Advanced Feature Combination Methods: A Performance Analysis in Object Classification

Amitav Mahapatra

Research Scholar, Biju Patnaik University of Technology, Rourkela, Odisha, India

Prashanta Kumar Patra

Dean, SRIC, SOA University, Bhubaneswar, Odisha, India

Abstract:-This study aims to evaluate advanced feature combination techniques regarding their impact on classification accuracy. The findings demonstrate that employing different advanced feature combination methods proves highly effective in practical applications. As additional features are systematically incorporated, classification accuracy improves. Our assessment of various advanced feature combination methods utilizes three diverse datasets: Xerox 7, UIUCTex, and Caltech-101, each comprising a variety of object types and classifiers.Comparing these methods, we find that the dominant set-based approach exhibits exceptional performance on the UIUCTex and Caltech-101 datasets, achieving classification accuracies of 86.44% and 86.55%, respectively. Conversely, for the Xerox 7 dataset, the clustering method excels, achieving a classification accuracy of 90.3%.

Keywords : Feature Combination, Classification Accuracy,MKL, CV Weight

1. Introduction

The accuracy and robustness of object classification systems are significantly improved by feature combination. Object classification entails the identification and categorization of objects within digital images or videos based on their visual features in the field of computer vision and machine learning. These characteristics may encompass basic descriptors, such as colour and texture, as well as more intricate ones, including shape, borders, and patterns.

The initial step in the object classification process is the extraction of pertinent features from the raw data, which may consist of images or recordings. These attributes function as distinguishing characteristics that differentiate one object from another. Nevertheless, no single feature descriptor may be sufficiently discriminative or robust in all situations. Therefore, the classification performance can be substantially enhanced by integrating multiple features, which captures various aspects of the objects and mitigates the effects of noise and variability.

Advanced methods of feature combination in object classification encompass a variety of sophisticated techniques designed to enhance classification accuracy and robustness by effectively integrating multiple types of features. These approaches leverage diverse feature representations and employ advanced algorithms to extract and combine pertinent information from data. By doing so, these methods aim to create comprehensive and discriminative feature sets that improve the performance of object classification systems across different applications and scenarios.In this paper, we evaluate the classification accuracy of a variety of advanced feature combination methods using three distinct datasets that that contain a variety of object types and classifiers.

The structure of the paper is as follows. Section 2 provides a concise overview of significant research advancements in advanced feature combination and demonstrates how these advancements serve as inspiration for our work in this study. Section 3 presents a range of advanced approaches for combining features, along with their respective algorithms. These methods are employed to assess the accuracy.Various classifier used are explained in section 4. The experimental results and analysis are presented in Section 5. Section 6 serves as the concluding section and future workof the paper.

2. Literature Survey

A new boosting-based feature combination method [1] integrates features. Variation boosting trains weak classifiers on numerous sets of features and combines them through weighted voting to produce a classifier output for each round, unlike standard boosting approaches. Studies in [1] show that this strategy incorporates feature selection, communication, and classifier learning, improving classification performance across datasets.

Schapire created boosting in 1990 to augment weak learning algorithms. AdaBoost grew from it. AdaBoost utilises weighted voting across two classes to combine poor classifiers and improve boosting methods. AdaBoost applies a uniform feature vector to weak classifier components, even when features fluctuate, yet each training instance has a fixed-length feature vector with ordered characteristics. In some cases, AdaBoost outperforms other approaches [1]. Each cycle, this boosting technique uses system-encoded characteristics to build a final classifier from several weak classifiers applied to data from each feature vector. Each iteration determines feature combinations using weighted voting. AdaBoost uses unreliable learning methods like decision trees and neural networks.Generic boosting works across classes. (unlike AdaBoost) surpasses conventional classification algorithms on three datasets. Boosting, feature extraction optimisation, and neural network parameter adjustment boost performance.

Other study [2] shows that boosting is reliable and persistent in complex multi-class scenarios. Basics, MKL, and boosting are covered in [2], along with advanced kernel feature combination. They introduce two MKL-inspired decision function approaches, LP-β and LP-B, along with LPBoost formulations. Next, apply these strategies to specific datasets.Mixed coefficients affect class attributes, but not in multi-class situations, says MKL. Evaluations treat all features equally. The LP-β technique chooses three of seven features, while other methods seek to include all. Oxford flower dataset trials show MKL and LP-β algorithms effectively remove irrelevant information. Over time, CG-Boosting decreases.

[2] employs boosting to combine features. In early Oxford flower dataset experiments, MKL with boosting can find significant kernels from many uninformative ones. This strategy's efficacy depends on attribute class identification. SIFT features employing MKL and CG-Boost's pyramid kernels outperform baseline approaches on Caltech101 and Caltech256. These beat PHOG pyramid kernels, CG-Boost, and MKL. The results of the LP-β (Boosting) technique were better than LPBoost and comparable to Caltech 101 MKL and baselines for both combinations. Additionally, LP-β has a runtime equivalent to MKL, indicating similar efficiency.The kernel combination in [3] incorporates prior knowledge, and classification performance affects attribute weight. Features are combined using kernel-based classifiers and bag-of-words histograms. Factorable features are weighted by predictive strength to effect forecasts. Features, kernels, and expertise improve intermediate data normalisation. Two approaches to use historical data:

Knowledge-weighted linear kernels

knowledge-weighted product kernel

Knowledge-weighted linear kernels may become product kernels when interacting with average kernels. The SVM classifier gets this kernel immediately. Extracting small picture portions requires the Harris-Laplace detector and careful sampling. The contrary, C-SIFT, rgSIFT, and modified colour SIFT explain these locations. The most successful feature combination method was KWPK. With little processing power, it can match cutting-edge approaches.

MKL analyses prefer KWPK and KWLK over product and average kernels [3]. MKL performs better in some datasets than KWPK and KWLK in chair, bus, and comparable datasets. Product and KWPK outperform KWLK and average. KWPK and KWLK perform similarly and competitively on the validation dataset, whereas MKL outperforms KWPK. Product and average kernels fail. KWPK and KWLK outperform MKL in speed and performance. MKL heavily invests on coefficient knowledge, while KWPK and KWLK train SVM classifiers and evaluate features. Pre-calculated kernels for each feature keep the coefficient constant during training, saving time without affecting performance.

Multiple descriptors and detectors enhance classification and feature combination [4]. Four datasets with different object categories are used to test how feature-related factors affect merging. PHOG, LBP, GIST, Gabor, RFS filter. Combining and assessing feature kernels:

Each classification feature's discrimination was assessed using the mean recognition rates from ten training-testing splitsEach scenario's ascending, falling, and mixed modes depend on discriminative strength.

Sequentially adding features tests feature combinations.Multiple powerful features exceed the best solo feature in all four datasets. Strong and weak qualities should cooperate, while weaker traits harm mixed mode performance. Sparse MKL or LP-β solutions [2] emphasise specific points in the final kernel combination. Feature combinations affect classification performance more than kernel combinations. Kernels behave like tops yet share many properties. This work developed a better kernel modification-based optimisation approach.Spatial pyramid research reveals levels boost performance but not performance. The kNN architecture offers average-product kernel compatibility, as widely investigated in [5]. Selection-Based Average Combination (SBAC) outperforms normal average combination in experiments [4]. builds on kNN wins. This combo descends with a climb, peak, and fall pattern. This pattern encourages kNN k selection caution. Cross-validation ensures feature class identification, affecting layout. SBAC excels on Caltech-101, Flower-17, Scene-15, and Event-8. The MKL combination is above average [2].MKL efficiency is adjustable [6, 7, 8, 9]. Innovative non-linear kernel fusing approaches like [10] boost performance. SMO accelerates training in large datasets and kernel spaces, especially for -norm Multiple Kernel Learning (MKL), where kernels boost performance [14]. Soft Salient Coding (SSaC) addresses SPM SaC information suppression [15]. Multiple encoding and pooling algorithms improve image categorization.According to [15], MKL's adaptive encoding and pooling increase image classification. In particular, SSaC prevents the original SaC methodology from suppressing information.More SSaC image data helps image categorization.Encoding, pooling, and code quality testing.Multiple regularisation approaches and additional training data improve MKL classification.HOG descriptors help SSaC experimentally beat GSaC and regular SaC. Multiple descriptions and category training examples are handled by SSaC. MKL with -norm performance needs lots of training data, confirming regularisation. With enough training data, MKL outperforms other algorithms in feature relevance across categories.A new feature processing method outperforms classification and reduces feature redundancy [16]. Study features are processed using specific procedures. These methods integrate features, locate redundant features, convert numerical feature values to categorical ones, and find hidden structures in original and enhanced data. The recommended method surpasses SVM in classification accuracy while keeping an equal ROC benchmark, according to UCI repository data.

Literature surveys demonstrate this research gap:

Despite advances in object categorization using feature extraction, feature fusion research and comparison are lacking. Many research focus on individual feature extraction rather than merging features for classification improvement.
There is little study on domain-specific feature combinations. The image databases include medical, satellite, and natural scene images. Strategies that integrate domain-specific traits to address domain-specific challenges are understudied.

This work investigates and evaluates advanced feature combination strategies for object classification in image datasets to fill the research gap

3. Advanced Feature Combination Methods and Algorithms

Feature combination in object classification refers to the process of integrating multiple types or sources of features derived from input data, such as images, with the goal of enhancing the accuracy and robustness of object classification algorithms.

3.1 Advanced Methods

Advanced methods of feature combination in object classification encompass a range of sophisticated techniques aimed at improving classification accuracy and robustness by integrating multiple types of features effectively. These methods leverage diverse feature representations and advanced algorithms. These methods leverage diverse feature representations and advanced algorithms to extract and combine relevant information from data. Here's an overview of some advanced approaches:

A) CV Weight Method

The CV Weight Method of Feature Combination is a methodical way to combining features, which is based on their relevance values obtained from cross-validation (CV). Below is a comprehensive algorithmic framework for implementing this method:

Algorithm for CV Weight Method of Feature Combination

1. **Input**:
   - **Dataset**: Contains objects to be classified and extracted features.
   - **Feature Extraction**: Obtain initial feature vectors for each object.
2. **Cross-Validation Setup**:
   - Choose a suitable cross-validation strategy (e.g., k-fold cross-validation) depending on the size and nature of your dataset.
   - Split the dataset into training and validation sets according to the chosen CV strategy.
   -
3. **Feature Importance Calculation**:
   - **Initialization**:
     - Select a machine learning model suitable for your classification task (e.g., Random Forest, Gradient Boosting Machine).
     - Define a metric for evaluating model performance (e.g., accuracy, F1-score).

- **Cross-validation loop**:
  - For each fold in the cross-validation:
    - Train the model on the training data subset.
    - Evaluate the model on the validation data subset using the chosen metric.

- Compute feature importance scores specific to the trained model. Methods for feature importance calculation include:
    - **Mean Decrease Accuracy**: Measure how much model performance (e.g., accuracy) drops when a feature is randomly shuffled.
    - **Gini Importance**: Measure based on the decrease in impurity in decision trees.
    - **Permutation Importance**: Assess the impact of feature shuffling on model performance.
- Aggregate feature importance scores across all folds to get a robust estimation.

4. **Feature Combination Using Weights**:
    - Normalize the feature importance scores (optional) to ensure they sum up to 1 or have meaningful relative weights.
    - Combine the original feature vectors into a single combined feature vector for each object using the computed importance weights:
        - **Weighted Average**: Combine numerical features by taking a weighted average, where weights are determined by their importance scores.
        - **Feature Selection**: Select a subset of features based on their importance scores if using a sparse model or if reducing dimensionality is beneficial.
5. **Clustering or Classification**:
    - Apply clustering or classification algorithms to the combined feature vectors:
        - **Clustering**: Group similar objects together based on the combined feature representations.
        - **Classification**: Train a classifier using the combined feature vectors to predict object labels.
6. **Output**:
    - Obtain clusters or predicted labels for each object based on the chosen algorithm.

By amalgamating features based on their significance weights, you concentrate on the most enlightening portions of the data, potentially enhancing model accuracy and generalisation. By employing feature selection based on relevance ratings, it is possible to decrease the dimensionality of the data, resulting in quicker training times and more straightforward models. Gaining an understanding of the features that have the biggest impact on predictions might offer valuable insights into the fundamental qualities of the objects being classified.

B) MKL Method

Using the MKL (Multiple Kernel Learning) approach of feature combination, several kernels (representing several feature spaces or transformations) are combined into a single kernel matrix. Subsequently, this unified kernel matrix finds application in a variety of machine learning problems, such clustering and classification. The MKL method of feature combining is implemented as follows algorithmically:

Algorithm for MKL Method of Feature Combination

1. **Input**:
    - **Dataset**: Contains objects to be classified and extracted features.
    - **Multiple Kernels**: Kernels representing different feature spaces or transformations. These could be based on different feature extraction techniques or representations.
2. **Initialization**:
    - Choose a set of kernels($K_1, K_2, \ldots, K_m$)each corresponding to a different feature space or transformation.

- Initialize weights $\alpha=(\alpha_1,\alpha_2,\ldots,\alpha_m)$ for combining the kernels. These weights determine the contribution of each kernel to the combined kernel matrix.

3. **Objective Function**:
   - Define an objective function that balances between the fit to the data and the complexity of the model. Typically, this involves optimizing over the weights α

$$\min_{\alpha} \quad \frac{1}{n}\sum_{i=1}^{n} L(y_i, f(x_i)) + \lambda R(\alpha)$$

where L is a loss function (e.g., hinge loss for SVM, cross-entropy for logistic regression), $f(x_i)$ is the predicted output for object x_i. R(α) is a regularization term on the weights α and λ controls the regularization strength.

4. **Optimization**:
   - Solve the optimization problem to find the optimal weights α\alphaα that minimize the objective function. This step involves:
     - Using techniques such as gradient descent, coordinate descent, or convex optimization methods depending on the chosen loss function and regularization.
5. **Combined Kernel Matrix Construction**:
   - Construct the combined kernel matrix Kcombined using the optimized weight α .

$$K_{\text{combined}} = \sum_{j=1}^{m} \alpha_j K_j$$

Where k_j is the matrix corresponding to the jth kernel.

6. **Machine Learning Task**:
   - Use the combined kernel matrix $K_{combined}$ in a machine learning task such as:
     - **Classification**: Train a classifier (e.g., SVM, kernel-based methods) using $K_{combined}$ as the kernel matrix.
     - **Clustering**: Apply clustering algorithms (e.g., kernel k-means) using $K_{combined}$.

Choose kernels that reflect distinct aspects or representations of the data, such as Gaussian kernels for various feature representations or domain-specific kernels.The regularisation parameter λ should be adjusted to achieve a balance between preventing overfitting and fitting the data.Employ effective optimisation strategies that are appropriate for the objective function and constraints that have been selected.

The potential to enhance classification or clustering performance is achieved by combining multiple kernels, which enables the capture of complementary information from distinct feature spaces.By selecting suitable kernels and adjusting weights, the method can be customised to accommodate various datasets and duties. MKL is theoretically grounded in learning theory, which offers a principled approach to the integration of feature representations.

C) Dominant Set based Method

The Dominant Set based feature combination approach is a methodology employed in clustering and feature selection. Its purpose is to find and leverage dominant features that make a substantial contribution to the clustering process. Below is a systematic framework for implementing this method:

Algorithm for Dominant Set Based Feature Combination

1. **Input**:
   - **Dataset**: Contains objects to be clustered and extracted features.
   - **Similarity Measure**: Define a similarity measure S(i,j) between objects i and j, often based on feature distances or similarities.
2. **Feature Selection**:
   - **Feature Extraction**: Extract relevant features from each object in the dataset.
3. **Similarity Matrix Construction**:
   - Construct a similarity matrix **S** where S(i,j) represents the similarity between objects i and j. This matrix is typically symmetric and can be based on feature distances or similarities.

Dominant Set Identification:

- **Initialization**: Start with an empty set DS=∅
- **Iteration**:
  - For each object i:
    - Compute the dominance score Dom(i) which measures how well object i can dominate other objects based on their similarities. This can be computed as:

$$\mathrm{Dom}(i) = \sum_{j \neq i} S(i,j)$$

Add i to DS if Dom(i) is above a certain threshold or relative to other objects in the dataset.

Repeat the iteration until no more objects can be added to DS.

Feature Combination:

- Extract features corresponding to objects in the dominant set DS.
- Combine these features into a single feature vector or representation for each object in the dataset. This can be done using:
  - Concatenation of feature vectors from objects in DS.
  - Aggregation (e.g., averaging, weighted averaging) of features from DS objects.

 Clustering or Classification:

- Apply clustering algorithms (e.g., K-means, DBSCAN) or classification models (e.g., SVM, Random Forest) using the combined feature vectors obtained from step 5.

Establish a specific value for the dominance score Dom(i) that will be used to determine the inclusion of items in the dominant set DS.Sequentially examine each object in the dataset to progressively construct the dominant set according to their dominance scores.Customise the method of combining features (such as concatenation or aggregation) according to the characteristics of the features and the clustering/classification algorithm employed. Evaluate the quality of the clusters or classification performance by utilising suitable metrics such as the silhouette score for clustering and accuracy for classification.

D) Clustering Method

The feature combination clustering method entails the clustering of data objects based on combined or aggregated feature representations. This method is beneficial when dealing with datasets that necessitate the integration of various types of features or representations in order to achieve effective clustering. Here is an algorithmic framework for the clustering technique of feature combination:

Algorithm for Clustering Method of Feature Combination

1. **Input**:
    - **Dataset**: Contains objects to be clustered and extracted features.
    - **Feature Extraction**: Extract relevant features from each object using appropriate techniques (e.g., HOG, SIFT for images; TF-IDF, word embeddings for text).
2. **Feature Combination**:
    - Combine the extracted features into a single feature vector or representation for each object. This can involve:
        - **Concatenation**: Combine feature vectors if they are of compatible dimensions.
        - **Dimensionality Reduction**: Apply techniques like PCA, t-SNE, or autoencoders to merge different feature representations into a unified space.
        - **Aggregation**: Calculate statistical summaries (e.g., mean, median, variance) of feature values across different feature sets.
3. **Clustering Algorithm Selection**:
    - Choose an appropriate clustering algorithm suited to the combined feature space:
        - **K-means**: Suitable for clusters with spherical shapes and similar sizes.
        - **Hierarchical Clustering**: Builds a tree of clusters that can be cut at different levels.
        - **DBSCAN**: Effective for clusters of varying shapes and densities.
        - **Mean-shift**: Automatically finds the number of clusters and their centers.
4. **Clustering Process**:
    - Initialize the chosen clustering algorithm with the combined feature representations obtained from step 2.
    - Apply the clustering algorithm to group objects into clusters based on similarities in their combined feature representations.
    - Iteratively refine clusters based on the convergence criteria of the clustering algorithm.
5. **Cluster Evaluation**:
    - Evaluate the quality of clusters obtained using:
        - **Internal Evaluation Metrics**: such as silhouette score, Davies-Bouldin index, or cohesion and separation metrics.

        - **External Evaluation Metrics**: if ground truth labels are available, such as Adjusted Rand Index or Fowlkes-Mallows Index.
6. **Post-processing** (Optional):
    - Refine clusters or perform post-processing steps based on domain knowledge or additional criteria:
        - Merge or split clusters based on further analysis of cluster characteristics.
        - Remove outliers or noise points if using density-based clustering algorithms like DBSCAN.
7. **Utilization**:
    - Use the identified clusters for various purposes, such as:
        - **Object Classification**: Assign labels to clusters based on majority voting or centroid characteristics.
        - **Anomaly Detection**: Identify clusters with few members or unusual feature combinations.

The integration of diverse features enhances clustering by amalgamating multiple types of features or representations.Utilises clusters as more precise feature representations for subsequent classification tasks, potentially enhancing accuracy.Capable of being adjusted to various forms of data and methods for extracting features, hence increasing its usefulness across different fields.

4. Classifier algorithms used

Our work includes several classifier methods to demonstrate their accuracy effects.
They are:

A. KNN

Basic instance-based KNN.A new data point's class is predicted using its K nearest neighbours' majority class. Number of neighbours (K) and distance metric (e.g., Euclidean distance) matter.KNN prediction is simple yet computationally expensive with large datasets.

B. SVM

SVM is a supervised learning method for classification and regression, particularly classification. For class splitting, SVM calculates the right hyperplane.The hyperplane's maximum distance from the nearest data points. Uses polynomial, RBF, or sigmoid kernel functions to efficiently handle non-linear decision boundaries.SVMs do well in high-dimensional domains with more features than samples.

C. Random Forest

The Random Forest technique uses an ensemble learning approach. Many decision trees are trained by Random Forest.Each forest tree forecasts independently using bootstrap data. Final predictions come from aggregating all tree forecasts (most votes for classification, average for regression).Random Forests score feature significance, resist overfitting, and handle high-dimensional data. Their efficiency and scalability in machine learning make them popular.

D.Adaptive Boosting

AdaBoost builds strong classifiers from poor ones via ensemble learning.It constantly trains weak classifiers on updated data. Adjusts erroneously categorised instance weights to focus classifiers on difficult cases.Poor learners improve accuracy without overfitting with AdaBoost. Classification jobs with several classifiers use it widely.

E. Gradient Boosting

Gradient Boosting Classifier (GBM) is a sophisticated ensemble learning technique for classification:Gradient Boosting builds decision trees that correct each other.Trees that minimise loss improve a loss function (typically gradient descent) and let shallow decision trees learn and forecast properly.Gradient Boosting performs well with complex variables. Competitions and machine learning use it for its adaptability and non-linear simulation.

F.Bagging

It increases machine learning algorithm stability and accuracy. Bagging generates dataset bootstrap samples. Bootstrapping samples train an independent base classifier, usually decision trees.Final predictions come from averaging regression base classifier predictions or voting classification jobs.Bagging reduces variance and overfitting by integrating predictions from multiple data subset models. It improves projected performance for unstable models sensitive to modest training data changes.

G. Logistic Regression

Logistic regression (LR) is a key binary classification supervised learning method.Logistic regression describes binary outcome probability with 0–1 values.Fitting a linear decision boundary to feature space divides classes.MLE and gradient descent estimate model parameters.Logistic Regression is simple, fast, and effective when characteristics and aim variables are linear. Many fields use it for binary classification due to its simplicity and efficacy.

H. Naïve Bayes Classifier

The Naïve Bayes Classifier is a rapid and effective machine learning algorithm for prediction. Probabilistic classifiers assess object likelihood.

I. Classifier Decision Tree

The Decision Tree Classifier is a supervised classification and regression technique. Recursively partitioning data by feature values builds a decision tree.The approach chooses the best feature to split data at each node to maximise classification information gain or decrease regression variance. After training, a leaf node determines the class label (for classification) or predicted value (for regression) from new data points from the root.Non-linear interactions and complex decision limitations are evaluated via decision trees. Prune or ensemble methods like Random Forests or Gradient Boosting can reduce overfitting.

5. EXPERIMENTAL RESULTS

The results and discussions of this paper are the result of extensive research that was conducted using Python programming. The investigations were performed on a computer utilising the Ubuntu 16.04 operating system, 4GB of memory, 500GB of storage, and a Core i5 processor.

Here are three data sets that we have included.

Data set 1- Xerox7
Data set 2- UIUCTex
Data set 3 – Caltech-101

The experimental configuration involved using 70% of the stated image sets for training and 30% for testing.

We evaluated the classification accuracy by integrating several features to showcase the efficacy of feature combination. We have evaluated the same utilising diverse classifiers. The acquired outcomes are converted into graphs and further scrutinised. To ensure optimal outcomes, measures were taken to maintain system stability.

5.1

Classification Accuracy

1. Xerox7 Dataset

The dataset [37] comprises 1776 photos categorised into seven distinct classes: Faces, buildings, trees, automobiles, phones, bikes, and books. The object poses exhibit a great degree of variability, and there is a substantial quantity of background clutter, including elements from various categories, which adds to the complexity of the classification challenge.

Table1: Xerox7 Classification Accuracy

Feature Combination Method Used	Classifier Used								
	Decision tree	KNN	SVM	Random Forest	AdaBoost	Gradient Boosting	Bagging	Logistic Regression	Naïve Bayes
CV Weight	82.2	82.6	**83.29**	80.27	79.2	77.6	81	79.2	73.88
MKL	75.1	75.9	**79.21**	76.21	76.1	73.9	77.53	73.2	76.23
Dominant Set	84.7	87.5	88.44	87.44	84.7	85.5	**89.44**	83.11	87.25
Clustering	87.6	89.1	88.3	86.3	88.6	86.1	**90.3**	84.22	81.21

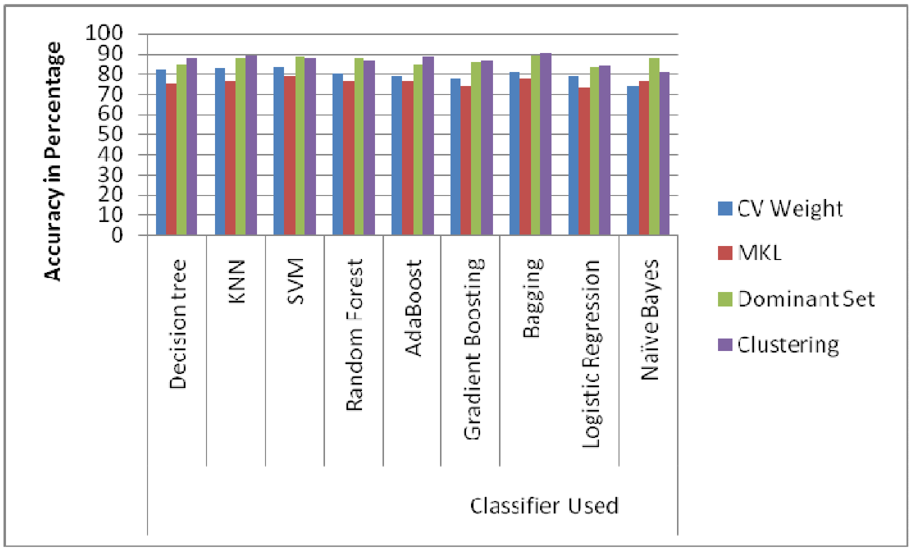

Fig-1 Performance comparison of various Feature Combination Methods for Xerox7 dataset

According to the data in Table 1, the highest accuracy of 90.3% was achieved using the clustering method with the Bagging Classifier on the Xerox7 dataset.

2. UIUCTex Dataset

The dataset[38] of 25 distinct texture classes, each consisting of 40 images. The surfaces exhibit varying textures mostly caused by differences in albedo (e.g. wood and marble), three-dimensional shapes (e.g. gravel and fur), or a combination of both (e.g. carpet and brick). Additionally, it exhibits notable alterations in perspective, unregulated lighting, random rotations, and variations in scale within each category.

Table 2 : UIUCTex Dataset Classification Accuracy

Feature Combination Method Used	Classifier Used								
	Decision tree	KNN	SVM	Random Forest	AdaBoost	Gradient Boosting	Bagging	Logistic Regression	Naïve Bayes
CV Weight	79.6	82.6	81.25	80.25	79.2	79.6	**83**	74.2	71.88
MKL	73.2	74.5	**82.21**	73.21	78.1	72.9	74.52	73.2	76.23
Dominant Set	86.5	87.5	**86.44**	84.45	83.7	84.5	86.21	82.11	81.25
Clustering	86	87.1	**86.3**	85.3	85.6	84.1	84.3	83.22	84.21

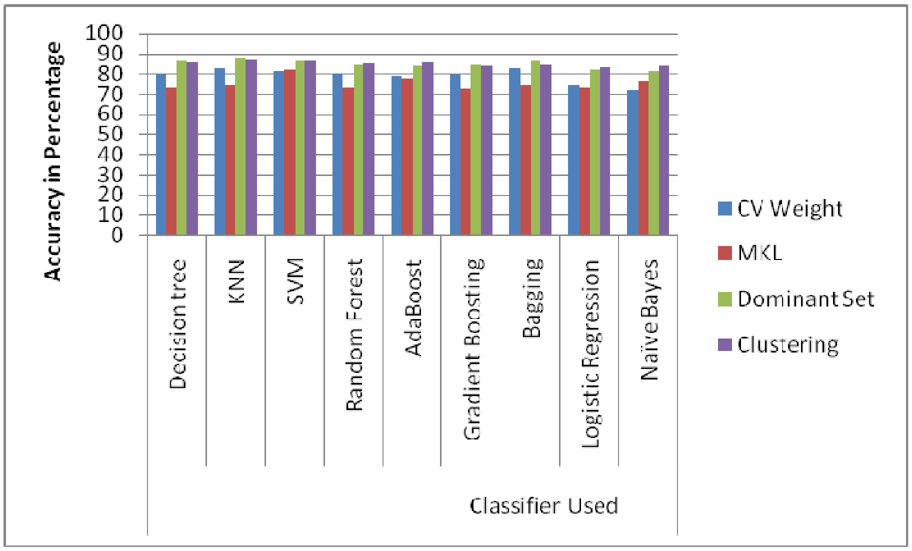

Fig-2 Performance comparison of various Feature Combination Methods for UIUCTex Dataset

According to the data in Table 2, the highest accuracy of 86.44% was achieved using the dominant set based feature combination method with the SVM Classifier on the UIUCTex dataset.

3. Caltech-101data set

The dataset [39] comprises a total of 9,146 photos, distributed across 101 distinct item categories, along with an extra background/clutter category. The category ranges from 31 to 800. Categories that are commonly and widely recognised, such as faces, typically have a greater quantity of photos compared to other categories.

Table 3: Caltech 101 data set Classification Accuracy

Feature Combination Method Used	Classifier Used								
	Decision tree	KNN	SVM	Random Forest	AdaBoost	Gradient Boosting	Bagging	Logistic Regression	Naïve Bayes
CV Weight	73.6	80.6	82.25	81.25	73.2	73.6	**83.33**	71.2	77.88
MKL	74.2	81.9	**83.21**	72.21	71.1	73.9	74.51	72.2	74.23
Dominant Set	81.5	85.5	**85.44**	83.11	81.7	84.5	**86.58**	85.12	84.23
Clustering	83.21	75.1	**85.3**	85.37	77.62	84.11	82.2	78.22	72.21

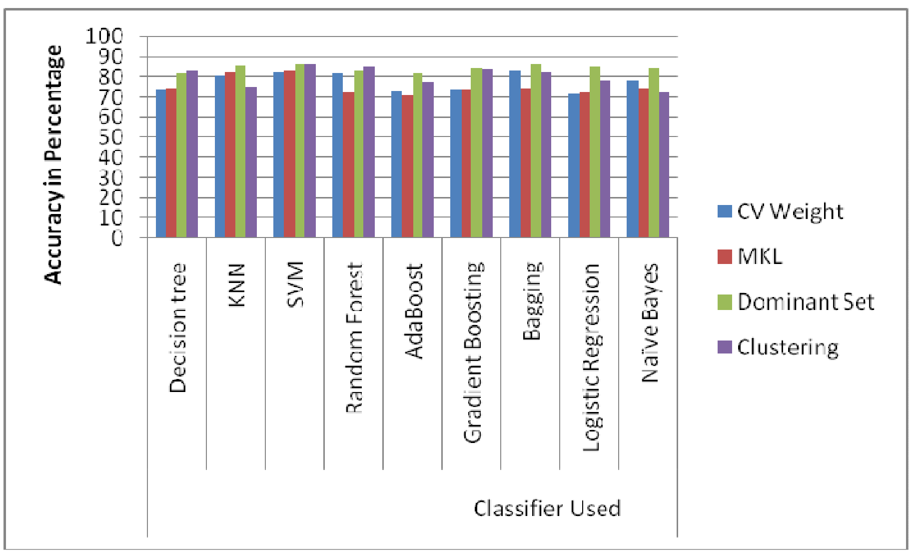

Fig 3 Performance comparison of various Feature Combination Methods for Caltech 101 Dataset

According to the data in Table 3, the highest accuracy of 86.58% was achieved using the dominant set based feature combination method with the Bagging Classifier on the Caltech 101 Dataset

Table 4 : Running Time Copmparision of by different classifiers for different datasets

Method used	Xerox7	UIUCTex	Caltech-101
CV Weight	171.6	112.2	172.25
MKL	272.2	323.9	274.21
Dominant Set	97.9	94.5	112.43
Clustering	83.4	88.8	94.12

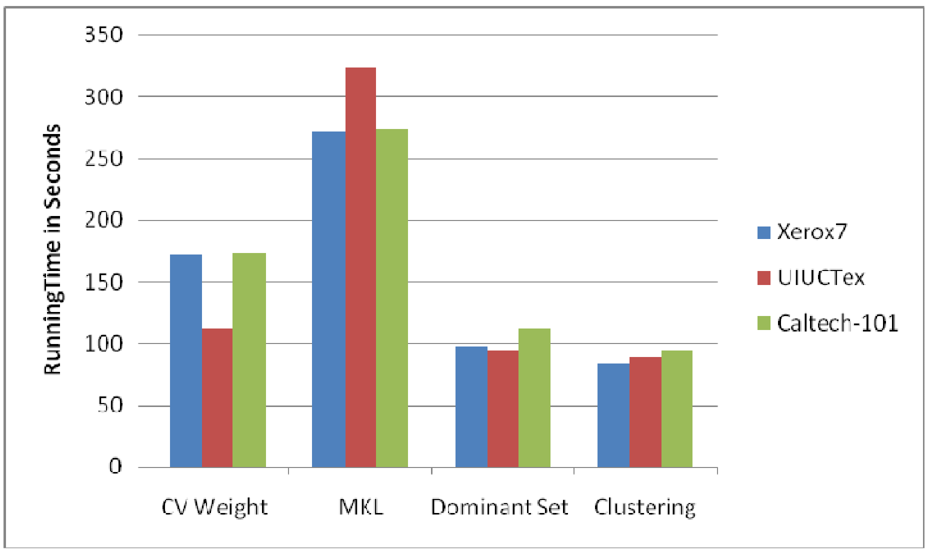

Fig 4 Running Time Copmparision for different feature combination method on different dataset

According to the data in Table 4, the MKL technique has the longest running time among all the data sets. This could be attributed to the enlargement of the feature matrix, which results in increased processing time.Clustering method takes minimum running time in all the datasets.

6. Conclusion & Future Work

The findings indicate that combining features greatly improves the accuracy of classification in real-world scenarios. Consistently incorporating new features in a step-by-step manner enhances performance. Therefore, these systems provide useful functionality and can be effectively used in various everyday situations. By optimising several parameters that control feature integration, we may improve the results of applications even more. The efficacy of classification accuracy is intricately linked to the deliberate management of various factors, emphasising the significance of strategic handling.

Future feature combination for object classification research should pursue various possible avenues. Studying attention mechanisms could improve feature integration by emphasising informative features at different categorization phases. Hierarchical feature fusion approaches like pyramidal pooling or nested feature extraction should be researched to capture local details and global context. Flexible feature combination strategies that adapt fusion techniques to input features or learning stages can improve flexibility and performance. Graph Neural Networks (GNNs) could improve classification accuracy by using relational information between features. Bayesian uncertainty modelling can improve robustness in uncertain or noisy situations. Feature integration techniques that align features with high-level semantic ideas may increase interpretability and performance. Meta-learning and cross-modal feature fusion for optimised feature combining across datasets are also promising. Finally, viable deployment-friendly real-time feature integration mechanisms must be developed. All of these methods strive to improve feature combination in object categorization.

Refrences

[1] Xu Cheng Yin, Chang –Ping Liu and Zhi Han, “Feature combination using boosting” X.-C. Yin et al. / Pattern Recognition Letters 26 (2005) 2195–2205

[2] P. Gehler and S. Nowozin, “On feature combination for multiclass object classification,” in *Proc. IEEE 12th Conf. Comput. Vis.*, Kyoto, Japan, Sep./Oct. 2009, pp. 221–228.

[3] Linbo Zhang, Baihua Xiao, Chuncheng Wang, Gang Cheng and Yunxue Shao, “An efficient strategy for features combination” 2021 3rd International Congress on Image and Signal Processing (CISP2021)

[4] J. Hou, B.-P. Zhang, N.-M. Qi, and Y. Yang, “Evaluating feature combination in object classification,” in *Proc. Int. Symp. Vis. Comput.*, Las Vega, NV, USA, Sep. 2011, pp. 597–606.

[5] Jian Hou, Wei-Xue Liu and Hamid Reza Karimi, “Exploring the best classification from average feature combination” Volume 2014, Article ID 602763, 7 pages

[6] G. R. G. Lanckriet, N. Cristianini, P. Bartlett, L. El Ghaoui, and M. I. Jordan, “Learning the kernel matrix with semidefinite programming,” *J. Mach. Learn. Res.*, vol. 5, pp. 27–72, Jan. 2004.

[7] A. Kumar and C. Sminchisescu, “Support kernel machines for objectrecognition,” in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Rio de Janeiro,Brazil, Oct. 2007, pp. 1–8.

[8] Y.-Y. Lin, T.-L. Liu, and C.-S. Fuh, “Local ensemble kernel learning forobject category recognition,” in *Proc. IEEE Conf. Comput. Vis. PatternRecognit.*, Minneapolis, MN, USA, Jun. 2007, pp. 1–8

[9] M. Varma and D. Ray, “Learning the discriminative power-invariance trade-off,” in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Rio de Janeiro,Brazil, Oct. 2007, pp. 1–8.

[10] J. Yang, Y. Li, Y. Tian, L. Duan, and W. Gao, “Group-sensitive multiple kernel learning for object categorization,” in *Proc. IEEE 12th Int. Conf.Comput. Vis.*, Kyoto, Japan, Sep./Oct. 2009, pp. 436–443.

[11] C.Cortes, M.Mohri, A.Rostamizadeh, “Learning non-linear combinations of kernels”, in: Advances in Neural Information Processing Systems, 2021, pp.396–404.

[12] M.Varma, and B.R.Babu, “More generality in efficient multiple kernel learning”, in: International Conference on Machine Learning , 2009, pp.1065–1072.

[13] S.V.N.Vishwanathan, Z.Sun, N.T.Ampornpunt, M.Varma, “Multiple kernel learning and the SMO algorithm”,in: Advances in Neural Information Processing Systems, 2010, pp.2361–2369.

[14] F.R.Bach, “Exploring large feature spaces with hierarchical multiple kernel learning”, in: Advances in Neural Information Processing Systems, 2008, pp. 105–112.

[15] Wei Luo, Jian Yang, Wei Xu, Jun Li and Jian Zhang, “Higher –level feature combination via multiple kernel learning for image classification”

[16] Jian-hua Yeh, Chen Lin, Yuan-ling Chang, ‘Classification improvement based on feature combination and topic vector model.” 2012 International Conference on Systems and Informatics (ICSAI 2012)

[17] Li Zhou, Zongtan Zhou and Dewen Hu, “Scene classification using multi resolution low level feature combination” L. Zhouetal./Neurocomputing122(2013)284–297

[18] S. Lazebnik, C. Schmid, and J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in *Proc.IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, New York, NY, USA, Jun. 2006, pp. 2169–2178.

[19] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holisticrepresentation of the spatial envelope,” *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.

[20] J. Wu and J. M. Rehg, “Beyond the Euclidean distance: Creating effective visual codebooks using the histogram intersection kernel,” in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Kyoto, Japan, Sep./Oct. 2009, pp. 630–637

[21] J. Hou and M. Pelillo, “A simple feature combination method based on dominant sets,” *Pattern Recognit.*, vol. 46, no. 11, pp. 3129–3139,2013.

[22] M. Pavan and M. Pelillo, “Dominant sets and pairwise clustering,” *IEEETrans. Pattern Anal. Mach. Intell.*, vol. 29, no. 1, pp. 167–172, Jan. 2007.

[23] Jian Hou, Huijun Gao, Qi Xia and Naiming Qi, “Feature combination and the kNN Framework in object classification” ieee transactions on neural networks and learning systems, Vol.27, No.6,June 2016

[24] Jian Hou, Huijun Gao and Xuelong li, “Feature combination via clustering” ”ieee transactions on neural networks and learning systems, 2017.

[25] S. R. Bulò, M. Pelillo, and I. M. Bomze, “Graph-based quadratic optimization: A fast evolutionary approach,” *Comput. Vis. Image Understand.*, vol. 115, no. 7, pp. 984–995, 2011.

[26] Lin Yuan, Fanglin Chen, Li Zhou and Dewen Hu, “Improve scene classification by using feature and kernel combination.

[27] Mehrdad Ahmadi Soofivand, Abdollah Amirkhani, Mohammad Reza Daliri, GholamaliRezaeirad “Feature level combination for object classification” 2014 4th conference on International Conference on Computer and Knowledge Engineering.

[28] Taiwan Huang, Lei Li, Yazhao Zhang, Junqi Chi “Summarization based on multiple feature combination”

[29] Aibo Song, Wei Qian, Zhiang Wu, Jinghua Zhao “Discriminative Feature combination Selection For Enhancing Multiclass Classification.” 2015 International Conference on Behavioral, Economic and Socio-Cultural Computing.

[30] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,”*Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[31] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, USA, Jun. 2005, pp. 886–893.

[32] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “SURF: Speeded up robust features,” *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, 2008

[33] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scaleand rotation invariant texture classification with local binary patterns,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002

[34] L.-J. Li and L. Fei-Fei, “What, where and who? Classifying events by scene and object recognition,” in *Proc. IEEE 11th Int. Conf. Comput.Vis.*, Rio de Janeiro, Brazil, Oct. 2007, pp. 1–8.

[35] Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from fewtraining examples: an incremental bayesian approach tested on 101 object categories.In: CVPR, Workshop on Generative-Model Based Vision, p. 178 (2004)

[36] O. M. Parkhi, A. Vedaldi, A. Zisserman, C. V. Jawahar, “cats and dogs” IEEE conference on computer vision and pattern recognit

[37] Csurka, Gabriella, et al. "Visual categorization with bags of keypoints." *Workshop on statistical learning in computer vision, ECCV*. Vol. 1. No. 1-22. 2004.

[38] Lazebnik, Svetlana, Cordelia Schmid, and Jean Ponce. "A sparse texture representation using local affine regions." *IEEE transactions on pattern analysis and machine intelligence* 27.8 (2005): 1265-1278.

[39] Fei-Fei, Li, Rob Fergus, and Pietro Perona. "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories." *2004 conference on computer vision and pattern recognition workshop*. IEEE, 2004.