma. "Bi-SCM: bidirectional spiking cortical model with adaptive unsharp masking for mammography image enhancement." Multimedia Tools and Applications 82, no. 8 (2023): 12081-12098.
32. Jana, Bhaskara Rao, Haritha Thotakura, Anupam Baliyan, Majji Sankararao, Radhika Gautamkumar Deshmukh, and Santoshachandra Rao Karanam. "Pixel density based trimmed median filter for removal of noise from surface image." Applied Nanoscience 13, no. 2 (2023): 1017-1028.
33. Burger, Wilhelm, and Mark J. Burge. "Scale-invariant feature transform (SIFT)." In Digital Image Processing: An Algorithmic Introduction, pp. 709-763. Cham: Springer International Publishing, 2022.
34. Tarek, Zahraa, Samaa M. Shohieb, Abdelghafar M. Elhady, El-Sayed M. El-Kenawy, and Mahmoud Y. Shams. "Eye Detection-Based Deep Belief Neural Networks and Speeded-Up Robust Feature Algorithm." Computer Systems Science & Engineering 46, no. 1 (2023).
35. Shabbir, Kazi Safayet Md, Md Imteaz Ahmed, and Marzan Alam. "Detection of Glaucoma using ORB (Oriented FAST and Rotated BRIEF) Feature Extraction." Journal of Engineering Advancements 2, no. 03 (2021): 153-158.
36. Ghasemi Yegane, Ardeshir, Kourosh Kiani, and Razieh Rastgoo. "Copy-Move Forgery Detection Using Fast Retina Keypoint (FREAK) Descriptor." Modeling and Simulation in Electrical and Electronics Engineering 2, no. 2 (2022): 1-7.
37. Ghaffari, Sina, David W. Capson, and Kin Fun Li. "A fully pipelined FPGA architecture for multi-scale BRISK descriptors with a novel hardware-aware sampling pattern." IEEE Transactions on Very Large Scale Integration (VLSI) Systems 30, no. 6 (2022): 826-839.
38. AlSaeed, Duaa, and Samar Fouad Omar. "Brain MRI analysis for Alzheimer's disease diagnosis using CNN-based feature extraction and machine learning." Sensors 22, no. 8 (2022): 2911.
39. Garcia-Martin, Raul, and Raul Sanchez-Reillo. "Vision transformers for vein biometric recognition." IEEE Access 11 (2023): 22060-22080.
40. Liu, Linrunjia, Gaoshuai Wang, and Qiguang Miao. "ADYOLOv5-Face: An Enhanced YOLO-Based Face Detector for Small Target Faces." Electronics 13, no. 21 (2024): 4184.
41. Wang, Jianfei. "Optimizing support vector machine (SVM) by social spider optimization (SSO) for edge detection in colored images." Scientific Reports 14, no. 1 (2024): 9136.