



An efficient technique for object recognition using Shi-Tomasi corner detection algorithm

Monika Bansal¹ · Munish Kumar² · Manish Kumar³ · Krishan Kumar⁴

Published online: 27 November 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

An efficient feature detection algorithm and image classification is a very crucial task in computer vision system. There are various state-of-the-art feature detectors and descriptors available for an object recognition task. In this paper, the authors have compared the performance of Shi-Tomasi corner detector with SIFT and SURF feature descriptors and evaluate the performance of Shi-Tomasi in combination with SIFT and SURF feature descriptors. To make the computations faster, authors have reduced the size of features computed in all cases by applying locality preserving projection methodology. Features extracted using these algorithms are further classified with various classifiers like K-NN, decision tree and random forest. For experimental work, a public dataset, namely Caltech-101 image dataset, is considered in this paper. This dataset comprises of 101 object classes. These classes have further contained many images. Using a combination of Shi-Tomasi, SIFT and SURF features, the authors have achieved a recognition accuracy of 85.9%, 80.8% and 74.8% with random forest, decision tree and K-NN classifier, respectively. In this paper, the authors have also computed true positive rate, false positive rate and area under curve in all cases. Finally, the authors have applied the adaptive boosting methodology to improve the recognition accuracy. Authors have reported improved recognition accuracy of 86.4% using adaptive boosting with random forest classifier and a combination of Shi-Tomasi, SIFT and SURF features.

Keywords Decision tree · LPP · k-NN · Random forest · Shi-Tomasi · SIFT · SURF

1 Introduction

Nowadays, object recognition is a hot research area in the domain of image processing and computer vision where an object is recognized from an image. This system works just like a child learned in school. A child is trained in school by learning various shapes and object names. When he learned all objects, he can determine all similar objects that

he has already learned. The machine must be fully trained with various machine learning algorithms by adding all the object names just like a child learn. The training process is done by storing all the features of similar objects in a database. Then, an input image is to be tested by matching the features of the input image with the stored feature dataset of the images. An efficient object recognition system will output the name of the object correctly and in less time. An object can have many features as texture, color, shape, etc. A database of these extracted features is maintained and matched with the features of the query image to recognize an object. In this paper, the authors have used a corner detection algorithm, i.e., Shi-Tomasi corner detector to identify the object in an image. Based on experimental work, authors have noticed that this algorithm alone is not enough to achieve efficient results in terms of accuracy. So, the authors further used SIFT and SURF feature descriptors with Shi-Tomasi corner detector in combination to improve the object recognition results.

Authors have depicted the individual results of all these three feature detectors and descriptors by using K-NN,

Communicated by V. Loia.

✉ Munish Kumar
munishcse@gmail.com

¹ Department of Computer Science, Punjabi University, Patiala, India

² Department of Computational Sciences, Maharaja Ranjit Singh Punjab Technical University, Bathinda, Punjab, India

³ Department of Computer Science, Baba Farid College, Bathinda, Punjab, India

⁴ University Institute of Engineering and Technology, Panjab University, Chandigarh, India

decision tree and random forest classifier. Based on the experimental work, authors have noticed that alone Shi-Tomasi corner detector algorithm outperforms as compared to SIFT and SURF descriptors. But, when they combined all these three algorithms, they achieved a greater improvement in their results. Experiments are conducted using a public dataset, namely Caltech 101 image dataset. This dataset is a collection of 101 object classes that contain 40–800 images in each class. It is a very large database having more than 9000 images. Authors have considered 80% of the total images in each class as training, and the remaining 20% from each class is taken for testing. Shi-Tomasi, SIFT and SURF output a large collection of features which take a lot of memory. So, k-means clustering algorithm is used after SIFT and SURF descriptor algorithms and locality preserving projection (LPP) on all these three algorithms. The use of k-means clustering and LPP made the size of the feature database small. The extracted features with all three feature detection algorithms were further classified using k-NN, decision tree and random forest algorithm. Recognition accuracy of 74.7%, 80.8% and 85.8% has been achieved using k-NN, decision tree and random forest classifier, respectively. In this paper, the authors have also presented true positive rate (TPR), false positive rate (FPR) and area under curve (AUC). The experimental results are discussed in Sect. 6. Authors have observed that random forest classifier outperforms as compared to other classifiers 85.8% (TPR), 0.2% (FPR) and 99.7% (AUC). Adaptive boosting and the bootstrap aggregating scheme were also applied to improve the accuracy results which made a slight increase in recognition accuracy and true positive rate. This paper summarizes into 7 sections. Section 2 presents the prior work related to object recognition. In Sect. 3, authors have described a short description on Shi-Tomasi corner detector and scale-invariant feature transform (SIFT) and speed up robust features (SURF) feature descriptors. Section 4 will present some briefing on various classifiers used in our experiment. Section 5 will give the detail of the proposed work and the other methods used to improve the results for object recognition in this paper. In Sect. 6, authors have discussed experimental results. Section 7 will present the conclusion drawn from this work.

2 Related work

In this section, the authors have presented existing work related to object recognition, for example, Murphy-Chutorian et al. (2005) presented an appearance-based object recognition system. They used Gabor wavelet response to extract the feature only at the corner points. They experimented this method on 50 objects in cluttered

scenes. A feature sharing approach was adopted by the authors that worked in three phases—clustering, training and recognition. Through feature sharing technique, they avoided the learning and storing feature representation phases. This made the recognition process fast. A feature dictionary was created by applying k-means clustering algorithm on the features extracted using the Gabor algorithm in the first phase. In the training phase, the image is cropped to differentiate background and foreground regions. The Gabor-jet is extracted at each interest point in the foreground and is compared with the feature dictionary. And in recognition phase, a bottom-up search approach is used to compare the extracted features using Gabor-jet with feature dictionary. This proposed approach increased the speed of the object recognition process. Welke et al. (2006) proposed a fast and robust recognition technique for finding multiple objects in an image. They have done experiments using their proposed method on a collection of images of the kitchen environment. The database was comprised of 20 objects, and this system will analyze the scene in 350 ms. They used k-means clustering algorithm and principal component analysis (PCA). Bosch et al. (2007) presented an image classification by combining three approaches. The feature extraction algorithm was selected based on shape and appearance. SIFT algorithm was used to detect the appearance of an image, and histogram of gradients (HOG) was used to detect the local shape of the image. Using these algorithms, a spatial pyramid was developed over a region of interest which is based on spatial pyramid matching algorithm. Second approach they adopted was to select region of interest in the background cluttering. In the third approach, they used random forest classifier over multi-way SVM classifier for classification. The experiment was done on the Caltech-101 dataset and Caltech-256 dataset.

Kang et al. (2007) presented principal component analysis (PCA) and k-NN classifier for the object recognition system. They made some improvement in k-NN classifier that helped to improve the recognition accuracy to 91.2%. Here, they analyzed the features by using a group of object models of the same class and then recognize the image by matching a group of similar object class with the features of an inputted image. Azad et al. (2009) proposed a combination of the Harris corner detector and SIFT descriptor for object recognition. They used this approach to speed up the recognition process. In this paper, the results are evaluated based repeatability, accuracy and speed. Schmidt et al. (2010) presented a comparative performance evaluation of different feature detector-descriptor pairs. They used various corner detector algorithms—Harris corner detector, Shi-Tomasi feature detector, FAST corner detector, SURF feature detector and star key point detector. These feature detectors are combined with various

feature descriptors like sum of absolute differences (SAD), sum of squared differences (SSD), normalized cross-correlation (NCC) and SURF family of descriptors. This experiment proved that Shi-Tomasi performed better with SURF family. Wu et al. (2012) proposed an efficient pruning technique to speed up the feature detection task using Shi-Tomasi and Harris corner detector algorithm. Using this technique, they cropped the non-corners of an image by applying a threshold on the corner measures. The experiment done by them proved 90% and 70% execution time reduction for Shi-Tomasi and Harris corner detector, respectively. Muralidharan et al. (2014) proposed k-nearest neighbor classifier using eigenvalues. The features were computed using local and global features of the image. These computed features were further classified using k-NN classifier, back propagation neural network (BPN) and fuzzy k-NN classifier. A comparison among these three classifiers was presented and the results of k-NN classifier using eigenvalues outperformed as compared to BPN and fuzzy k-NN classifier with 97% accuracy. The experiments were performed on 36 objects of COIL-100 dataset. Fularz et al. (2015) presented a hardware implementation of a decision tree classifier on image-based object recognition. They computed the uniform local binary patterns (ULBPs) for each image. The individual ULBPs were not enough for classification. So, they divided the image into a number of rectangular cells. The features extracted from the division of the image are taken as characteristics for a given class of object. This implementation performs more efficiently in terms of processing speed and power consumption. Gupta et al. (2019) have presented an efficient object recognition system using SIFT and ORB feature detector. They have achieved a precision rate of 69.8% and 76.9% using ORB and SIFT feature descriptors, respectively. Using a combination of ORB and SIFT feature descriptors, they have achieved a precision rate of 85.6%. Bansal et al. (2020a) have presented a comprehensive study in the field of 2D object recognition. In this paper, various feature extraction techniques and classification algorithms are discussed which are required for object recognition. They have also presented an extreme gradient based technique for object recognition (Bansal et al. 2020b).

3 Feature detector and descriptors

An image can be easily recognized by obtaining its features. For an example, one can easily recognize a table, chair, animal, etc. Every object has its own properties that make it different from other objects. But sometimes the identity of an object may be unrecognized in the case of illumination, rotation, scaling or occlusion. There are various feature detection and description algorithms available

that help to extract the features from an image. These algorithms determine color, texture, shape, etc., depending on the type of their functioning. Some objects can be easily identified with color, while others can be identified with shape or texture or some other feature. In this paper, the authors have experimented Shi-Tomasi corner detector algorithm because it works better than other corner detector algorithms. The authors have also experimented scale-invariant feature transform (SIFT) and speed up robust feature (SURF). SIFT and SURF are state-of-the-art feature detector and descriptor algorithms. To analyze the recognition accuracy, a most challenging image dataset, i.e., Caltech-101 image dataset, is considered for this work.

3.1 Shi-Tomasi corner detector algorithm

This algorithm was developed by Shi et al. (1994). In computer vision, feature extraction is the most crucial task. Shi et al. (1994) have proposed a good feature to track in their paper. They have done experiments with a dataset of large images and large collection of features. Their experiment maximized the quality of tracking. In this paper, even they covered all the problems that arise during the tracking of the image. To achieve the efficient results, they have done a small improvement in Harris corner detector by computing the minimum of absolute values of the eigenvalues of structural data.

$$R = \min(\lambda_1, \lambda_2)$$

Corners are the local maxima of R . if R is greater than certain predefined value, then it may be marked as a corner. Rather Harris corner detector algorithm adopted corner selection criteria in which a score is calculated for each pixel. This algorithm was developed by Harris et al. (1988). The algorithm is also based on two eigenvalues of a matrix. The score calculated on two eigenvalues is compared against a certain value. If score exceeds that value, the pixel is considered as the corner. The score (R) is calculated using the following function:-

$$R = \det M - k(\text{trace}M)^2$$

$$\det M = \lambda_1 \lambda_2$$

$$\text{trace}M = \lambda_1 + \lambda_2$$

By making the change in score calculation method, Shi-Tomasi has improved the corner detection results. The execution of this algorithm is also very fast as compared to the Harris corner detector. Kenney et al. (2005) compared various corner detector algorithms in the paper. The algorithms compared were Harris-Stephens, Forstner, Shi-Tomasi, Rohr and the p-norm condition detector. They set up a framework of four axioms on which basis the comparison is analyzed among these algorithms. These axioms were

formulated based on rotation, isotropy condition, orthonormal columns and some other factors. In the experiment, Shi-Tomasi algorithm was the only algorithm that satisfied all these four axioms. Li et al. (2016) proposed region of interest (ROI) and optimal bag-of-words (BOW) model for object recognition system. The proposed method works in many phases. First, ROI is obtained by extracting the features of images using Shi-Tomasi corner detector and Itti saliency map. In the second phase, SIFT feature detector and descriptor is extracted on these interest points. In the third phase, k-mean + + clustering is applied to make the clusters, and then, Gaussian mixture model (GMM) is used to model the feature cluster as visual words. In fourth phase, posterior pseudo-probabilities discriminative is used to find the similarities between visual words and corresponding local feature. This helps to build visual word histogram for image representation. In fifth phase, support vector machine (SVM) classifier is used for classification and recognition. The experiment was done on MSRC-21 dataset which contains 591 images from 21 object classes. They compared the accuracy of their proposed work with Galleguillos et al. (2008) and McFee et al. (2011). They proved more accurate results than the work done by these authors.

3.2 SIFT (scale-invariant feature transform)

SIFT is a feature detector and descriptor for object recognition that is developed by Lowe (1999). SIFT extracts 128 features for each point of interest in the image. SIFT algorithm creates an array of oriented histogram of size 4×4 with 8 bins, and these 8 bins are 8 angles of orientation. SIFT is invariant to rotation, scale, illumination and viewpoint. But it has one major drawback that it takes more computation time to extract the features.

3.3 SURF (speed up robust feature)

SURF is an image feature detector and descriptor. This algorithm is an improvement over the SIFT (scale-invariant feature transform) feature detector and descriptor as SIFT extracts the feature at slow speed. So, to make the calculations faster, SURF was designed by using Haar wavelet approximation of blob detector that is based on the Hessian determinant and on gradients. SURF extracts 64 or 128 features for each point of an image. SURF was introduced by Bay et al. (2006) for object recognition. Sometimes, SURF shows more accurate results as compared to SIFT feature detector. In this paper, we determined more performance accuracy in SURF as compared to SIFT.

4 Classification Techniques

Image classification is an important part of an object recognition system. Various classification algorithms are used to classify similar objects from other images. Each group of similar images assigns a class name. The object recognition system uses an image classification algorithm to assign a name to the identified image based on various features extracted from the image. There are various state-of-the-art classifiers available like Naïve Bayes, support vector machine (SVM), k-NN, decision tree, random forest classifier, etc. In this paper, the authors have considered k-NN, decision tree and random forest classifier, because these three classifiers are performing better than other existing classifiers.

4.1 K-NN (K-Nearest Neighbor)

K-Nearest neighbor is a supervised learning algorithm that creates the group of objects by classifying their features based on their nearest neighbor. To find the nearest neighbor, Euclidean distance is used as a distance metric to find the distance between two feature vectors. The distance calculated is then sorted and k-minimum distances are picked (the value of k may be any integer value). As this algorithm chooses the k-nearest points, this algorithm is called k-NN classifier. After choosing k sorted distance points, a majority voting is applied on this to assign the name to the object. Anupama et al. (2015) presented k-NN-based object recognition system. They implemented the approach to analyze EEG signals in real life. To implement it, they used k-NN classifier, principal component analysis (PCA) and singular value composition (SVC). This approach achieved 75% accuracy in recognizing the objects. Kim et al. (2012) presented a comparative analysis between k-NN classifier and support vector machine (SVM). They used Caltech-4 cropped image dataset for the recognition task. Bag-of-words (BOW) model was adopted using scale-invariant feature transforms (SIFT) algorithm. The computed features are then clustered into similar groups using k-means clustering algorithm. A histogram of these code words is created that is referred as training data. These data are then classified using k-NN classifier and SVM classifier. The results showed that SVM outperformed as compared to k-NN classifier with 90.56% accuracy.

4.2 Decision tree

The decision tree is a multistage classifier that works by partitioning the extracted features of a collection of images into subparts. These subparts are taken as nodes in this tree.

The partitioning of features should satisfy some decision rules. The division of features into subgroups goes on till no further division possible. Decision tree consists of a root, nodes and leaves. A root is a collection of features extracted from an image dataset. Each node will represent a subset of features, and each leaf will represent an object class. This classifier works in two phases—in the first phase, a decision tree is created from the image dataset under training, and in the second phase, the features extracted from the inputted image are compared with each trained node till the leaf is not found. Finally, the leaf node will output the name of the class to which that object belongs. Hauska et al. (1975) described a detail on the working of the decision tree classifier. They used two methods to design the decision tree and discussed the experimental results derived from these methods. In their paper, they also explained various advantages and disadvantages of both approaches. Du et al. (2002) presents how a decision tree classifier can be created with private data. They presented the solution to the problems arise out of semi-trusted servers. They also discussed the security issues in a decision tree classification method.

4.3 Random forest

Random forest classifier is a multistage classifier that contains a collection of binary decision trees. Forest is designed by ensemble various binary tree decision trees. This classifier can be used for clustering, classification or regression problem. A binary decision tree is built by splitting the input feature vector using decision rules. Splitting of features continued till no further division possible. Each leaf of the tree represents the class label of the object. Random forest classifier is very simple and fast algorithm to train the data and classify various images. Gall et al. (2012) proposed a multi-class object detection approach using random forest classifier. They have used random forest classifier on various object classes and achieved acceptable recognition results.

4.4 Adaptive boosting

Adaptive boosting is a method to improve the image classification results. Adaptive boosting is a machine learning algorithm. In this algorithm, the output of various classifiers is combined into a weighted sum that outcomes the results of adaptive boosting. The basic purpose of this algorithm is to transform the weaker classifier into strong ones and improve the recognition accuracy. Guo et al. (2001) proposed adaptive boosting algorithm for face recognition. They explained that AdaBoost is a typical classifier that classifies only two classes. But, the majority voting scheme can be used to solve multi-class recognition.

Majority voting scheme combines the results of various classifiers using a weighted sum.

4.5 Bootstrap aggregation

Bootstrap aggregation is a machine learning ensemble approach used to improve the performance computed using a decision tree. Decision tree has high variance as if there is some change in trained data, the resulting decision tree will be quite different. The bootstrap aggregation method is used to reduce the variance for these algorithms. The bootstrap aggregation approach also avoids over-fitting of the data. A majority voting scheme or average is used to combine the results of various classifiers. The outcome of bootstrap aggregation will provide improved accuracy for weak classifiers like decision tree, but not for strong classifiers like Naïve Bayes.

5 Proposed Algorithm

In this paper, the authors have experimented various feature detectors and descriptors like Shi-Tomasi, scale-invariant feature transform (SIFT) and speed up robust feature (SURF) for object recognition system. These extracted features are further classified to determine the class label for an object. The experiments are done using various classifiers like k-nearest neighbor, decision tree and random forest classifier. A comparative view of the results taken is also shown in Sect. 6. Authors observe that random forest classifier depicts better results with a combination of Shi-Tomasi, SIFT and SURF features. The Caltech-101 image dataset is used for the experiment. Caltech-101 is a large dataset that comprises more than 9000 images. These images are grouped into 101 classes and each class contains 40–800 images. To improve the accuracy, we have also applied to the majority voting scheme, adaptive boosting and bootstrap aggregation on the features computed using Shi-Tomasi, SIFT and SURF feature detector. There was slight improvement after applying adaptive boosting scheme.

Proposed system works under following steps:

1. Chose the image dataset as Caltech101 image dataset
2. Computed feature vector using Shi-Tomasi corner detector algorithm, SIFT and SURF individually and with a different combination.
3. Applied normalization on each feature (V_i) computed using Shi-Tomasi corner detector. A normalization function (f) as

Fig. 1 Block diagram of the proposed object recognition system

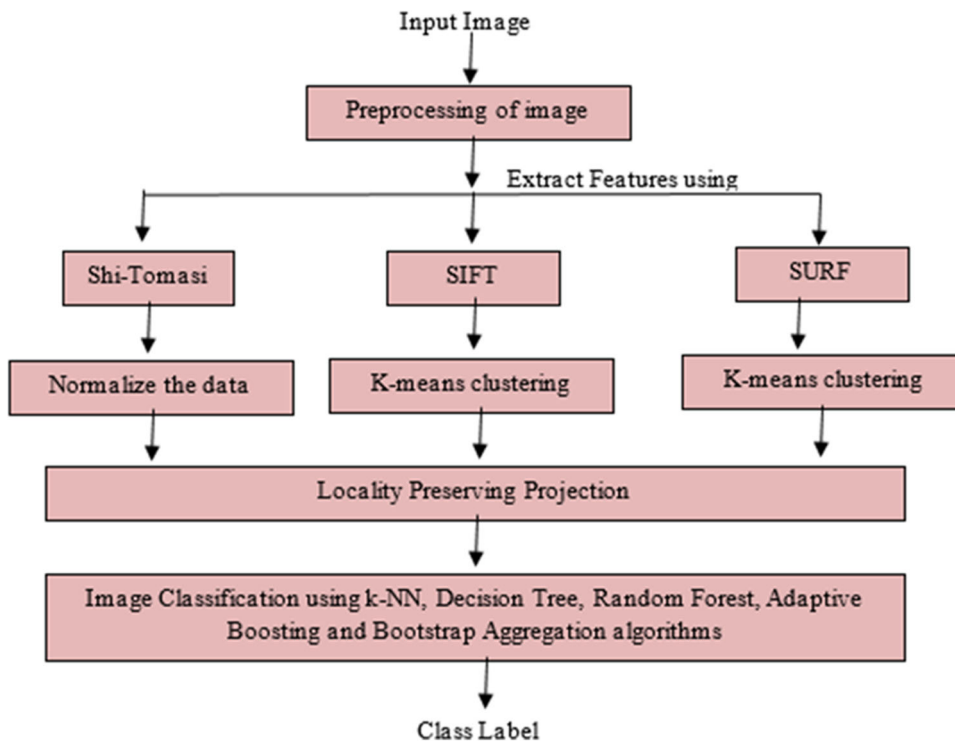


Table 1 Quantitative Comparison of Different Feature Extraction Techniques for Object Recognition (classifier wise recognition accuracy)

Feature extraction techniques	Recognition accuracy (Training set: 80% and testing set: 20%)			
	k-NN (C1)	Decision tree (c2)	Random forest (C3)	C1 + C2 + C3 (Majority voting scheme)
Shi-Tomasi (F1)	63.5%	64.1%	72.4%	67.6%
SIFT (F2)	35.0%	37.3%	39.7%	37.7%
SURF (F3)	64.7%	59.4%	66.4%	67.9%
Shi-Tomasi (F1) + SIFT (F2)	68.0%	71.4%	78.4%	70.5%
Shi-Tomasi (F1) + SURF (F3)	70.3%	75.0%	78.1%	77.4%
SIFT (F2) + SURF (F3)	71.2%	71.4%	78.4%	76.4%
Shi-Tomasi (F1) + SIFT (F2) + SURF (F3)	74.8%	80.8%	85.9%	82.7%

$$f = \frac{Vi - \min}{\max - \min}$$

where min and max are the minimum and maximum values of the feature vector.

4. K-means clustering was applied to the output of SIFT and SURF feature descriptors so that nearest neighbor features of an object can be used for recognition phase. As it will be very time-consuming to make a correspondence among all the features of an object, so this clustering algorithm helps to determine the

nearest centroids and correct correspondence for each member of a class.

5. Then, locality preserving projection (LPP) method is applied to the data after normalization and clustering results. So, the size of the feature vector can be reduced.
6. In the training phase, various classifiers like k-NN, decision tree and random forest are used on 80% of data computed in step 5.
7. In the testing phase, remaining 20% of data is used.
8. The results are evaluated by taking a comparative study of these combinations. In this experiment, we

Table 2 Quantitative Comparison of Different Feature Extraction Techniques for Object Recognition (classifier wise true positive rate)

Feature extraction techniques	True positive rate (Training set: 80% and testing set: 20%)			
	k-NN (C1)	Decision tree (C2)	Random forest (C3)	C1 + C2 + C3 (Majority voting scheme)
Shi-Tomasi (F1)	63.5%	64.1%	72.4%	67.6%
SIFT (F2)	35.0%	37.3%	39.7%	37.7%
SURF (F3)	64.7%	59.4%	66.4%	68.0%
Shi-Tomasi (F1) + SIFT (F2)	68.0%	71.4%	78.4%	70.5%
Shi-Tomasi (F1) + SURF (F3)	70.3%	75.0%	78.1%	77.4%
SIFT (F2) + SURF (F3)	71.2%	71.4%	78.4%	76.4%
Shi-Tomasi (F1) + SIFT (F2) + SURF (F3)	74.7%	80.8%	85.8%	82.7%

Table 3 Quantitative Comparison of Different Feature Extraction Techniques for Object Recognition (classifier wise false positive rate)

Feature extraction techniques	False positive rate (Training set: 80% and testing set: 20%)			
	k-NN (C1)	Decision tree (C2)	Random forest (C3)	C1 + C2 + C3 (Majority voting scheme)
Shi-Tomasi (F1)	0.5%	0.4%	0.3%	0.4%
SIFT (F2)	0.7%	0.7%	0.6%	0.6%
SURF (F3)	0.7%	0.9%	0.7%	0.7%
Shi-Tomasi (F1) + SIFT (F2)	0.3%	0.3%	0.2%	0.3%
Shi-Tomasi (F1) + SURF (F3)	0.5%	0.3%	0.3%	0.3%
SIFT (F2) + SURF (F3)	0.5%	0.3%	0.3%	0.3%
Shi-Tomasi (F1) + SIFT (F2) + SURF (F3)	0.3%	0.2%	0.2%	0.2%

Table 4 Quantitative Comparison of Different Feature Extraction Techniques for Object Recognition (classifier wise area under curve)

Feature extraction techniques	Area under curve (Training set: 80% and testing set: 20%)			
	k-NN (C1)	Decision tree (C2)	Random forest (C3)	C1 + C2 + C3 (Majority voting scheme)
Shi-Tomasi (F1)	81.5%	93.9%	98.7%	83.6%
SIFT (F2)	67.1%	76.2%	81.2%	68.6%
SURF (F3)	82.0%	91.6%	96.5%	83.7%
Shi-Tomasi (F1) + SIFT (F2)	83.9%	95.5%	99.2%	85.1%
Shi-Tomasi (F1) + SURF (F3)	84.9%	96.9%	99.4%	88.5%
SIFT (F2) + SURF (F3)	85.4%	94.8%	99.0%	88.1%
Shi-Tomasi (F1) + SIFT (F2) + SURF (F3)	87.2%	97.5%	99.7%	91.2%

Fig. 2 Classifier wise recognition accuracy

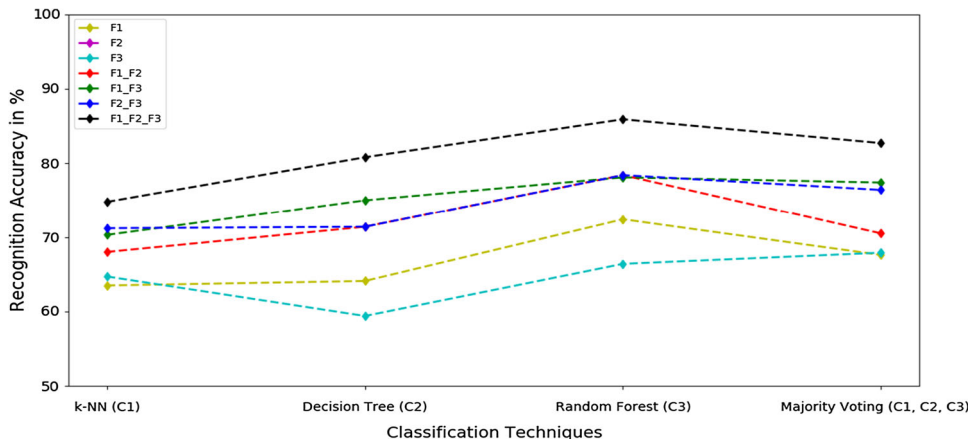


Fig. 3 Classifier wise true positive rate

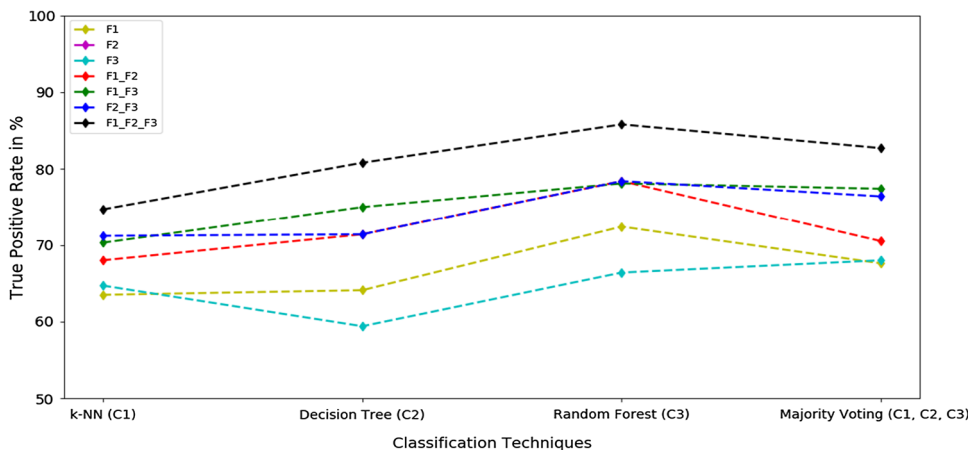
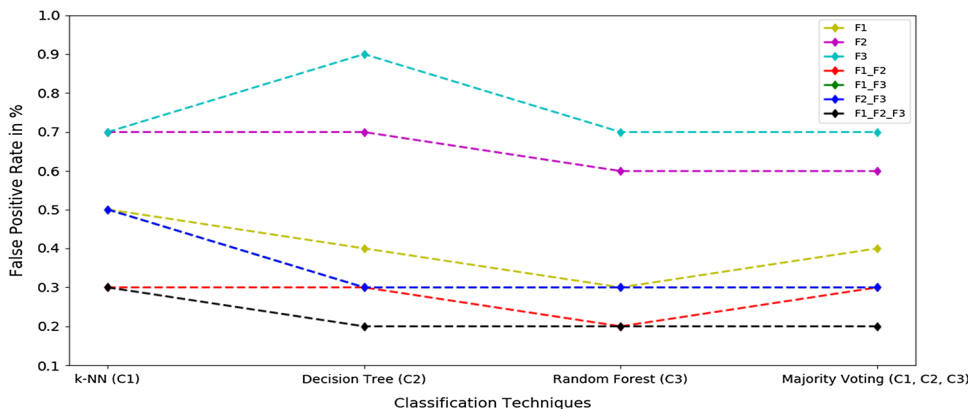


Fig. 4 Classifier wise false positive rate



determined that random forest classifier outperformed as compared to other classifiers when a combination of above three feature detector and descriptors is used.

- To improve the accuracy, the authors have also applied adaptive boosting scheme and bootstrap aggregation scheme and got some improvement after applying adaptive boosting.

6 Performance Analysis

We evaluated the performance of our proposed system on the Caltech-101 image dataset. This dataset contains more than 9000 images that are classified into 101 classes. Each class contains 40–800 images (Fig. 1). We implemented our proposed system by taking 80% of images from each class of dataset as training data and 20% images for each class as testing dataset. In this work, we have evaluated

Fig. 5 Classifier wise area under curve

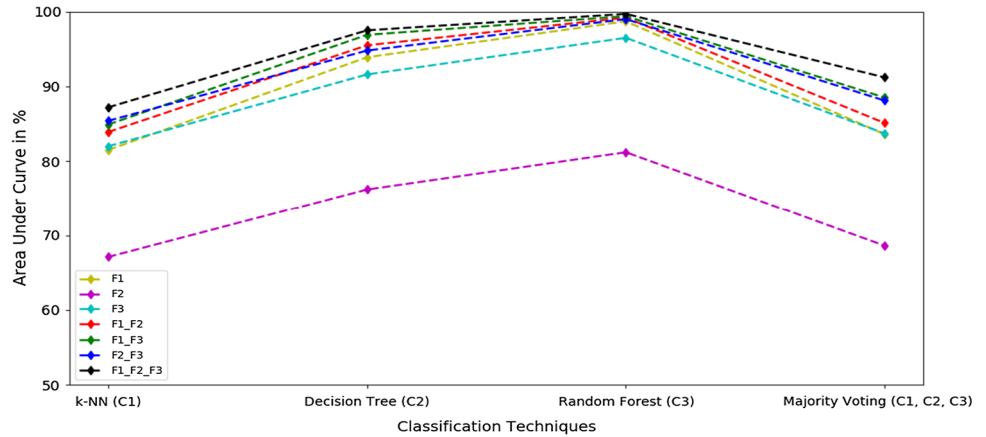


Table 5 Quantitative Comparison of Different Feature Extraction Techniques for Object Recognition using Adaptive Boosting

Feature extraction techniques	Adaptive boosting (Training set: 80% and testing set: 20%)			
	Accuracy	TPR	FPR	AUC
Shi-Tomasi (F1)	73.1%	73.1%	0.3%	98.7%
SIFT (F2)	38.3%	38.3%	0.6%	76.3%
SURF (F3)	66.1%	66.1%	0.7%	94.3%
Shi-Tomasi (F1) + SIFT (F2)	78.9%	78.9%	0.2%	99.2%
Shi-Tomasi (F1) + SURF (F3)	78.7%	78.7%	0.3%	99.3%
SIFT (F2) + SURF (F3)	78.1%	78.1%	0.3%	99.0%
Shi-Tomasi (F1) + SIFT (F2) + SURF (F3)	86.4%	86.4%	0.2%	99.7%

Table 6 Quantitative Comparison of Different Feature Extraction Techniques for Object Recognition using Bootstrap Aggregating

Feature extraction techniques	Bootstrap aggregating (Training set: 80% and testing set: 20%)			
	Accuracy	TPR	FPR	AUC
Shi-Tomasi (F1)	72.7%	72.7%	0.3%	98.8%
SIFT (F2)	39.7%	39.7%	0.6%	84.9%
SURF (F3)	66.2%	66.2%	0.8%	97.1%
Shi-Tomasi (F1) + SIFT (F2)	78.4%	78.4%	0.2%	99.1%
Shi-Tomasi (F1) + SURF (F3)	76.9%	76.9%	0.3%	99.4%
SIFT (F2) + SURF (F3)	77.5%	77.5%	0.3%	99.0%
Shi-Tomasi (F1) + SIFT (F2) + SURF (F3)	85.5%	85.6%	0.2%	99.7%

recognition accuracy as shown in Table 1, true positive rate (TPR) as shown in Table 2, false positive rate (FPR) as shown in Table 3 and area under curve (AUC) as shown in Table 4. These results are also graphically depicted in Figs. 2, 3, 4 and 5. The results computed using adaptive boosting are shown in Table 5, and the results computed using bootstrap aggregation are shown in Table 6. After evaluating the results in all cases, we determined that the random forest classifier performs better with 85.9% recognition accuracy than other classifiers. The results shown after applying adaptive boosting are also better for

the combination of Shi-Tomasi, SIFT and SURF feature detector and descriptor algorithm.

7 Conclusion and future work

This paper proposed a combination of feature detectors and descriptors that made the object recognition task easier and faster. This experiment is implemented on a large public dataset Caltech101 which is still a very challenging dataset in the object recognition problem. The features extracted using Shi-Tomasi, SIFT and SURF, are classified using

various classifiers, i.e., K-NN, decision tree and random forest. To reduce the size of features, k-means clustering and locality preserving projection (LPP) are also used. This approach increased the speed of the object recognition process. Adaptive boosting and bootstrap aggregation are used to improve the performance of object recognition. Eighty percent of images are used for training phase, and remaining 20% are used for testing. Among three classifiers, random forest classifier outperforms with 85.9% recognition accuracy and 86.4% using the adaptive boosting scheme. This proposed methodology can be useful for other different applications of image processing and computer vision areas.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest. For experimental work, authors have considered Caltech-101 image dataset in this paper. This dataset comprises of 101 object classes.

References

- Anupama HS, Cauvery NK, Lingaraju GM (2015) k-NN based object recognition system using brain computer interface. *Int J Comp Appl* 120(2):35–38
- Azad P, Asfour T and Dillmann R (2009). Combining Harris interest points and the sift descriptor for fast scale-invariant object recognition. In: Proceedings of the International Conference on Intelligent Robots and Systems, pp 4275–4280.
- Bansal M, Kumar M, Kumar M (2020) 2D Object recognition techniques: state-of-the-art work. *Arch Comput Methods Eng.* <https://doi.org/10.1007/s11831-020-09409-1>
- Bansal M, Kumar M and Kumar M (2020b). XGBoost: Object Recognition Using Shape Descriptors and Extreme Gradient Boosting Classifier. https://doi.org/10.1007/978-981-15-6876-3_16.
- Bay H, Tuytelaars T and Gool LV (2006). Surf: Speeded up robust features. In: Proceedings of the European Conference on Computer Vision, pp 404–417.
- Bosch A, Zisserman A and Munoz X (2007). Image classification using random forests and ferns. In: Proceedings of the IEEE 11th International Conference on Computer Vision, 1–8.
- Du W and Zhan Z (2002). Building decision tree classifier on private data. In: Proceedings of the IEEE International Conference on Privacy, Security and Data Mining, 14:1–8
- Fularz M, Kraft M (2015) Hardware implementation of a decision tree classifier for object recognition applications. *Measure Autom Monitor* 61(7):379–381
- Gall J, Razavi N and Gool LV (2012) An introduction to random forests for multi-class object detection. In: Proceedings of the Outdoor and Large-Scale Real-World Scene Analysis, pp 243–263.
- Galleguillos C, Rabinovich A and Belongie S (2008). Object categorization using co-occurrence, location and appearance. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1–8
- Guo GD and Zhang HJ (2001). Boosting for fast face recognition. In: Proceedings IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, p p96–100.
- Gupta S, Kumar M, Garg A (2019) Improved object recognition results using SIFT and ORB feature detector. *Multimed Tools Appl* 78(34157–34171):2019
- Harris CG and Stephens M (1988). A combined corner and edge detector. In: Proceedings of the Alvey Vision Conference, 15(50):10–52
- Hauska H and Swain PH (1975). The decision tree classifier: design and potential. In: Proceedings of the Symposium on Machine Processing of Remotely Sensed Data, pp 38–48.
- Kang M and Kim J (2007). Real time object recognition using K-nearest neighbor in parametric Eigen space. In: Proceedings of International Conference on Life System Modeling and Simulation, pp 403–411.
- CS Kenney, M Zuliani, BS Manjunath 2005 An axiomatic approach to corner detection. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 1: 191-197
- Kim J, Kim BS and Savarese S (2012). Comparing image classification methods: K-nearest-neighbor and support-vector-machines. In: Proceedings of the Applied Mathematics in Electrical and Computer Engineering, pp 133–138.
- Li W, Dong P, Xiao B, Zhou L (2016) Object recognition based on the region of interest and optimal bag of words model. *Neurocomputing* 172:271–280
- DG Lowe 1999 Object recognition from local scale-invariant features In: Proceedings of the International Conference on Computer Vision 99(2): 1150-1157
- McFee B, Galleguillos C, Lanckriet G (2011) Contextual object localization with multiple kernel nearest neighbor. *IEEE Trans Image Process* 20(2):570–585
- Muralidharan R (2014) Object recognition using k-nearest neighbor supported by Eigen value generated from the features of an image. *Int J Innov Res Comp Commun Eng* 2(8):5521–5528
- Murphy-Chutorian E and Triesch J (2005). January. Shared features for scalable appearance-based object recognition. In: Proceedings of the 7th IEEE Workshops on Applications of Computer Vision, 1(1):16–21.
- Schmidt A, Kraft M and Kasiński A (2010). An evaluation of image feature detectors and descriptors for robot navigation. In: Proceedings of the International Conference on Computer Vision and Graphics, pp 251–259.
- Shi J and Tomasi C (1994). Good Features to Track. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp 593–600.
- Welke K, Azad P and Dillmann R (2006). Fast and robust feature-based recognition of multiple objects. In: Proceedings of the 6th IEEE-RAS International Conference on Humanoid Robots, pp 264–269.
- Wu M, Ramakrishnan N, Lam SK and Srikanthan T (2012) Low-complexity pruning for accelerating corner detection. In: Proceedings of the 2012 IEEE International Symposium on Circuits and Systems, pp 1684–1687.