Chapter 6

Automatic Image Annotation using Texture Features and Sparse Reconstruction

This chapter describes the image annotation model. The main contributions are: choosing the best feature selection method for image annotation; proposing a subset of features using DoubleSort technique; and improvising the traditional multi-label classification method. The model is introduced in Section 6.1. The theoretical foundations of the proposed model and brief introduction of state-of-the-art methods are described in Section 6.2. The description of the proposed model is given in Section 6.3. Section 6.4 gives the result generated by the proposed model and its analysis. Section 6.5 concludes the chapter.

6.1 Background

In this chapter, a general framework of image annotation is proposed by involving salient object detection (SOD), feature extraction, feature selection, and multi-label classification. For SOD, Augmented-Gradient Vector Flow (A-GVF) is proposed, which fuses the benefits of GVF and Minimum Directional Contrast. The model also suggests controlling the background information to be included for annotation. This chapter brings about a comprehensive study of all primary feature selection methods for a survey of four publicly available datasets. The study concludes with the proposition of using Fisher's method for reducing the dimension of features. Moreover, this model also proposes a set of features that are found to be strong discriminants by most of the methods. This reduced set for image annotation gives 3-4% better accuracy across all the four datasets. This model also proposes an improved multi-label classification algorithm C'-MLFE.

6.2 Survey of salient object detection, feature selection and image annotation methods

This section gives a brief explanation of the related work in the field of salient object detection, feature selection, and image annotation. All the relevant work mentioned here have been used for comparison to the proposed model. The results of the comparison and their discussion are present in Section 6.4.

6.2.1 Salient Object Detection

The salient object detection model used in this chapter is the same as described in Section 3.3.3. To evaluate the procedure proposed in this chapter, a comprehensive study of 19 different salient object detection benchmark algorithms was done. A brief explanation is given below:

1. Context-aware saliency detection (CA) [201]: This algorithm, proposed by Stas *et al.*, is based on four psychological principles of saliency: local consideration (contrast, color), global considerations, visual organization rules and priors about the location of the object.

2. Visual saliency estimation by nonlinearly integrating features using region covariances (COV) [202]: In this paper, authors use covariance matrices of simple image features to form meta-features for estimating saliency.

3. Saliency detection via dense and sparse reconstruction (DSR) [181]: Superpixels are used to form image boundaries and then for each region dense and sparse reconstruction errors are calculated. Pixel-level saliency is calculated by the integration of multiscale reconstruction errors.

4. Fast and Efficient Saliency Detection Using Sparse Sampling and Kernel Density Estimation (FES) [203]: This method studies local contrast in Bayesian structure. It uses the center prior.

5. Graph-regularized saliency detection with convex-hull-based center prior (GR) [204]: The authors in this paper use center, smoothness, and contrast priors to determine the saliency. They utilize the convex hull of interest points to find object location rather than directly approaching the center of the image. The saliency energy function uses graph regularization for minimization. Smoothness is used to highlight the salient object and suppress the background object homogeneously.

6. Salient region detection and segmentation (ICVS) [205]: Saliency maps are obtained using contrast prior. The maps are calculated at different scales and are finally integrated pixel-wise.

7. Saliency detection via absorbing Markov chain (MC) [206]: In this paper, appearance divergence and spatial distribution determine virtual boundaries of the image, which are then used to set up absorbing nodes of the Markov chain. Using the absorbed time as a metric, salient objects are separated from the background.

8. Salient object detection via minimum directional contrast (MDC) [177]: Huang and Zhang use spatial distribution of contrast as a metric to calculate the raw saliency map. The map is refined by using marker-based watershed segmentation to mark foreground and background pixel and to suppress the saliency accordingly.

9. What makes a patch distinct? (PCA) [184]: The authors find unique patterns using color and various other high-level cues and priors.

10. Saliency optimization from robust background detection (RBD) [207]: This work focuses on background connectivity. It also proposes an idea of a principled optimization framework for integrating multiple low-level cues.

11. Global contrast based salient region detection (RC) [166]: In this work, the histogram-based global contrast map is produced for marking salient regions. The regions are assigned scores based on regional contrast and distance from other regions of the image.

12. Segmenting salient objects from images and videos (SEG) [208]: The authors of this paper combine saliency measure with the conditional random field. Saliency measure is obtained using a statistical framework and local contrast information regarding illumination and color. The implementation is done using integral histogram and graph cut solvers.

13. Saliency estimation using a non-parametric low-level vision model (SIM) [209]: The saliency model in this work is built by working on the color property of human vision and inherent spatial pooling mechanism. An inverse wavelet transform is used for scale integration.

14. Using self-resemblance for saliency detection (SeR) [210]: In this work, the likeness of a pixel to the next pixel is used to compute the local regression kernels. Matrix cosine similarity is used for calculating the saliency measure.

15. Structured matrix decomposition (SMD) [43]: Peng *et al.* use matrix decomposition where a matrix is decomposed into a low-rank matrix to represent the background and a sparse matrix to represent salient objects. Laplacian regularization is used to increase the gap between the background and the salient object.

16. Saliency detection: a spectral residual approach (SR) [211]: The analysis of the image in the spectral domain is done using the log-spectrum method, and the saliency map is built in the spatial domain.

17. Image signature: highlighting sparse salient regions (SS) [212]: In this work, image signatures are developed, which are then used for estimating the foreground region of the image. These signatures are further used in developing the saliency algorithm.

18. SUN: a Bayesian framework for saliency using natural statistics (SUN) [213]: Saliency is obtained by the pointwise mutual information between the features.

19. Visual saliency detection by spatially weighted dissimilarity (SWD) [214]: The authors first reduce image dimension using Principal Component Analysis. Three pillars of their work are - calculating how different image patches are, the distance between image patches, and the center bias.

6.2.2 Feature Selection

For selecting the best method to curb the 'curse of dimensionality', this work studies and compares the result of using eighteen different feature selection methods for image annotation. They are as follows:

1. Infinite Latent Feature Selection (ILFS) [215]
2. Infinite Feature Selection (InfFS) [216]
3. Eigenvector Centrality Feature Selection (ECFS) [217]
4. ReliefF [218]
5. Mutual information-based Feature Selection (mutinffs) [219]
6. Feature selection via concave minimization (FSV) [220]
7. Laplacian Score [221]
8. Multi-cluster Feature Selection (MCFS) [222]
9. Recursive Feature Elimination (RFE) [223]
10. L0 -norm feature selection (L0) [224]
11. Fisher score for feature selection [225]
12. L2,1-norm regularized discriminative feature selection (UDFS) [226]
13. Feature selection for local learning-based clustering (LLCFS) [227]

14. Feature ranking with correlation coefficients (CFS) [228]

15. Feature selection algorithm with adaptive structure learning (FSASL) [229]

16. Dependence Guided Unsupervised Feature Selection (DGUFS) [230]

17. Unsupervised Feature Selection with Ordinal Locality (UFSOL) [231]

18. Least Absolute Shrinkage and Selection Operator (LASSO) [232]

Infinite Latent Feature Selection uses graphs for ranking the features while considering all feature subsets using paths. It considers 'Relevancy' as the latent factor determining the rank allotted to each of the features. Infinite Feature Selection uses the properties of convergence of power series of matrices for selecting the features. Eigenvector Centrality Feature Selection methods give the importance to a feature in relation to the importance of its neighbors. Relief-F ranks features as per how they distinguish themselves from the nearest k instances. Mutual information-based feature selection selects features based on the mutual information that each feature carries. The maximum the mutual information carried by a feature, the higher the rank of the feature. In FSV, features lying on the normal vector of the plane separating other features are suppressed. Laplacian Score ranks feature according to their strength of locality preserving. MCFS selects features such that the multi-cluster structure of the data is not harmed. RFE eliminates features which have minimum ranking criterion according to an objective function. Fisher score selects feature based on inter and intraclass variance when the samples are projected along a single dimension. L2, 1-norm regularized discriminative feature selection selects

the most discriminative feature in batch mode. In LLCFS, weights are assigned to each feature depending on whether it is suitable for clustering. Weights of irrelevant features are shrunk towards zero. According to CFS, the most important feature is a higher predictive of the class than an irrelevant feature. FSASL selects features that preserve the structure of the data. DGUFS proposes a model to select features from a projection-free L2, 0-norm equality constraint. UFSOL selects features based on the preservation of the ordinal locality of original data. LASSO selects features stepwise, which have a strong relationship with the classes.

6.2.3 Image Annotation

There are two techniques for image annotation: Parametric and Non-parametric. Parametric techniques depend on multi-label characterization to produce the annotation tag. Non-parametric techniques use sparse reconstruction method and depend on reconstruction coefficients to produce accurate tags [93]. For image annotation, five algorithms have been studied. They are:

1. Deep Convolutional Ranking (DCR) [233]: Gong *et al.* use convolutional neural networks to produce annotation tags.

2. Multi-View Low-Rank Regression (MVLR) [234]: Zheng *et al.* use multi-view regression model. They impose low-rank constraints on their model for better performance.

3. Inductive Zero-Shot Image Annotation via Embedding Graph (IEG) [134]: Wang *et al.* use word embeddings instead of training the network on huge databases. A graph convolutional network is employed, which involves test images in training.

4. Structured Max-Margin Learning [235] (SMM): In this paper, an image is divided into several image patches and then using an automatic image labeling algorithm the image patches are assigned multiple labels. To cluster the image patches and determine kernel weights, a K-way min-max cut algorithm is used. Several base kernels are joined for solving the problem of intra-concept visual diversity. A visual concept network is built for the characterization of intra-concept visual similarity. This network is used for inter-related learning tasks directly in the feature space. The whole algorithm is built by using: max-margin Markov networks, visual concept network, and multitask learning.

5. Multi-Layer Group Sparse Coding [236] (MLG): A multi-layer group sparse structure of the reconstruction coefficients is proposed to capture the dependency between class labels and tags. A multi-layer group tag propagation method is used for transferring tags from images with similar tag distribution to test images.

The most important application of image annotation is keyword-based image searching in large databases. It plays an important role in content management and retrieval of images and videos [93]. Image annotation is widely used to annotate

medical images [113].

There are various challenges associated with the image annotation systems. Various methods for evaluation of annotations conflict with each other. Most of the annotations present are incomplete and inaccurate. Moreover, sometimes the image itself injects inaccuracy - like blur, distortion, and occlusion. There also exists ambiguity in the vocabulary of annotations [88]. Most of the works focus on reducing high-dimension feature to low-dimension rather than extracting meaningful semantic features [95]. Also, researchers do not use feature selection and tend to use the full feature set for annotation. This greatly reduces their performance [108].

Apart from the algorithms mentioned in this chapter, lots of research is going on with deep learning methods. Deep learning methods [16, 237–240] achieve far superior performance in the fields of salient object detection and image annotation. But these methods require long training time and expensive GPUs. Thus, they cannot perform well in a resource-constrained environment. Keeping in mind the resource constraints, this chapter proposes a model for salient object detection and image annotation using hand-crafted features and simple machine learning algorithms.

6.3 Method and Model

This chapter presents a new method of image annotation on four publicly available datasets. The procedure can be depicted by a flowchart, as shown in FIGURE 6.1. The salient object detection method is explained in Section 3.3.3. The remaining

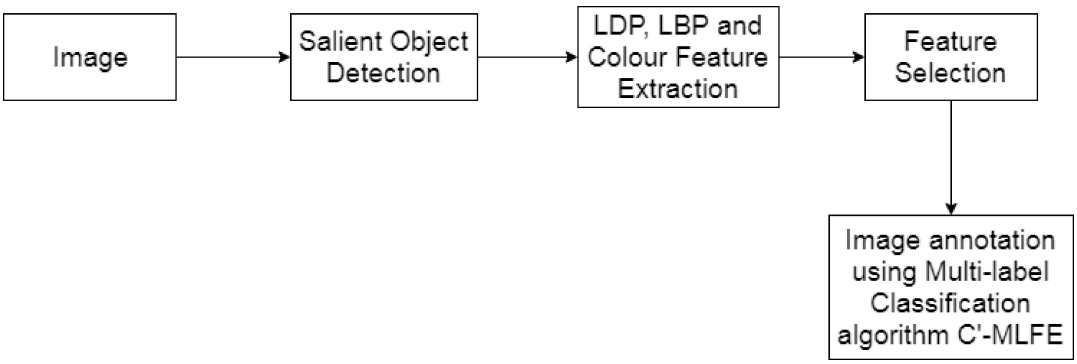

FIGURE 6.1: The flowchart depicting the procedure of image annotation.

steps of the procedure are described in the following subsections.

6.3.1 Extraction of LBP, LDP and Color Features

In this chapter, Local Binary Pattern [241], Local Derivative Pattern (LDP) [242], and Color Features are extracted from images for image annotation. LBP and LDP provide excellent texture features and is only next to SIFT in terms of use by researchers in the areas of computer vision. Colors are a dominant feature of an image and play a significant role in the computer vision field. The final feature set is obtained by concatenating features obtained from each of the above methods.

A 256 - dimensional feature factor is obtained by taking the histogram of local binary patterns of an image. The working of LBP is shown by the diagram in FIGURE 6.2. The top left matrix shows a 3×3 neighborhood for pixel value 5. The bottom left matrix shows the weight at each pixel location concerning pixel value 5. In the clockwise direction, weights are assigned in increasing power of 2. The weight matrix is multiplied element by element to the binary code matrix. The right bottom matrix

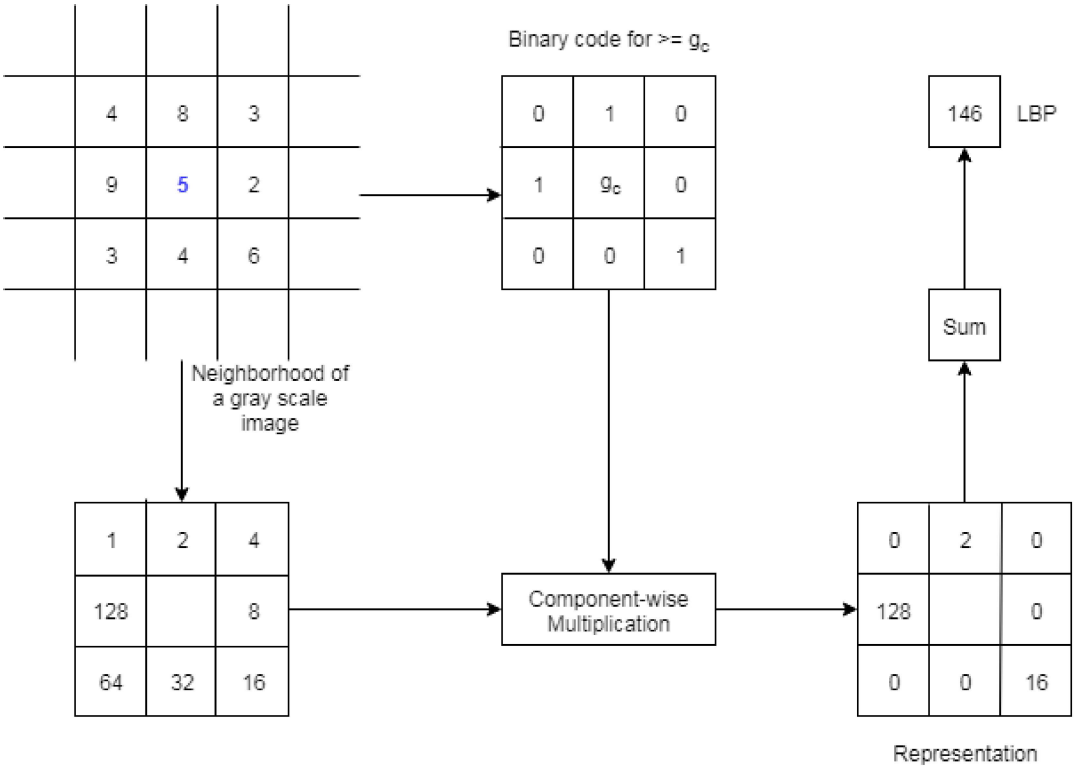

FIGURE 6.2: Working of LBP.

shows the result. LDP [242] was designed to obtain higher-order texture features as LBP can gather only first-order texture features. Histogram of LDP gives a 1024-dimensional feature vector.

Color is an important feature as it helps in fast and accurate object location and identification [243]. It is also invariant to rotation and partial occlusion. For obtaining the color histogram, the hue part of an HSV image is used. The count of pixels in each of the 256 bins is used as the color feature vector.

Combining features from all the above methods, a feature vector of dimension 1536 is obtained. After normalizing and removing columns of all zeros, the final feature vector is of length 1526.

6.3.2 Winner of Feature Selection Methods

For choosing the winner of feature selection methods, Technique for Order Performance [or Ordered Preference] by Similarity to Ideal Solution (TOPSIS) [244] method was used for ranking the performance of all the feature selection methods. It uses a distance-based approach to find the orders for participating candidates on multiple criteria. Here, the candidates are the participating feature selection algorithms, and the criteria are the evaluation parameters: Precision, Coverage, One-error, Ranking Loss, Micro-F1 Score, and Macro-F1 Score. A diagram representing the TOPSIS method of ranking is shown in FIGURE 6.3.

The performance matrix is the matrix formed by noting the performance value of the candidate for each criterion. Normalizing the matrix brings all the values in the range 0-1. The ideal and anti-ideal points are set to the possible best and worst value for criteria. Each criterion is assigned equal weights in the proposed model. For each candidate, we obtained the weighted distance from the ideal and anti-ideal points. The aggregate weight is calculated as the ratio of the distance from the anti-ideal point to the sum of distances. Higher the aggregate distance better the performance.

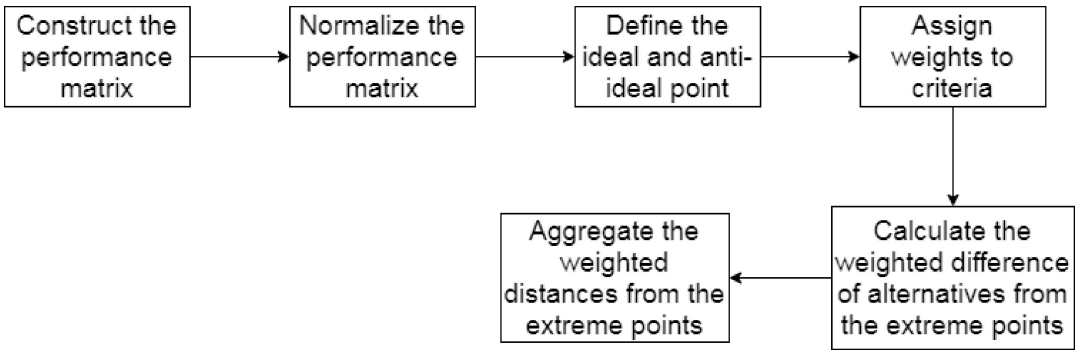

FIGURE 6.3: TOPSIS method of ranking.

6.3.3 Proposed subset of features: Double Sort Feature Selection

After studying the results of all the feature selection algorithms, an attempt is made to find the strongest features. For this, first, we find top 100 ranked features for all the above algorithms for each dataset. Therefore, for each dataset, we find 1800 strong features. This matrix is referred to as **F**. The next step is to find unique features from this set. The higher the frequency of a particular feature, the stronger it is deemed to be. In other words, if a particular feature is present in **F** nine times, it is considered to be a higher discriminator than a feature that occurs in **F** only once. In this way, we obtain nearly 835 powerful features for each database. The final step is when we select a subset of features that are deemed strong for all four databases. For this, we consider a feature to be strong if it is a strong discriminator for at least two databases. By this sorting technique, we obtain a subset of 465 features. Since the final subset is obtained by sorting the features twice, the procedure is termed as double sort.

The feature set is directly reduced to 30% of its original size with 3-4% better average precision for all datasets. This feature set is referred to as Double Sort Feature Set (DSFS). The results are shown in Table 6.5, 6.6 and 6.7.

6.3.4 Image Annotation

This model modifies the multi-label classification algorithm proposed by Zhang *et al.* [245]. The algorithm solves the problem of multi-label classification in which an instance is related to multiple label and topics. They approach the problem by finding structures in feature matrix of training samples. This rich information is added to the label information of the training set. Sparse reconstruction is used to characterize the structure present in the feature matrix. The enriched label information is used for training. The details of the algorithm are given in the following subsections.

1. **Structural Information Discovery**: A weighted directed graph $G = (V, E, Y)$ is defined where vertex set $V = \{z_k | 1 \leq k \leq r\}$ is the set of training instances. The edge set is defined as $E = \{(z_k, z_l) | y_{kl} \neq 0, 1 \leq k \neq l \leq r\}$. y_{kl} is the non-zero weight of the edge connecting z_k and z_l. The weight matrix $Y = [y_{kl}]_{(r \times r)}$ encodes the relationship between the training instances. y_{kl} determines the influence of y_k over y_l. This influence is calculated so as to reconstruct a sample from all other samples. Let $C_k = [z_1, z_2, z_3, \ldots, z_{k-1}, z_{k+1}, \ldots, z_r]$ be the set of training instances for constructing z_k and $x_k = [y_{1k}, \ldots, y_{(k-1)k}, y_{(k+1)k}, \ldots, y_{rk}]^T$

be the weight matrix. x_k is obtained from the solution of the following optimization problem:

$$\min_{x_k} \|C_k x_k - z_k\|_2^2 + \mu \|x_k\|_1 \tag{6.1}$$

2. **Information enrichment of training labels**: Let $A = \{a_1, a_2, a_3, \ldots, a_s\}$ define the label space containing s class labels. G represents the training set given by $G = \{(z_k, A_k) | 1 \leq k \leq r\}$ where $A_k \subseteq A$ defines the label subset for z_k. For every training instance a label vector is defined in the following form: $v_k = (v_{k1}, v_{k2}, v_{k3}, \ldots, v_s)^T$ where $v_{km} = 1$ if $a_m \in A_k$ otherwise it is -1. Using the structure information obtained in the previous step, the binary labeling vector is transformed into a numerical one as $w_k = (w_{k1}, w_{k2}, w_{k3}, \ldots, w_{ks})^T$. The new label space is given as $w_k = [w_1, w_2, w_3, \ldots, w_r]$. To obtain Z, solution of the following optimization problem should be found:

$$\min_{Z} \sum_{k=1}^{r} \left\| w_k - \sum_{l=1}^{r} y_{lk} w_l \right\|_2^2 s.t. \quad d_1 \leq v_{kl}, w_{kl} \leq d_2 \, (1 \leq k \leq r, 1 \leq l \leq s) \tag{6.2}$$

d_1 and d_2 are set as 1 and 2 respectively by author's choice. The predicted model is generated by using multi-output regression techniques. Multi-regression SVM is used for non-linear modeling.

3. **C'-MLFE: Proposed modification to C-MLFE [245]**: The modification to C-MLFE is proposed at the stage of generation of weight matrix. The

weight matrix in C-MLFE is generated to find the influence of each instance on another. But there are cases when samples do not influence each other in any manner. For example an image containing shoes will have no influence on images containing train. In such cases calculating influence may not only hamper the result but also wastes enough time for large databases. To overcome this problem, the dataset is clustered into two. Each cluster contains similar images and then the influence of training samples is calculated only for the images in the same cluster. Thus, two weight matrices are generated. Generating more clusters, will unnecessary increase the complexity of the algorithm. The matrix can be generated in parallel thereby saving time. K-Means is used for clustering with a squared Euclidean distance metric. The algorithm was tested for four other distance metrics - city block, cosine, correlation and hamming. None of these metrics provided better results than squared Euclidean metrics.

6.3.5 Algorithm of the proposed image annotation model

The algorithms for the implementation of the proposed model is given below. Algorithm 2 in Chapter 3 shows the procedure for salient object detection. For image annotation, the procedure is given by Algorithm 4.

Algorithm 4: Image Annotation using Feature Selection and C'-MLFE

Input: Salient Object with relevant background
Output: Annotated Images
1 Obtain the salient object from Algorithm 2 in Chapter 3;
2 Extract LBP, LDP, and color feature for the image;
3 Select DSFS feature from the feature vector obtained in step 2;
4 Use C'-MLFE for multi-label classification;

6.4 Results

This section explains the results obtained by the proposed model. Subsection 6.4.1 shows the saliency results obtained by the proposed model. The scene analysis results are shown in subsection 6.4.2. Subsection 6.4.3 deals with the feature selection results. The double sort feature selection results are given in subsection 6.4.4. In subsection 6.4.5, image annotation results are discussed. Failure cases are discussed in subsection 6.4.6.

6.4.1 Salient Object Detection Results

A comparison of the proposed algorithm with other existing procedures is shown by performance evaluation graphs in FIGURE 6.4. The performance of the other algorithms is evaluated on a sample set of 1000 images obtained from the three datasets (DUT-OMRON, MSRA10K, and PASCAL-S). In FIGURE 6.4(a), the precision of the proposed algorithm is less than DSR and FES, but the recall of the proposed algorithm is much better than theirs. It is because the saliency map built by using minimum directional contrast rarely misses the foreground object. A similar

explanation applies to high precision and low recall values of GR, MC, and PCA in FIGURE 6.4(b) and for RBD in FIGURE 6.4(c). In FIGURE 6.4(d), the high precision value of SUN has no meaning because of extremely low recall value.

In FIGURE 6.4 (f), the performance of the proposed algorithm on three datasets is shown.

FIGURE 6.5 shows the result of applying the proposed algorithm to some sample images from the database.

6.4.2 Scene Analysis Results

Here, the role of factor θ is shown. FIGURE 6.6 shows how more and more background is brought into context by increasing the value of θ. The image shows the variation of adding background details from 1% to 96% in steps of 8. In none of the previous works, such a factor was observed which can control background information.

6.4.3 Feature Selection Results

For each of the databases, we ranked 1526 features using all the above algorithms. For each feature selection method, we obtained results for choosing 1 to 1526 features in steps of 100. the algorithms were then ranked based on precision, one-error, coverage, ranking loss, macro F1, and micro F1 score.

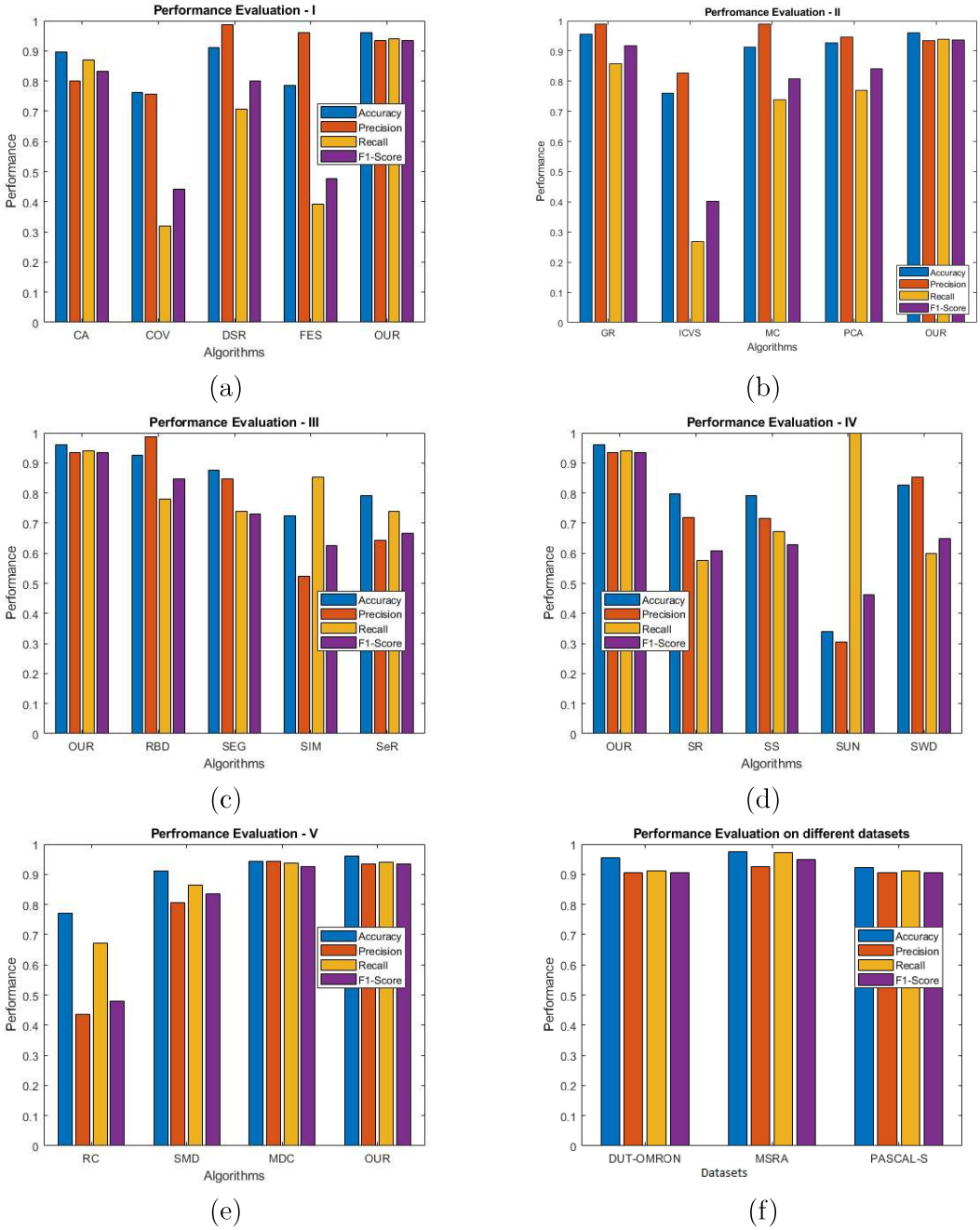

FIGURE 6.4: Performance evaluation graphs (a) Comparison with CA [201], COV [202], DSR [246] and FES [203] (b) Comparison with GR [204], ICVS [205], MC [206] and PCA [184] (c) Comparison with RBD [207], SEG [208], SIM [209] and SeR [210] (d) Comparison with SR [211], SS [212], SUN [213] and SWD [214] (e) Comparison with RC [166], SMD [43] and MDC [177] (f) Performance evaluation of the proposed algorithm on different datasets.

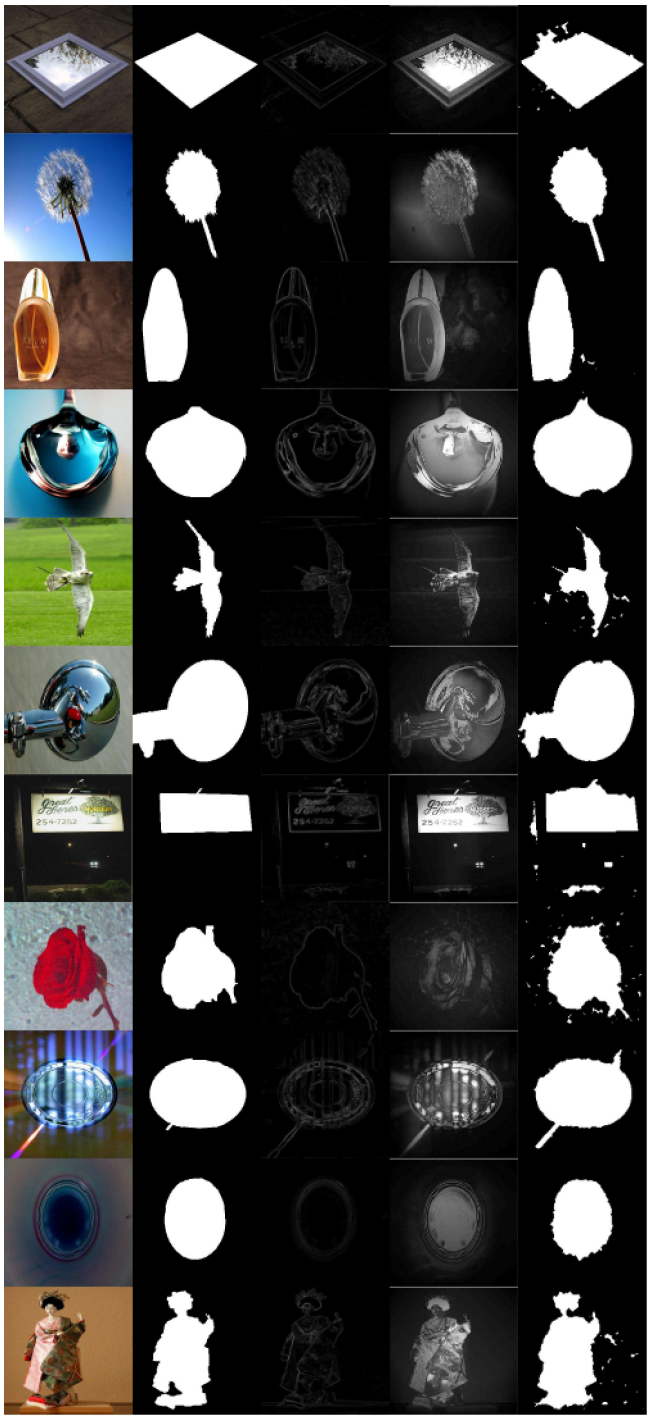

FIGURE 6.5: Results of Salient object detection for sample images from the database. The first column is the input image. The second column is the ground truth. The third column is the edge map. The fourth column is the saliency map. The last column is the output.

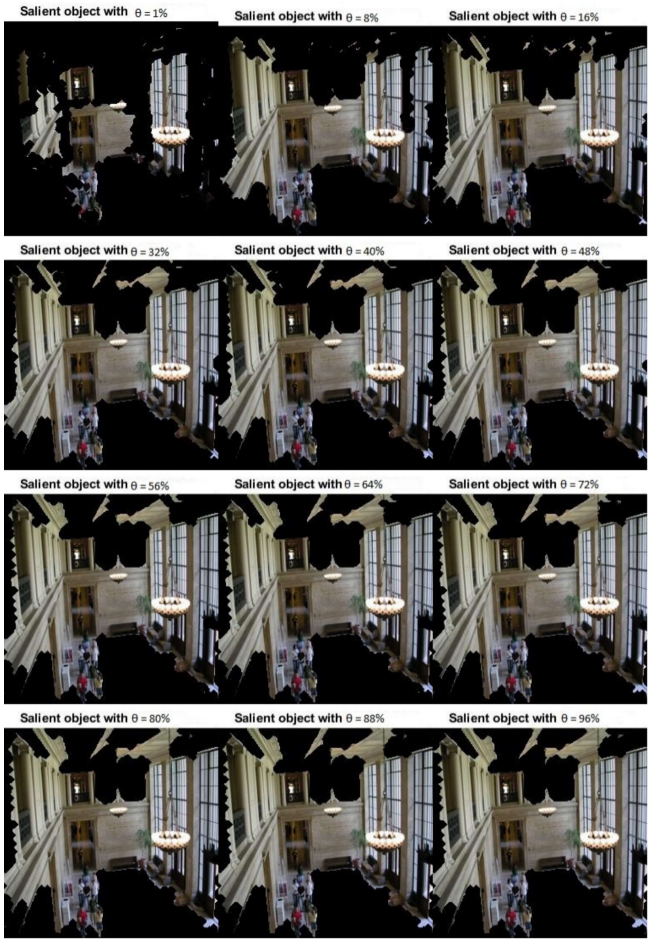

FIGURE 6.6: Result of increasing θ in the proposed algorithm.

For example, for the UIUC Sports dataset, the model obtained the following result, as shown in TABLE 6.1 for the ILFS algorithm. The upward-pointing arrow means higher the value better the performance. The downward pointing arrow means lower the value, the better the performance. The best results are marked in bold.

In a similar way, we accumulate results for each of the other feature ranking algorithms. In the next step, the overall results for each of the algorithm are combined to see which algorithm gives the best results in terms of evaluation parameters. TABLE 6.2 shows the results for all the candidate algorithms.

TABLE 6.1: The table shows the performance of the ILFS algorithm on UIUC dataset. The top row determines the evaluation parameters. The first column determines the number of features used.

	Precision ↑	**One-Error ↓**	**Coverage ↓**	**Ranking Loss ↓**	**Mac F1 ↑**	**Mic F1 ↑**
1	0.7621	0.1273	6.2101	0.1636	0.1309	0.6777
100	0.8528	0.0700	4.0254	0.0822	0.4483	0.7427
200	0.8764	0.0700	3.5668	0.0625	0.5034	**0.7766**
300	0.8801	0.0828	3.4777	0.0592	0.4999	0.7663
400	0.8786	0.0700	3.4968	0.0606	**0.5048**	0.7713
500	0.8795	0.0764	3.5095	0.0609	0.4956	0.7713
600	0.8800	0.0764	3.4267	0.0586	0.4831	0.7635
700	0.8775	0.0764	3.5095	0.0606	0.4697	0.7642
800	0.8780	**0.0636**	3.4777	0.0600	0.4549	0.7585
900	0.8787	0.0700	3.4585	0.0588	0.4614	0.7602
1000	0.8815	0.0636	3.4203	0.0569	0.4505	0.7634
1100	0.8816	0.0764	3.3757	0.0555	0.4544	0.7670
1200	0.8818	0.0764	3.3566	0.0553	0.4563	0.7622
1300	0.8820	0.0764	3.3503	0.0552	0.4544	0.7636
1400	0.8814	0.0764	3.3566	0.0555	0.4573	0.7661
1500	**0.8836**	0.0764	**3.2929**	**0.0530**	0.4681	0.7622

In the third step, we rank each of the algorithm's performance based on TOPSIS method of multiple criteria ranking.

Similarly, we find TOPSIS ranking for all the algorithms for each of the database. We obtain the following ranking for UIUC Sports, NUS-WIDE, MSRC2, and LableMe database. The results are shown in TABLE 6.3.

In the fourth step, we again apply TOPSIS ranking to find which algorithm performs best for all databases. The TABLE 6.4 ranks the algorithm in terms of their performance. 1st rank means higher performance.

Fisher gives the best result for all algorithms for a selection of top 1000 features.

TABLE 6.2: The table depicts the performance of various feature selection algorithms on UIUC sports dataset.

	Precision ↑	**One-Error ↓**	**Coverage ↓**	**Ranking Loss ↓**	**Mac F1 ↑**	**Mic F1 ↑**
ILFS	0.8836	0.0636	3.2929	0.0530	0.5048	0.7766
InfFS	0.8848	0.0636	3.2738	0.0523	0.4757	0.7642
ECFS	0.8833	0.0636	3.2866	0.0528	0.4840	0.7649
relieff	0.8868	0.0636	3.3121	0.0536	0.5080	0.7784
mutinffs	0.8894	0.0700	3.2802	0.0523	0.4901	0.7717
fsv	0.8941	0.0636	3.2611	0.0511	**0.5352**	0.7880
laplacian	0.8831	0.0700	3.2929	0.0532	0.4696	0.7644
mcfs	0.8890	0.0636	3.2484	0.0517	0.5117	0.7816
rfe	0.8875	0.0636	3.2611	0.0517	0.5164	0.7762
L0	0.8906	0.0636	**3.2356**	0.0515	0.5200	0.7791
fisher	0.8899	0.0700	3.2675	0.0515	0.4911	0.7717
UDFS	0.8883	0.0700	3.2420	0.0515	0.5177	**0.7881**
llcfs	0.8836	**0.0573**	3.2929	0.0528	0.5002	0.7725
cfs	0.8886	**0.0573**	3.2738	0.0511	0.5021	0.7736
fsasl	0.8832	0.0636	3.2802	0.0528	0.4602	0.7643
dgufs	0.8875	0.0636	3.2611	0.0517	0.5164	0.7762
ufsol	**0.8947**	0.0636	3.2420	**0.0505**	0.5163	0.7791
lasso	0.8894	**0.0573**	3.2547	0.0521	0.5278	0.7724

6.4.4 Double Sort Feature Selection Results

The results of applying the proposed algorithm on different datasets are shown in TABLE 6.5, 6.6 and 6.7. By using feature selection, the model obtains a speedup of at least 2, and the average precision has increased by nearly 4% for all datasets. It can also be seen that Fisher feature selection gives a maximum speedup of 1.45 with 1-2% increase in accuracy.

TABLE 6.3: Ranking the performance of various feature selection algorithms for the four datasets.

Methods \ Datasets	**UIUC Sports**	**NUS-WIDE**	**MSRC2**	**LabelMe**
cfs	7	8	4	4
dgufs	8	9	2	13
ECFS	15	7	13	14
fisher	9	3	6	2
fsasl	16	14	17	5
fsv	1	17	5	1
ILFS	12	1	7	18
InfFS	14	13	11	10
L0	3	6	1	12
laplacian	17	12	16	3
lasso	4	11	3	6
llcfs	10	2	9	16
mcfs	5	16	10	15
mutinffs	11	15	12	17
relieff	13	4	14	8
rfe	8	10	2	7
UDFS	6	18	8	9
ufsol	2	5	15	11

6.4.5 Image Annotation Results

For comparison of the result, SMM and MLG algorithms were studied. It can be seen that for all the four databases, the proposed procedure performs much better than these algorithms. Another algorithm DCR [233] has 32% precision on the NUS-WIDE dataset. The model MVLR [234] shows 37.7% precision on NUS-WIDE. IEG [134] obtains a maximum of 29.72% precision on the NUS-WIDE dataset.

For a visual result of the proposed procedure, FIGURE 6.7 is given. Sample images are taken for databases. The third column shows the result with image tags. A comparison of image annotation methods using various salient object detection algorithms proposed in this thesis is also done. The results are shown in TABLE 6.10.

TABLE 6.4: Ranking various feature selection algorithms on the basis of their performance on all four datasets.

Algorithms	**TOPSIS rankings**
Fisher	1
L0	2
CFS	3
FSV	4
LASSO	5
RFE	6
DGUFS	7
UFSOL	8
LLCFS	9
ILFS	10
RELIEFF	11
UDFS	12
MCFS	13
InfFS	14
Laplacian	15
ECFS	16
FSASL	17
MutInfFS	18

TABLE 6.5: Performance of the proposed algorithm without feature selection

	Without Feature Selection					
	One Error ↓	Coverage ↓	Ranking Loss ↓	Average Precision ↑	MacF1 ↑	MicF1 ↑
LabelME	0.2697	8.955	0.2050	0.6648	0.2714	0.4955
MSRC	0.1702	3.5540	0.0680	0.8061	0.4989	0.6454
NUS-WIDE	0.7099	9.6011	0.1946	0.3452	0.4451	0.0202
UIUC Sports	0.0595	3.2172	0.0511	0.8994	0.5040	0.7991

It is clearly visible that the improvement is seen from statistical to deep learning models.

6.4.6 Failure Cases

The proposed model fails a few times, as shown in FIGURE 6.8. In the top left image, the pattern of two cycles overlay each other and might have resulted in ambiguous

TABLE 6.6: Performance of the proposed algorithm with Fisher feature selection

	With Fisher Feature Selection						
	One Error ↓	Coverage ↓	Ranking Loss ↓	Average Precision↑	MacF1 ↑	MicF1 ↑	Speedup Obtained
LabelME	0.2	8.85	0.2043	0.6846	0.2696	0.496	1.0678
MSRC	0.1186	3.5254	0.0547	0.8396	0.4577	0.6434	1.0284
NUS-WIDE	0.7241	10	0.1841	0.3575	0.4109	0	1.3081
UIUC Sports	0.0443	3.0063	0.0406	0.9181	0.4951	0.8312	1.4550

TABLE 6.7: Performance of the proposed algorithm with the proposed Double Sort feature selection

	With Double Sort Feature Selection						
	One Error ↓	Coverage ↓	Ranking Loss ↓	Average Precision↑	MacF1 ↑	MicF1 ↑	Speedup Obtained
LabelME	0.3157	7.75	0.1608	0.7083	0.2384	0.5496	3.5288
MSRC	0.1166	3.7833	0.0608	0.8423	0.5345	0.6461	1.9174
NUS-WIDE	0.6666	10.3461	0.2172	0.3854	0.4306	0.0380	1.8880
UIUC Sports	0.0189	3.0886	0.0412	0.9205	0.4910	0.82687	4.3329

TABLE 6.8: Comparison with MMG algorithm result on three databases.

	Precision		
Methods \ Datasets	LabelME	NUS-WIDE	UIUC Sports
MLG	55.21	11.38	66.11
Proposed Method	70.83	38.54	92.05

TABLE 6.9: Comparison of precision with SMM algorithm result on MSRC database.

Methods \ Dataset	MSRC
SMM	76.9
Proposed Method	84.2

TABLE 6.10: Image Annotation Result by using various Salient Object Detection Methods

	Precision			
Methods \ Dataset	LabelMe	NUS-WIDE	UIUC Sports	MSRC
GVF+MDC	70.83	38.54	92.05	84.2
GaborFilters	72.17	39.23	93.98	86.05
YOLOv2	74.99	40.72	97.83	89.66

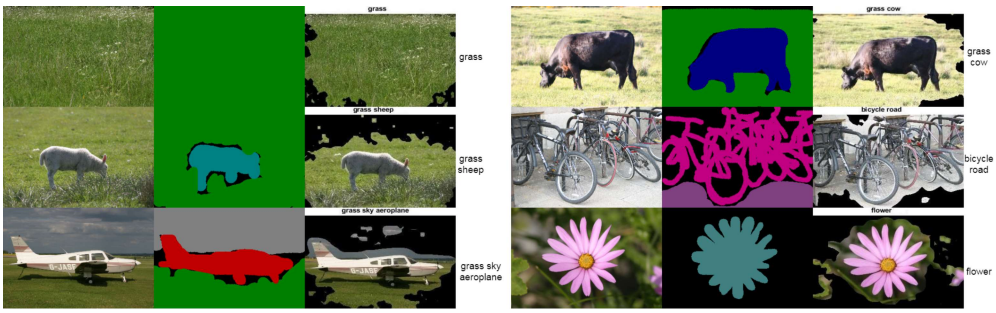

FIGURE 6.7: Results of the proposed image annotation algorithm. The first column shows the input image. The second column shows the ground truth. The third column shows annotated images.

features. In the bottom right image, the boat has been left out of annotation. The reason is that it occupies a tiny area in the whole scene. Mixing of tree colors with background shadow of black causes feature ambiguity. The missing annotation in the top right corner might be due to pattern ambiguity because of matching color in the database with the cow category. Thus, much effort still needs to be put in the proposed model for removing intra-class variation and overcoming background clutter.

6.5 Conclusion

In this chapter, an attempt was made to carry out the process of image annotation using a modular approach. First, the salient object was detected using a fusion of GVF snakes and minimum directional contrast. After the object is detected, LBP, LDP, and color features are extracted for the object. A 1526 dimensional feature vector was obtained. Then, using the DoubleSort feature selection technique, 456

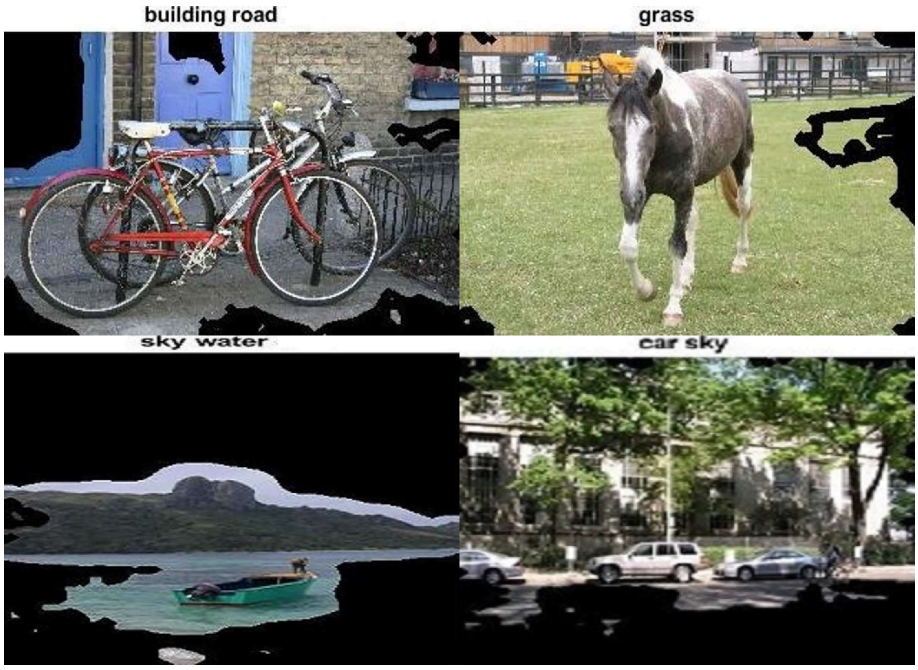

FIGURE 6.8: Few examples of failure cases of the proposed algorithm.

reliable discriminators are obtained. This feature vector was then fed to the multi-label classifier C'-MLFE, which produced tags as outputs. Salient object detection results were evaluated on three publicly available datasets. For the evaluation of image annotation, four public datasets were used. Eighteen feature selection methods were studied to find the best feature selection technique for image annotation datasets. The results obtained using the procedure mentioned in this chapter were better than the works introduced earlier.

Moreover, the proposed DoubleSort technique brought the feature vector down to 30% with a 3-4% increase in accuracy of results. This was a tremendous benefit for saving the computation time of the classification algorithm. For the future, the aim is to improve upon the multi-label classification technique. The weight matrix

requires much time to build and is the bottleneck for achieving faster results. Comparison of image annotation methods by using the salient object detection models proposed in previous chapters also justify that salient object detection increases the accuracy of image annottaion results.