ORIGINAL RESEARCH

Transfer learning for image classification using VGG19: Caltech-101 image data set

Monika Bansal[1] · Munish Kumar[2] · Monika Sachdeva[3] · Ajay Mittal[4]

Abstract
Image classification is getting more attention in the area of computer vision. During the past few years, a lot of research has been done on image classification using classical machine learning and deep learning techniques. Presently, deep learning-based techniques have given stupendous results. The performance of a classification system depends on the quality of features extracted from an image. The better is the quality of extracted features, the more the accuracy will be. Although, numerous deep learning-based methods have shown enormous performance in image classification, still due to various challenges deep learning methods are not able to extract all the important information from the image. This results in a reduction in overall classification accuracy. The goal of the present research is to improve the image classification performance by combining the deep features extracted using popular deep convolutional neural network, VGG19, and various handcrafted feature extraction methods, i.e., SIFT, SURF, ORB, and Shi-Tomasi corner detector algorithm. Further, the extracted features from these methods are classified using various machine learning classification methods, i.e., Gaussian Naïve Bayes, Decision Tree, Random Forest, and eXtreme Gradient Boosting (XGBClassifier) classifier. The experiment is carried out on a benchmark dataset Caltech-101. The experimental results indicate that Random Forest using the combined features give 93.73% accuracy and outperforms other classifiers and methods proposed by other authors. The paper concludes that a single feature extractor whether shallow or deep is not enough to achieve satisfactory results. So, a combined approach using deep learning features and traditional handcrafted features is better for image classification.

Keywords ORB · SIFT · SURF · K-Means · LPP · PCA

✉ Monika Bansal
write2monikabansal@gmail.com

Munish Kumar
munishcse@gmail.com

Monika Sachdeva
monasach1975@gmail.com

Ajay Mittal
ajay_mittal825@yahoo.com; ajaymittal@pu.ac.in

[1] SSD Women Institute of Technology, Bathinda, Punjab, India

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

[3] Department of Computer Science and Engineering, I.K.G. Punjab Technical University, Mohali Campus-1, Mohali, India

[4] Department of Computer Science and Engineering, University Institute of Engineering and Technology, Panjab University, Chandigarh, India

1 Introduction

Image classification is considered as the main research topic in the area of computer vision and artificial intelligence. Image classification works on correctly identifying an object in an image. Earlier, various machine learning algorithms were used to solve this problem. Various handcrafted feature extraction methods were adopted to acquire the features from the image. The features used for image classification may be local, global, or both. Then, single or ensemble machine learning classification algorithms are employed to classify the images based on color, shape, texture, or some other feature. In the current era, the deep learning has given outstanding results in all the applications of computer vision like image classification, object detection, security, image processing, etc. Deep learning is a subset of machine learning. In the deep learning approach, both feature extraction and classification are done automatically to classify the images having similar objects. There is no

need for the researchers to perform both the tasks manually as done in classical machine learning. Deep learning uses a collection of various neural layers to process a huge amount of data. So, it is also known as a Deep Convolutional Neural Network (DeepCNN).

This system is modeled on the architecture of the human brain. Just like the human brain that functions on a mesh of neurons, deep learning processes the data through the network of neural layers, filters outliners, spots familiar entities, and produces the final output i.e., the label of the object. A description of the functioning of classical machine learning and deep learning for image classification is depicted in Fig. 1.

However, recognizing the correct class of a given object is very challenging due to the low resolution of the image, inadequate extraction of local and/or global features, geometric variation, etc. While considering these issues, the paper reveals that no single feature extraction algorithm can classify a wide range of images accurately. The Caltech-101 dataset is considered for the experiment as it is one of the most challenging datasets. This dataset contains 101 classes and 1 background scene class. Each class has 40–800 images and this dataset has a total of 9146 images. In the paper, a supervised learning approach is adopted where feature extraction through a pre-trained deep learning model (VGG19 model) is done and these features are further classified using various state-of-art classifiers (Naïve Bayes, Decision tree, Random Forest, XGBClassifier). These days, the VGG19 model has shown good performance for image classification. But the experimental results show that even this model is not enough for accurate image classification. So, a fusion of deep features (pre-trained VGG19) and various state-of-art handcrafted features extraction algorithms (SIFT, SURF, ORB, and Shi-Tomasi corner detector) is experimented in the paper to provide very high recognition rates on the Caltech-101 dataset. The combined feature vector is further classified using various state-of-art machine learning classification algorithms (Gaussian Naïve Bayes, Decision tree, Random Forest, XGBClassifier). A standard data-partitioning strategy is followed for the study in which 70% of the images of each class are considered in the training dataset and the rest 30% are used in the testing dataset. The experiment proved that the fusion of the above five

feature extraction methods with the Random Forest classifier outperforms with high recognition accuracy, precision, recall, the area under curve (AUC), and low false-positive rate, root means squared error, and CPU time. The proposed fusion feature extraction system achieves 93.73% recognition accuracy which outperforms the approaches given by many researchers. The paper exhibits all the performance measure outcomes using the proposed approach as Precision (93.70%), Recall (93.73%), F1_score (93.22%), Area Under Curve (96.79%), False Positive Rate (0.15%), Root Mean Square Error (20.05%), Average CPU Time (0.39 min).

The rest of the paper is organized as. In Sect. 2, the problem that occurred in the use of the deep neural network is described for image classification. Section 3 lists the related work done by various authors. Sections 4, 5 describes the feature extraction algorithms used in the experiment. Section 6 mentions the machine learning classification algorithms used in the experiment. In Sect. 7, the techniques used in the proposed system are explained. The results based on experiments are demonstrated in Sect. 8 and the whole paper is concluded in Sect. 9.

2 Challenges

Over the past few years, deep Convolutional Neural Network (CNN) has revealed tremendous results in the area of computer vision. But still, the researchers are facing many challenges to execute the CNN model. The proposed system is implemented to resolve the issues that have arisen for the image classification task using a deep neural network. The first challenge is to design a network model. CNN is designed with many layers, so they require millions of parameters to learn during the training phase. Designing a CNN model from the scratch demands a few resources for the execution, such as a large memory capacity, a fast processor, a huge dataset, enormous power consumption, etc. Deep learning needs an extremely large memory capacity as deep learning extracts a huge amount of data during the feature extraction phase. Basically, deep learning evaluates the value for each pixel of the image using various mathematical operations. Deep learning takes a lot of time for the computation (can be many hours or many days) depending

Fig. 1 Machine learning vs. deep learning
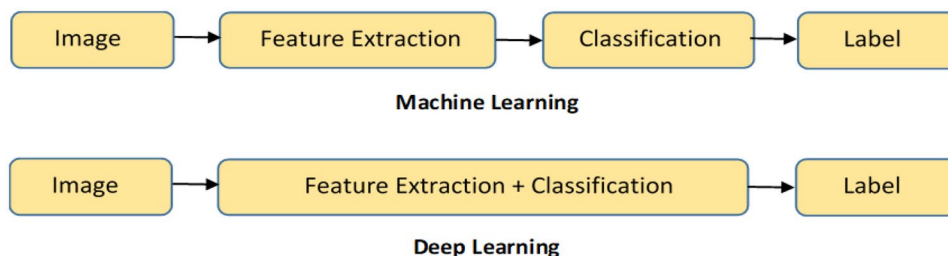
on the computational capabilities of the hardware. So, power backup is required to make it a continuous process. Deep learning algorithms cannot be implemented on the general CPU system rather they need GPUs and TPUs enabled systems. These systems are very expensive and are not easily affordable. Deep learning works well with a large collection of data. The accuracy depends on the size of data which is very difficult to assemble in the real world. Even it makes the use of data augmentation to consider the various aspects of the image and to increase the size of the dataset, but still, it does not help to achieve the satisfactory results.

In image classification, deep learning shows adequate results in high-resolution images. It uses various pre-processing steps before feature extraction, but still, it is not able to extract the accurate global features of the image. Several state-of-art deep learning methods are highly sensitive to translation, scaling, and rotation. Data augmentation has resolved this issue in the neural networks to some extent, but this increases the size of the dataset that will again need more storage capacity and computation time. Keeping these issues in view, there is still a demand for handcrafted feature extraction methods. Deep learning extracts low-level features that help to acquire the best results, but these are not enough for image classification. Therefore, the proposed system uses a fusion of features extracted using a pre-trained model of deep learning, i.e., VGG19, and various handcrafted feature extraction algorithms, i.e., SIFT, SURF, ORB, and Shi-Tomasi corner detector for image classification.

3 Related work

Kataoka et al. (2015) demonstrated a description of the feature evaluation on various deep learning networks for object recognition and detection. They experimented that VGGNet architecture performed over AlexNet architecture. Further, they carried out feature tuning by concatenating some layers of both the architectures and transformed them using Principal Component Analysis (PCA). Caltech101 and DaimlerPedestrian Benchmark Datasets are used for the experiment and achieved 91.8% accuracy. Mahmood et al. (2017) presented a hybrid approach for image classification where the ResNet model is used for feature extraction and then extracted features are fine-tuned using PCA-SVM for image classification. Four datasets are taken for experiments are MIT-67, MLC, Caltech-101, and Caltech-256. The model was trained by using 30 images from each class and outperforms other methods. Ren et al. (2017) implemented a combined approach for image classification where features are acquired using Convolutional Neural Network (CNN) architecture and eXtreme Gradient Boost (XGBClassifier) Classifier for recognition of the image. The experiment was

implemented on MNIST and CIFAR-10 dataset and proved the best results.

Srivastava et al. (2017) proposed an ensemble of local and deep features for image classification. They compared various pre-trained convolutional neural networks for feature extraction. A combined feature extraction approach is followed using SIFT and various pre-trained neural networks. The proposed model is trained using an SVM classifier that is followed by a majority voting scheme to recognize the image. The model is evaluated on the CIFAR-10 dataset and achieved 91.8% accuracy. Shaha and Pawar (2018) proposed a fusion of the deep learning model (VGG19) for feature extraction and support vector machine (SVM) for image classification. They compared different neural models, i.e., AlexNet, VGG16, VGG19 for feature extraction and fine-tuned these models over GHIM10K and Caltech256 datasets for image classification. VGG19 architecture showed better performance results over AlexNet and VGG16 that are represented by using three evaluation parameters, i.e., precision, recall, and F-score. Mingyuan and Wang (2019) used the CNN model for feature extraction and presented a comparative analysis among various classification algorithms—CNN, SVM, RF, DT, KNN, NB and GBDT for image classification. Pandey et al. (2018) proposed Common Sense Knowledge (CSK) by embedding three deep learning models using CNN, R-CNN and R-FCN for object detection. The experiment has been conducted to aid smart mobility. Singh and Singh (2019) presented a fusion of various handcrafted features for image classification. They made a comparative analysis of the proposed work over a deep neural network (DNN) i.e., AlexNet and achieved high accuracy. They also exhibited various challenges of image classification that cannot be solved with the AlexNet model. The experiment was taken on five dataset- PASCAL VOC2005, Soccer, SIMPLIcity, Flower, and Caltech-101. Yadav (2019) evaluated the performance of the CNN based model using VGG16 and inception over the traditional image classification model using ORB and SVM. The experiment has been conducted on various medical images. Transfer learning is used to improve the accuracy of the image classification. The experiment using transfer learning achieved the best results on chest X-ray images.

Garg et al. (2020) proposed an object detection system, named, CK-SNIFFER to automatically identifies a large number of errors based on common sense knowledge. Karthikeyan et al. (2020) investigated the transfer learning approach on a huge dataset of X-ray images from patients with common bacterial pneumonia, confirmed COVID-19 cases and healthy cases with three pre-trained models— VGG16, VGG19 and RestNet101. They achieved the best results with the proposed approach. Talaat et al. (2020) proposed an improved hybrid approach for image classification using CNN for feature extraction and swarm-based

feature selection algorithm (Marine Predators algorithm) to select the relevant features. Liu et al. (2020) developed a deep learning model for automatic multiclass pest detection using global and local activated feature pyramids. The approach was designed using two stages CNN-based pest detection and classification pipeline. In the first stage, a global activation feature pyramid network (GaFPN) was introduced which was aggregated on each convolutional block in order to screen and activate depth and spatial information from feature maps outputed by each block. In the second stage, the feature map created from stage one was used to design a local activated feature pyramid network (LaFPN) which was adopted for pest classification and position regression. Further, final localization and classification were done using various fully connected layers. The experiment was conducted on their own pest dataset and the model was compared with Faster R-CNN and FPN.

Kumar et al. (2021) analyzed the performance variations of Deep Learning (DL) and Classical Machine Learning (CML) classifiers with different feature vector representations and proposed an ensemble approach for classification using DL and CML. The aim of the experiment was to improve the performance of single models. Seemendra et al. (2021) analyzed various pre-trained CNN models with fine-tuning to detect and classify invasion ductal carcinoma. The models used were VGG16, VGG19, ResNet, DenseNet, MobileNet and EfficientNet. The authors achieved the best results using fine-tuned VGG19 by 93.05% as sensitivity and 94.46% as Precision which was higher among other ones. The experimented was conducted on about 90,000 images.

4 Feature extraction algorithms

Feature extraction plays the most significant role in image classification. The performance of the classification task highly depends on the crucial features of the images. The features of an object are classified into local and global features based on color, shape, or texture. Color and texture features are considered as local features and shape as global features. In this paper, deep features and handcrafted features are extracted for image classification. Deep features are extracted from a pre-trained deep neural network (i.e., VGG19). The deep model extracts both the local and global features of an image. SIFT, SURF, ORB, and Shi-Tomasi corner detectors are employed to extract shape features. A brief description of each method is mentioned as follows:

4.1 VGG19

VGG19 proposed by Simonyan and Zisserman (2014) is a convolutional neural network that comprises 19 layers with 16 convolution layers and 3 fully connected to classify the images into 1000 object categories. VGG19 is trained on the ImageNet database that contains a million images of 1000 categories. It is a very popular method for image classification due to the use of multiple 3×3 filters in each convolutional layer. The architecture of VGG19 is shown in Fig. 2a. This shows that 16 convolutional layers are used for feature extraction and the next 3 layers work for classification. The layers used for feature extraction are segregated into 5 groups where each group is followed by a max-pooling layer. An image of size 224×224 is inputted

(a) Architecture of VGG19 model

(b) Ensemble of deep feature extraction using VGG19 model and machine learning classification
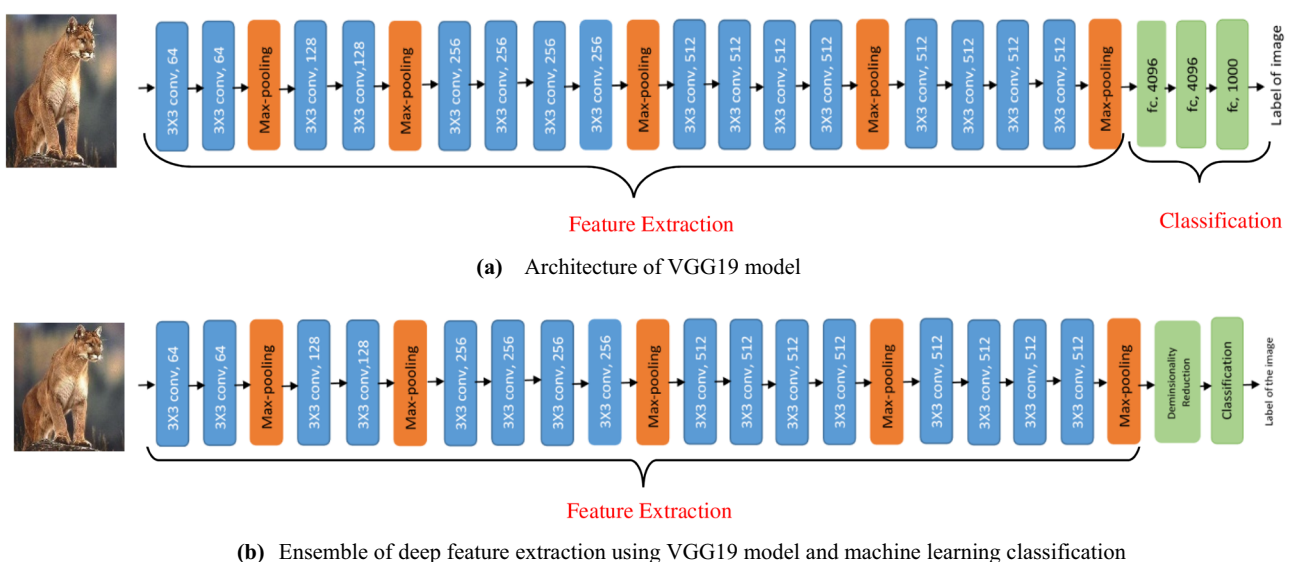
Fig. 2 **a** Architecture of VGG19 model. **b** Ensemble of deep feature extraction using VGG19 model and machine learning classification

into this model and the model outputs the label of the object in the image. In the paper, features are extracted through a pre-trained VGG19 model, but for classification, various machine learning approach is followed. As the CNN model computes huge parameters after feature extraction, there is a need for dimensionality reduction to minimize the size of the feature vector as shown in Fig. 2b. The dimensionality reduction is done with Locality Preserving Projection that is followed by a classification method.

4.2 Scale invariant feature transform (SIFT)

SIFT is one of the most widely used shape feature extraction algorithm. The algorithm is a key point detector and descriptor algorithm proposed by Lowe (2004) to extract key interest points from the image. It is highly robust towards the orientation and scaling of an image. It is invariant to illumination changes. It extracts the maximum interest points (features) even from low-resolution images. SIFT extracts 128 features from an image through a filtering approach which functions in four stages. The first stage detects the important locations from the image using the Difference-of-Gaussian (DoG) algorithm. Then localization is performed to determine the important features. This is followed by the computation of directions of gradients that makes the algorithm invariant to rotation. In the last stage, the computed keypoints are converted into a feature vector of size 128.

4.3 Speed up robust features (SURF)

SURF is a variant of the SIFT algorithm that is used as the keypoints detector and descriptor. It is developed by Bay et al. (2006). The interest points of an image are detected by approximating the Laplacian-of-Gaussian (LoG) with a Box filter. These detected keypoints are represented with the Hessian matrix. SURF is more invariant to geometric and photometric displacement deformation. SURF creates a feature vector of 64 or 128 dimensions.

4.4 Orient fast and rotated brief (ORB)

ORB algorithm is a local feature extraction algorithm that is presented by Rublee et al. (2011). The ORB uses a pyramid scheme with a FAST keypoint detector and a BRIEF keypoint descriptor that is followed by a Harris corner detector (Harris and Stephens 1988). This algorithm is faster than SIFT and SURF. It is also robust to noise, scale, rotation, translation.

4.5 Shi Tomasi corner detector

This algorithm is a variation of the Harris Corner Detector algorithm where a slight change is done in the selection

criteria. Shi and Tomasi (1994) presented this approach that helps in better corner detection than Harris 'algorithm and achieves better accuracy. Corners are used as global features of the image that identify the shape of the object which aids in the object recognition task.

5 Feature dimension reduction techniques

The following techniques are employed to select the important features from the large set of features (Varde et al. 2007) and to diminish the dimensions of the feature vector that are obtained using above mentioned feature extraction methods as the large size of the feature vector will cause the problem of overfitting.

5.1 k-means clustering

k-means clustering is a distance-based algorithm where distance is evaluated between the centroid of the cluster and the key descriptors of the object using Euclidean distance or max–min method. k-means clustering follows a number of steps as:

1. k is used as the number of clusters that is to be chosen randomly.
2. k descriptors are selected randomly from a set of n descriptors of an object as centroids.
3. All key descriptors are assigned to the closest cluster centroid.
4. Cluster centroid is recomputed from the newly formed clusters.
5. The process of updating cluster centroid goes on till further there is no change in centroid.
6. Finally, k clusters are obtained according to closest points and the mean of each cluster is computed. The resultant k values are used as a reduced feature vector.

5.2 Locality preserving projection

Locality preserving projection (LPP) is a linear dimensionality reduction algorithm. LPP retains the local neighborhood information of the data set by discarding undesired data. LPP operates in three steps as follows.

1. An adjacency graph is constructed by placing an edge between nodes i and j where the distance between these two nodes is very less.
2. Weights are chosen for each edge using two variations—Heat kernel and Simple-minded.
3. Finally, an Eigenmaps is designed by computing eigenvectors and eigenvalues.

The outcome of the LPP method lessens the size of the feature vector by considering the important ones and discarding unused data points.

6 Classification techniques

Various state-of-art classification methods are used for image classification. Each method has its own merits and demerits. Some methods work very fast while some present more accuracy. In the paper, we have analyzed the results of image classification using various well-known classification methods- Decision Tree, Gaussian Naïve Bayes, Random Forest, and XGB Classifier. These methods are described as follows.

6.1 Gaussian Naïve Bayes

Gaussian Naïve Bayes is an extended version of the Naïve Bayes algorithm that adopted the Probability approach. In Naïve Bayes, prediction of test data is computed using the distribution of data wherein Gaussian Naïve Bayes, prediction of the test data given a class is obtained from Gaussian distribution by employing the mean and standard deviation of the data. Gaussian Naïve Bayes is the simple and most popular probability-based approach used for image classification.

6.2 Decision Tree

The decision tree algorithm is proposed by Quinlan in 1986. It is a tree-based approach used for classification where all the features considered are placed at the root. Similar features are grouped in one category and taken as the nodes. The decision tree is recursive in nature as these nodes are further subdivided into various nodes representing similar features. The process of splitting the trained data into nodes continues till there is no further division possible. The classes are represented by the tree leaves. This algorithm has many advantages as it is very simple to understand, interpret. It is very fast to execute and shows better accuracy for image classification. But there is a problem of overfitting in the decision tree. It can create over-complex trees which do not generalize the data well.

6.3 Random Forest

Random Forest algorithm (developed by Kleinberg in 1996) is an ensemble tree that comprises many decision trees. In this algorithm, various decision trees are made on subsamples of the trained data, and averaging of all the results is computed to obtain more predictive accuracy. Random Forest has also solved the problem of overfitting. This algorithm

has shown better outcomes as compared to the decision tree but still, it has some drawbacks. This method is difficult to interpret, and the time taken by it is more as compared to the decision tree.

6.4 XGB Classifier

XGB Classifier stands for eXtreme Gradient Boosting Classifier which is a boosting algorithm based on Gradient Boosting Classifier. This method is proposed by Chen and Guestrin (2016). XGB Classifier is an ensemble classifier that uses the regularization technique to reduce the problem of overfitting. This method outperforms a Gradient Boosting Algorithm but compared to the above-mentioned classifiers, this takes more time to classify the data. Recently, this method has gained very popularity. It boosts the performance of an ensemble classification algorithm by making the stronger model from numerous weaker models using an iterative approach.

7 Proposed methodology

The architecture of the proposed system is depicted in Fig. 3. The proposed method is based on a combination of deep learning features and traditionally handcrafted feature extraction algorithms. The experiment analyzed the performance of the image classification system with deep learning features and ensemble of deep features and various traditional handcrafted feature extraction methods. The proposed system is used to represent that rather deep learning has gained worldwide popularity, but still, it does not fully support the system of image classification on the Caltech-101 dataset. The proposed model works in two phases: feature extraction and image classification.

The first phase of feature extraction consists of three components to create a feature vector:

1. Using the pre-trained model in Keras i.e., VGG19 and various handcrafted methods in OpenCV i.e., SIFT, SURF, ORB, and Shi_Tomasi corner detector to extract the features of images.
2. Using k-means clustering in OpenCV to select the important features and obtain a 64-dimensional feature vector for every descriptor.
3. Using Locality Preserving Projection to diminish the feature vector of size 64 into 8 components.

During the first phase, a combined feature vector having a total of 40 features is computed that is followed by a classification task.

The second phase is Image Classification where the performance of the recognition system is evaluated after
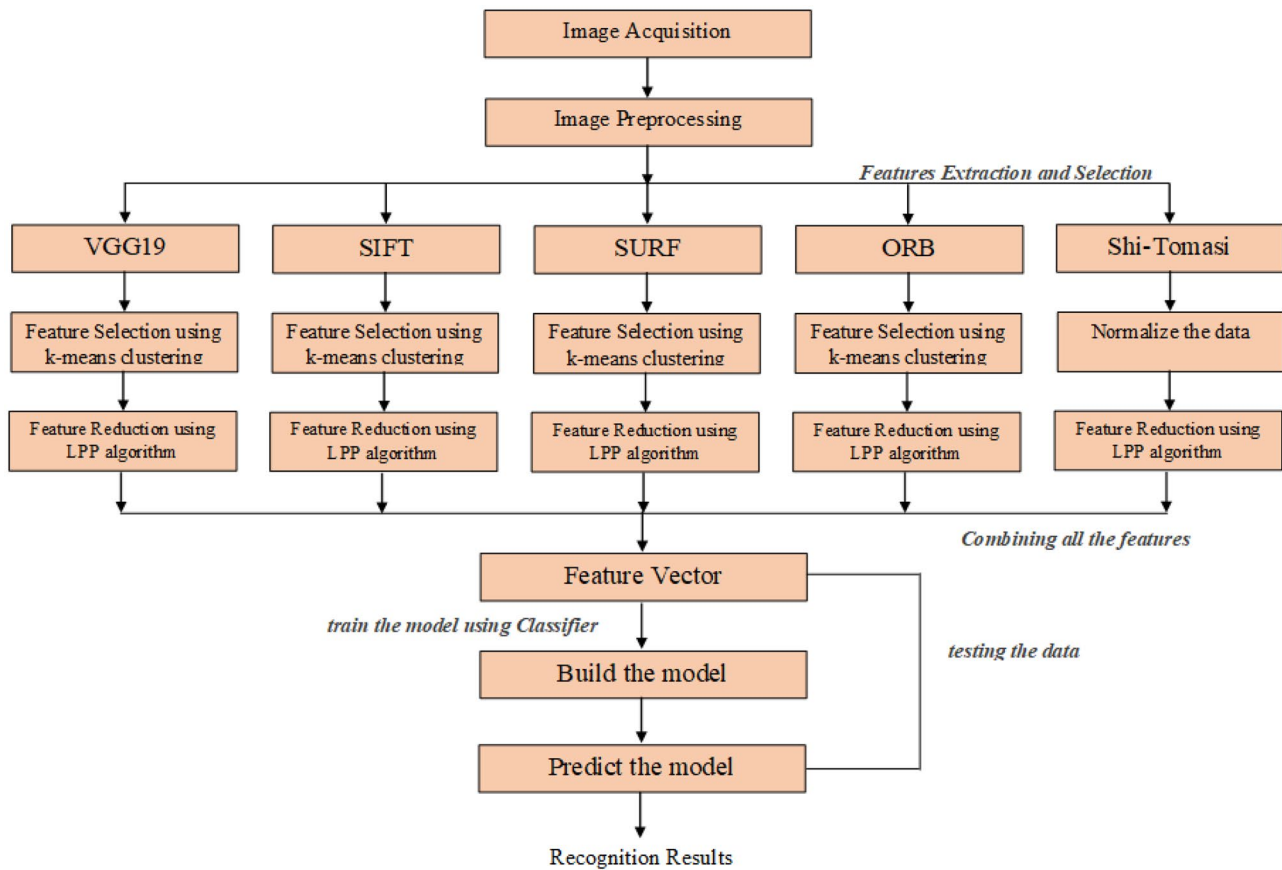
Fig. 3 Architecture of the proposed system

applying various machine learning classification algorithms, i.e., Gaussian Naïve Bayes, Decision Tree, Random Forest, and XGB Classifier. During this process, a model is built using standard data partitioning strategy (i.e., 70:30) where 70% of the images of each class is used for training purpose and remaining 30% of the images are used for testing the images for recognition. The performance of the model is predicted on the test dataset.

Table 1 A representation of labelling for various feature extraction methods

Label	Feature extraction method
F1	VGG19
F2	SIFT
F3	SURF
F4	ORB
F5	Shi-Tomasi

8 Experimental results

This section discusses the evaluation results on the Caltech-101 dataset. Caltech-101 is one of the most challenging multiclass datasets for the image classification problem. It consists of 102 categories where it comprises total of 9146 images. Out of 102 categories, one category is of Background scene which is not used in the experiment. So, the experiment is done on 101 categories having 8678 images. The dataset is unbalanced as each category contains a different number of images nearly 40–800 images.

The images in the dataset have low resolution and are noisy. Table 1 shows the labels used for various feature extraction algorithms. Various performance measures are used to represent the comparison among these feature extraction methods for image classification. Due to the multi-class dataset, macro-averaging has been adopted in the experiment as macro-averaging estimates the performance by averaging the predictive results of each class. The experiments are analyzed using eight parameters, i.e., accuracy, precision, recall, F1-score, False Positive Rate (FPR), Area Under

Curve (AUC), root mean square error (RMSE), and CPU execution time. A standard data partitioning methodology is adopted for the experiment in which 70% images from each class are considered in the training data and the rest 30% of images from all the classes are used for the analysis of the proposed system as test data. The experiment also demonstrates a comparative analysis, among various state-of-art classifiers i.e., Gaussian Naïve Bayes, Decision Tree, Random Forest, and XGB Classifier.

In Table 2, a comparison between various feature extraction methods is represented using recognition accuracy. These comparative results have been graphically presented using Fig. 4 which clearly describes the performance of the proposed system. Table 3 shows the comparison using Precision, Table 4 presents False Positive Rate (FPR), Table 5 presents Root Mean Square Error (RMSE), and Table 6

presents average CPU time which depicts the average execution time of object recognition increases with the number of features. All the tables witness the improvement in all performance measures due to the ensemble of VGG19 and all handcrafted methods. Table 7 demonstrates a detailed comparison among various classifiers on all the performance parameters. This shows that the proposed combined feature vector is more advantageous than a single feature extraction method. All the experiments have been performed on a machine with Microsoft Windows 10 Operating System (original) and Intel Core i3 processor with 4 GB RAM.

Recently, various researchers have analyzed various ensemble approaches for image classification due to the improvement in accuracy results. Table 8 shows a comparative analysis of the proposed system with some recent experiments on the Caltech-101 dataset. Through the comparison, it is observed

Table 2 Quantitative comparison among deep learning, various handcrafted and ensemble feature extraction methods (Classifier wise recognition accuracy (in %))

Features	Gaussian Naïve Bayes	Decision tree	Random forest	XGB Classifier
F1	55.37	54.49	57.47	63.13
F1 + F2	67.65	68.51	70.31	73.02
F1 + F3	64.69	63.52	66.79	71.59
F1 + F4	71.84	72.62	75.08	78.29
F1 + F5	70.37	70.93	72.88	76.64
F1 + F2 + F3	75.04	77.29	76.63	80.38
F1 + F2 + F4	80.71	80.93	82.24	83.74
F1 + F2 + F5	78.01	78.07	79.48	82.76
F1 + F3 + F4	78.46	80.58	81.01	84.16
F1 + F3 + F5	75.03	77.05	77.31	83.00
F1 + F4 + F5	81.77	83.68	84.88	84.55
F1 + F2 + F3 + F4	86.32	88.47	89.65	88.86
F1 + F2 + F3 + F5	82.69	83.90	84.00	87.87
F1 + F2 + F4 + F5	89.16	89.29	90.23	90.35
F1 + F3 + F4 + F5	85.63	88.33	89.42	88.84
F1 + F2 + F3 + F4 + F5	92.05	92.67	**93.73**	93.02

Bold face of text depicting the maximum accuracy achieved in each table

Fig. 4 Comparison of recognition accuracy achieved using various combinations of feature extraction methods
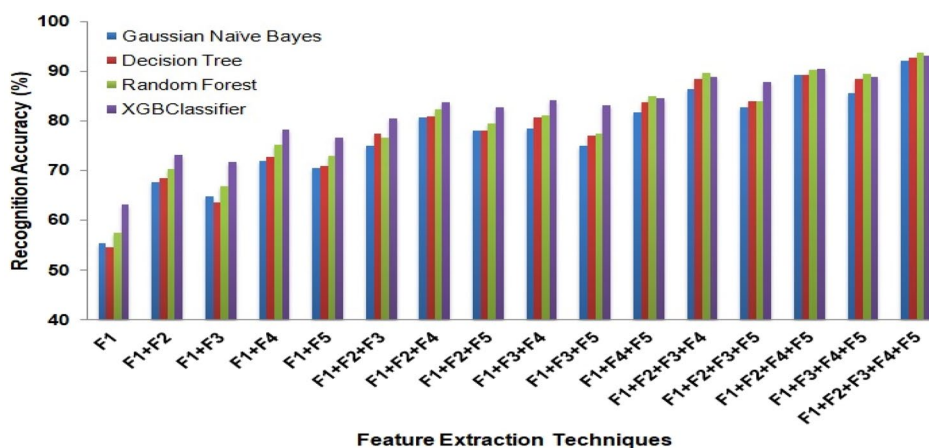
Table 3 Quantitative comparison among deep learning, various handcrafted and ensemble feature extraction methods (Classifier wise precision (in %))

Features	Gaussian Naïve Bayes	Decision Tree	Random Forest	XGB Classifier
F1	53.47	53.67	57.51	61.62
F1 + F2	66.13	67.86	68.96	71.40
F1 + F3	63.15	62.91	64.85	69.63
F1 + F4	70.68	72.04	73.35	77.09
F1 + F5	68.75	70.69	72.28	75.59
F1 + F2 + F3	74.46	76.64	76.63	79.70
F1 + F2 + F4	80.37	80.97	81.78	83.58
F1 + F2 + F5	77.05	77.56	78.18	82.22
F1 + F3 + F4	77.56	79.70	79.93	83.63
F1 + F3 + F5	73.85	76.54	76.80	82.57
F1 + F4 + F5	80.92	83.58	84.17	83.61
F1 + F2 + F3 + F4	86.16	88.63	90.13	88.99
F1 + F2 + F3 + F5	81.89	83.36	83.24	87.96
F1 + F2 + F4 + F5	89.42	89.50	90.38	90.47
F1 + F3 + F4 + F5	85.17	87.86	89.16	88.77
F1 + F2 + F3 + F4 + F5	91.90	92.66	**93.70**	93.10

Bold face of text depicting the maximum accuracy achieved in each table

Table 4 Quantitative comparison among deep learning, various handcrafted and ensemble feature extraction methods (Classifier wise false positive rate (in %))

Features	Gaussian Naïve Bayes	Decision Tree	Random Forest	XGB Classifier
F1	0.56	0.55	0.50	0.50
F1 + F2	0.46	0.43	0.41	0.41
F1 + F3	0.43	0.43	0.39	0.38
F1 + F4	0.36	0.33	0.30	0.29
F1 + F5	0.37	0.37	0.34	0.32
F1 + F2 + F3	0.34	0.31	0.31	0.29
F1 + F2 + F4	0.29	0.28	0.26	0.24
F1 + F2 + F5	0.29	0.29	0.28	0.26
F1 + F3 + F4	0.31	0.28	0.27	0.24
F1 + F3 + F5	0.33	0.31	0.30	0.26
F1 + F4 + F5	0.27	0.25	0.24	0.23
F1 + F2 + F3 + F4	0.24	0.21	0.20	0.20
F1 + F2 + F3 + F5	0.25	0.24	0.24	0.21
F1 + F2 + F4 + F5	0.20	0.20	0.19	0.17
F1 + F3 + F4 + F5	0.23	0.21	0.19	0.20
F1 + F2 + F3 + F4 + F5	0.18	0.17	**0.15**	**0.15**

Bold face of text depicting the maximum accuracy achieved in each table

Table 5 Quantitative comparison among deep learning, various handcrafted and ensemble feature extraction methods (Classifier wise root mean square error (in %))

Features	Gaussian Naïve Bayes	Decision Tree	Random Forest	XGB Classifier
F1	31.21	31.24	29.38	30.49
F1 + F2	27.11	26.76	25.63	25.96
F1 + F3	29.96	29.84	28.38	28.64
F1 + F4	25.78	24.75	23.35	22.71
F1 + F5	25.87	26.62	26.04	24.87
F1 + F2 + F3	25.99	24.95	24.61	24.72
F1 + F2 + F4	24.26	23.67	22.88	21.88
F1 + F2 + F5	24.14	23.88	23.66	23.66
F1 + F3 + F4	25.77	24.47	23.88	22.92
F1 + F3 + F5	25.93	25.66	25.12	24.41
F1 + F4 + F5	22.78	22.38	21.42	21.20
F1 + F2 + F3 + F4	22.84	21.63	21.76	21.32
F1 + F2 + F3 + F5	23.43	22.65	22.87	22.45
F1 + F2 + F4 + F5	21.32	21.41	20.51	**19.60**
F1 + F3 + F4 + F5	22.14	21.56	20.72	21.15
F1 + F2 + F3 + F4 + F5	21.01	20.92	20.05	20.01

Bold face of text depicting the maximum accuracy achieved in each table

Table 6 Quantitative comparison among number of features (Classifier wise average execution (CPU) Time (in seconds))

Number of Features	Gaussian Naïve Bayes	Decision Tree	Random Forest	XGB Classifier
1	0.00	0.01	0.22	2.39
2	0.00	0.01	0.29	3.84
3	0.00	0.02	0.30	5.32
4	0.00	0.02	0.34	6.54
5	0.00	0.02	0.39	7.66

that accuracy achieved by Mahmood et al. (2017) is higher but Singh and Singh (2019) in their paper has experimented that the recognition accuracy of the convolutional neural network decreases under rotation and scaling. Considering the methodology of Singh and Singh (2019), the authors will examine the

accuracy of the proposed system and Mahmood et al. (2017) under rotation and scaling conditions in their next experiment.

9 Conclusion

In this article, an analysis of various feature extraction techniques is discussed that includes a deep learning model (VGG19) and various handcrafted feature extraction methods, i.e., SIFT, SURF, ORB, and Shi-Tomasi corner detector algorithm. A survey on various classification methods, i.e., Gaussian Naïve Bayes, Decision Tree, Random Forest, and XGB Classifier is also conducted in the paper. The investigation indicates that the ensemble method for feature extraction performs better than a single feature extraction method. The results show that feature extraction using a popular method VGG19 is still not enough for image classification. The experiment

Table 7 Quantitative comparison analysis of all performance measurements

Performance measures	Gaussian Naïve Bayes	Decision Tree	Random Forest	XGB Classifier
Accuracy	92.05%	92.67%	93.73%	93.02%
Precision	91.90%	92.66%	93.70%	93.10%
Recall	92.05%	92.67%	93.73%	93.02%
False positive rate	0.18%	0.17%	0.15%	0.15%
F1-score	91.85%	92.54%	93.22%	92.84%
AUC	95.94%	96.25%	96.79%	96.43%
RMSE	21.01%	20.92%	20.05%	20.01%
Average execution time	0.00 min	0.02 min	0.39 min	7.66 min

Table 8 Performance evaluation on Caltech-101 dataset

Author	Year	Features	Technique used	Number of classes	Accuracy (%)	Time (min)
Mahmood et al	2017	ResNet-152	ResNet features with PCA-SVM classifier	101	94.7	
Rashid et al	2018	VGG16, AlexNet and SIFT	Hybrid of Deep CNN and SIFT Features along with entropy-controlled selection method and ensemble boosted tree	101	89.7	5.04
Singh et al	2019	Color Histogram (CH), Zernike Moments (ZMs), Gradient ZMs (GZMs), Multi-channel ZMs. (MZMs), Rotation Quaternion ZMs (RQZMs),	Fusion of these features with multi kernel learning (MKL) approach	10	84.60	0.08
Our system	2020	SIFT, SURF, ORB, Shi Tomasi	Fusion of these features with eXtreme Gradient Boosting Classifier	101	89.7	6.26
Proposed system	2020	VGG19, SIFT, SURF, ORB and Shi Tomasi corner detector	Fusion of these features with Random Forest Classifier	101	93.73	0.39

confirms that the proposed method is very powerful and consistently outperforms the methods proposed by other researchers. The paper has also notified on various challenges that occurred in the image classification task. This article will help other researchers to explore other combined approaches for image classification, also using various latest deep learning models.

Declarations

Conflict of interest The authors declare that they have no conflict of interest in this work. The authors have employed a public dataset, namely, Caltech-101 for performing the experiments in the considered work.

References

Bay H, Tuytelaars T, Van-Gool L (2006) Surf: speeded up robust features. In: Proceedings of the European conference on computer vision, pp 404–417

Chen T, Guestrin C (2016) XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp 785–794

Garg A, Tandon N, Varde A (2020) I am guessing you can't recognize this: generating adversarial images for object detection using spatial commonsense (student abstract). Proc AAAI Conf Artif Intell 34(10):13789–13790. https://doi.org/10.1609/aaai.v34i10.7166

Harris C, Stephens M (1988) A combined corner and edge detector. In: Proceedings of the Fourth Alvey vision conference, pp 147–151

Karthikeyan D, Varde AS, Wang W (2020) Transfer learning for decision support in Covid-19 detection from a few images in big data. IEEE Int Conf Big Data (big Data) 2020:4873–4881. https://doi.org/10.1109/BigData50022.2020.9377886

Kataoka H, Iwata K, Satoh Y (2015) Feature evaluation of deep convolutional neural networks for object recognition and detection. https://arxiv.org/abs/1509.07627

Kleinberg EM (1996) An overtraining-resistant stochastic modeling method for pattern recognition. Ann Stat 24(6):2319–2349

Kumar V, Recupero DR, Riboni D, Helaoui R (2021) Ensembling classical machine learning and deep learning approaches for morbidity identification from clinical notes. IEEE Access 9:7107–7126. https://doi.org/10.1109/ACCESS.2020.3043221

Liu L, Xie C, Wang R, Yang P, Sudirman S, Zhang J, Li R, Wang F (2020) Deep learning based automatic multi-class wild pest monitoring approach using hybrid global and local activated features. IEEE Trans Ind Inf 17(11):7589–7598

Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput vis 60(2):91–110

Mahmood A, Bennamoun M, An S, Sohel F (2017) Resfeats: residual network based features for image classification. In: 2017 IEEE International conference on image processing (ICIP). https://doi.org/10.1109/icip.2017.8296551

Mingyuan X, Wang Y (2019) Research on image classification model based on deep convolution neural network. EURASIP J Image Video Process 2019(1):1–11

Pandey A, Puri M, Varde A (2018) Object detection with neural models, deep learning and common sense to aid smart mobility. In: 2018 IEEE 30th international conference on tools with artificial intelligence (ICTAI), pp 859–863. https://doi.org/10.1109/ICTAI.2018.00134

Ren X, Guo H, Li S, Wang S, Li J (2017) A novel image classification method with CNN-XGBoost model. Lect Notes Comput Sci. https://doi.org/10.1007/978-3-319-64185-0_28

Rublee E, Rabaud V, Konolige K, Bradski GR (2011) ORB: an efficient alternative to SIFT or SURF. Int Conf Comput vis 11(1):2

Seemendra A, Singh R, Singh S (2021) Breast cancer classification using transfer learning. In: Evolving Technologies for computing, communication and smart world, pp 425–436. Springer

Shaha M, Pawar M (2018) Transfer learning for image classification. In: 2018 Second International conference on electronics, communication and aerospace technology (ICECA). https://doi.org/10.1109/iceca.2018.8474802

Shi J, Tomasi S (1994) Good features to track. In: Proceedings of IEEE conference on computer vision and pattern recognition, pp 593–600

Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. http://arxiv.org/abs/1409.1556

Singh C, Singh J (2019) Geometrically Invariant color, shape and texture features for object recognition using multiple kernel learning classification approach. Inf Sci. https://doi.org/10.1016/j.ins.2019.01.058

Srivastava S, Mukherjee P, Lall B, Jaiswal K (2017) Object classification using ensemble of local and deep features. In: 2017 ninth international conference on advances in pattern recognition (ICAPR), pp 1–6. IEEE

Talaat A, Yousri D, Ewees A, Al-qaness MAA, Damasevicius R, Elaziz MEA (2020) COVID-19 image classification using deep features and fractional-order marine predators' algorithm. Sci Rep 10(1):15364. https://doi.org/10.1038/s41598-020-71294-2

Varde A, Rundensteiner E, Javidi G, Sheybani E, Liang J (2007) Learning the relative importance of features in image data. In: 2007 IEEE 23rd international conference on data engineering workshop, pp 237–244. https://doi.org/10.1109/ICDEW.2007.4400998

Yadav SS, Jadhav SM (2019) Deep convolutional neural network based medical image classification for disease diagnosis. J Big Data 6:113. https://doi.org/10.1186/s40537-019-0276-2