

# Harmonic Mean based Classification of Images using Weighted Nearest Neighbor for Tagging

Mrs. Anupama D. Dondekar<sup>1</sup>, Mr. Balwant A. Sonkamble<sup>2</sup>  
Computer Engineering Department, Pune Institute of Computer Technology  
Pune, India

**Abstract**—On image sharing websites, the images are associated with the tags. These tags play a very important role in an image retrieval system. So, it is necessary to recommend accurate tags for the images. Also, it is very important to design and develop an effective classifier that classifies images into various semantic categories which is the necessary step towards tag recommendation for the images. The performance of existing tag recommendation based on k nearest neighbor methods can be affected due to the number of k neighbors, distance measures, majority voting irrespective of the class and outlier present in the k-neighbors. To increase the accuracy of the classification and to overcome the issues in existing k nearest neighbor methods, the Harmonic Mean based Weighted Nearest Neighbor (HM-WNN) classifier is proposed for the classification of images. Given an input image, the HM-WNN determines k nearest neighbors from each category for color and texture features separately over the entire training set. The weights are assigned to the closest neighbor from each category so that reliable neighbors contribute more to the accuracy of classification. Finally, the categorical harmonic means of k nearest neighbors are determined and classify an input image into the category with a minimum mean. The experimentation is done on a self-generated dataset. The result shows that the HM-WNN gives 88.01% accuracy in comparison with existing k-nearest neighbor methods.

**Keywords**—Image classification; k-nearest neighbor; weighted nearest neighbor; harmonic mean vector; color and texture features

## I. INTRODUCTION

Classification is a supervised method that categorizes the unknown data into a specific class or group. In classification, the labels or classes are known in advance. Image classification is used for indexing, categorization and annotation of the images. To classify images into various categories, the features of the images are extracted using different feature extraction techniques and stored in the database with the label. For training the classifier, the images stored in the database are divided into two sets—training and testing images. The classifier learns from the labeled training images stored in a database and predicts the class label of an unlabeled test image.

The k nearest neighbor (kNN) is a memory-based and non-parametric classification algorithm where the algorithm memorizes the training samples to predict the class label of an unknown sample. It is the most widely used classifier in machine learning because it is a very simple, effective and powerful recognition algorithm [1]. In a traditional k nearest

neighbor classifier the similarity between unknown sample and training samples are calculated. After similarity calculation, it selects k training samples that are closest to an unknown sample. The class label of an unknown sample is predicted by identifying the dominant category from k training samples. For kNN, the value of the k needs to be specified in advance and it denotes the number of samples that are nearest to an unknown sample. If the k value is too small it is susceptible to overfitting and would result in misclassification.

The steps of the kNN classifier algorithm are as follows:

- 1) Determine the value of k (no of neighbors).
- 2) Extract features of a test sample and calculate the distance between the feature vector of the training samples and test sample.
- 3) Sort the samples as per the distance in ascending order and select the first top k samples.
- 4) Get the majority voting from top k samples.
- 5) Predict the class with the majority voting.

However, the traditional k nearest neighbor classifier has the following limitations [2]:

- 1) It uses only a training set for the classifier model generation. It needs retraining if there is a change in training data.
- 2) It does not consider the situation in which the distribution of samples may not be equal. All training samples have equal weights.
- 3) Need to calculate the similarity between all training samples. It takes more time if the training set consists of a large number of samples.
- 4) The performance of the classifier depends on the number of neighbors (k). The small value of k affects classification results due to noisy and inaccurate samples. The large value of k degrades the performance of classifier due to outliers which comes from incorrect category in the k nearest neighbors.
- 5) It suffers from outliers if the size of the training sample size is small.

To determine the correct value of k and to reduce the sensitivity of the k value, the authors have proposed adaptive methods to determine the neighborhood for different input samples in [2, 3, 4]. In most of the kNN algorithms, all samples have equal weight but each neighbor may contribute differently in the classification. In [5, 6, 7] the method was

proposed where each nearest sample was assigned varying weights. In [8] the modified kNN method was proposed by assigning weight to the neighbor based on the local value of  $k$ . The weighted kNN was used in [9] where inverse Euclidean distance assigned as the weight to each neighbor. But, no weighting scheme perform consistently well under some condition.

To handle the outlier problem, the LMKNN classifier was proposed in [10]. The LMKNN classifier determines the neighbor of an input sample based on the difference between the mean values of the nearest training samples and an input sample. It has good classification accuracy but it has some limitations: the same value of  $k$  is used for different classes and all neighbors have uniform weights. To improve the accuracy of the LMKNN method, many local mean-based methods for classification have been proposed. In [11] the PNN method was proposed for classification based on the similarity between the weighted distance of an input sample and pseudo training samples. The LMPNN classifier was proposed in [12] where the weights are assigned to each local mean vector and distance was calculated between mean vector and pseudo neighbors of every class. The MLM-KHNN classification method was proposed in [13] which has used harmonic mean distance instead of Euclidean distance. The LMRKNN method was proposed in [14] where the input sample is represented as a combination of the mean vector of each category and weights are assigned to each local mean vector. In [15] GMDKNN method was proposed which predict the class label based on nested generalized mean distance.

To improve the accuracy of kNN based classification, the harmonic mean based weighted nearest neighbor classifier is proposed which is insensitive to the value of  $k$ . Instead of assigning weights to the local mean vector, the weights are assigned to the  $k$  nearest neighbor of each class for each feature instead of combined features. The reason behind using separate classifier for each feature is that it may contribute differently to determine number of neighbors depends on input sample which affects the classification accuracy. Also, the harmonic mean is used as it gives more importance to the neighbors closer to the given input image as compared to arithmetic mean.

The work done carried out in the paper can be summarized as follows:

- 1) In HM-WNN classifier, the separate kNN classifiers are trained for color and texture features and harmonic mean is determined for each category for every feature to reduce the impact of the choice of value  $k$ .
- 2) The HM-WNN classifier assigns different weights to each neighbor of each class instead of uniform weights to reduce the classification error rate.

The structure of the paper is organized as follows: a review of the existing algorithms for classification of the images is described in section 2, section 3 describes feature extraction technique of the images, HM-WNN proposed classifier is explained in section 4, section 5 and section 6 describes performance metric used to check classifier performance and

dataset used for experimentation respectively. The classification results are discussed in section 7. Finally, the conclusion and future work is described in section 8.

## II. RELATED WORK

The method was proposed for image annotation refinement using a two-pass kNN classifier and group sparse reconstruction algorithm in [16]. The method for annotation of images using visual attention mechanism and SVM particle swarm optimization was presented in [17]. The method to suggest tags for an image based on visual features and tag correlations using a neighbor voting scheme was proposed in [18]. The kNN algorithm is used to find visual neighbors of a given image. The approach for retagging the social images with diverse semantics was presented in [19]. The relevance of a tag to an image was determined using three approaches. The kNN classifier was used to map the tag from textual space to low-level feature space by representing the tag with a set of images containing the tag. The method was proposed for annotation of images using multiple SVM classifiers which combine different visual features and improve the image annotation performance in [20]. The histogram intersection distance was used as the kernel function to reduce the impact of intra-class variations and interclass similarities. A personalized image tag recommendation approach by using users' history was presented in [21]. The approach recommends tags to the users by counting the occurrence of each tag from the visually similar images where the similarities between the images are determined using kNN classifier. In [22] the method was proposed which suggests the tags for labeled as well as unlabeled images. The method identifies similar images based on visual features solely using kNN classifier. A system that classifies images as correct or incorrect with respect to a given tag from a database of annotated images was defined in [23, 24]. The system trained SVM classifier per tag to enable fast classification. In [25] 'SheepDog' method was proposed which adds photos into the proper group and also recommends suitable tags for the photos of the Flickr dataset. To predict correct tags of a photo the concepts were detected using SVM classifier. The method for annotation of images using random forest is presented in [26]. The methods find semantic neighbors of a query image from each leaf node and annotate the images using tags of semantic neighbors.

Most of the existing methods for image tag recommendation used a single classifier for the recommendation of tags. A little work is done on the selection of suitable classifier for the purpose of tag recommendation of the images. Also, the tag recommendation methods based on  $k$  nearest neighbor has used uniform weight and the same number of neighbors. In the paper, a new classifier is proposed and compared with the existing algorithms as a necessary step in an image tag recommendation.

## III. IMAGE FEATURE EXTRACTION

Color and texture are the most widely used features in retrieval of the images. Color moments are used in color feature representation. The advantage of the color moment method is that it works better when the some region of the two images are similar because the low order moments of the

corresponding region of two images will be different and the similarity score will be small. The color feature extraction is done as follows: i) convert color space of an image into  $L^*a^*b^*$  color space ii) divide an image into 2 by 2 blocks. Also, the central image is obtained of the same size as 2 by 2 blocks. ii) The first, second and third moment of an each block are taken as feature vector. For texture feature extraction, the image is decomposed into sub-bands using wavelet packet transform up to level 3 and calculates the energy and its variation of each sub-band at the last level as a feature vector. Both features are combined by assigning weight to them [27]. The color and texture feature values are not in the same scale. Both features are normalized using Min-Max normalization to scale the values in range 0 and 1.

#### IV. HM-WNN CLASSIFICATION METHOD

In this section HM-WNN classification is described. The main objective of the proposed algorithm is to improve classification accuracy and to assign different weights to each neighbors of different category to reduce the sensitivity to outlier.

##### A. Classification

Let  $CF = (x_i, C_i)_1^{NF}$  and  $TF = (y_i, C_i)_1^{NF}$  represents color and texture features vector of  $NF$  training images with  $N$  number of classes. Let  $C_1, C_2, \dots, C_N$  represent the category. Let  $CF_j = (x_i^j, C_i^j)_1^{NF_j}$  and  $TF_j = (y_i^j, C_i^j)_1^{NF_j}$  denotes color and texture feature vector of  $j^{th}$  category with  $NF_j$  training images. In HM-WMM, given an input image first  $m$  nearest neighbors are determined from  $CF_j$  and  $TF_j$  for each category instead of finding  $m$  nearest neighbor from entire  $NF$  number of training images. The weights are assigned to each neighbor and the harmonic mean is calculated for each category for each feature and combined. Finally, the category  $j$  is assigned to an input image with minimum mean.

The steps for prediction of the class of an input image  $I$  are as follows:

- 1) Determine the nearest/closest neighbors for each class  $C_j$  from  $CF_j$  and arrange  $m$  closest neighbors in increasing order according to the distance measure.
- 2) Determine  $n$  nearest/closest neighbors for each class  $C_j$  from  $TF_j$  and arrange  $n$  closest neighbors in increasing order according to the distance measure.
- 3) Allocate weight to each neighbor of  $m$  and  $n$  as follows:

$$w_i^j = \frac{1}{i} \quad i = 1, \dots, z \quad \& \quad z \in (m, n) \quad (1)$$

Where  $w_i^j$  represents weight assigned to  $i^{th}$  neighbor of  $j^{th}$  class. It gives more weight to the samples which are nearby and less weight to the samples which are farther away.

- 4) Compute the harmonic mean of  $m$  and  $n$  closest neighbor of color and texture for each class  $C_i$ .
- 5) Combine the mean of each class determined for color and texture features to form final mean.
- 6) Assign the class label  $C$  to  $I$  with minimum mean distance belongs to among all classes.

##### B. HM-WNN Algorithm

As explained in section A, the proposed HM-WNN method is brief in Algorithm 1.

---

**Algorithm 1:** Harmonic Mean based weighted Nearest Neighbor classifier

---

Input:

CI: Input image color feature vector

TI: Input image texture feature vector

$TF_j = (y_i^j, C_i^j)_1^{NF_j}$ : Texture feature vector of training images of the  $j^{th}$  class

$CF_j = (x_i^j, C_i^j)_1^{NF_j}$ : Color feature vector of training images of the  $j^{th}$  class

$M = C_1, C_2, \dots, C_M$ : the number of class labels

Result: Prediction of class  $C_j$  of an input image  $I$

Step 1: Calculate the distance between CI and  $CF_j$ . Find  $m$  number of nearest neighbor for each class  $C_j$  according to the ascending order of distances to CI. Denote it by  $CD_j$ .

for  $i= 1$  to  $NF_j$

$$dist(CI, x_i^j) = \sqrt{(CI_1 - x_{i1}^j)^2 + \dots + (CI_f - x_{if}^j)^2} \quad (2)$$

end for

where  $x_i^j \in CF_j$

Step 2: Calculate the distance between TI and  $TF_j$ . Find  $n$  number of nearest neighbor for each class  $C_j$  according to the ascending order of distances to TI. Denote it by  $TD_j$ .

for  $i= 1$  to  $NF_j$

$$dist(TI, y_i^j) = \sqrt{(TI_1 - y_{i1}^j)^2 + \dots + (TI_f - y_{if}^j)^2} \quad (3)$$

end for

Where  $y_i^j \in TF_j$

Step 3: Assign weight to the  $i^{th}$  nearest neighbor of  $j^{th}$  class for  $CD_j$  and  $TD_j$  using Eq. 1 as follows:

$$Cdist(CI, x_i^j) = dist(CI, x_i^j) * w_i^j \quad (4)$$

$$Tdist(TI, y_i^j) = dist(TI, y_i^j) * w_i^j \quad (5)$$

Step 4: Compute the harmonic mean of first  $m$  and  $n$  nearest neighbor of  $CD_j$  and  $TD_j$  for the  $j^{th}$  class

$$CD\_Mean_j = \frac{m}{\sum_{i=1}^m \frac{1}{Cdist(CI, x_i^j)}} \quad (6)$$

Where  $CD\_Mean_j$  represents color feature mean of class  $C_j$

$$TD\_Mean_j = \frac{n}{\sum_{i=1}^n \frac{1}{Tdist(TI, y_i^j)}} \quad (7)$$

Where  $TD\_Mean_j$  represents texture feature mean of class  $C_j$

Step 5: Calculate the final mean vector as follows:

$$F\_Mean_j = CD\_Mean_j * TD\_Mean_j \quad (8)$$

Step 6: Assign  $I$  a class label  $C$  which has smallest mean calculated for each class as follows:

$$C = \min_{c_j}(F\_Mean_j, I) \quad (9)$$

### V. PERFORMANCE MATRICES

The performance of the classifier is evaluated using confusion matrix on a test data. The confusion matrix analyses the test data to determine how classifier identifies test samples of different categories. It consists of count of predicted values and actual values. By using these count of values the precision, recall, F1-score and accuracy are determined to estimate the performance score of the classifier.

$$Precision = \frac{TP}{TP+FP} \tag{10}$$

$$Recall = \frac{TP}{TP+FN} \tag{11}$$

$$F1 - Score = \frac{2*precision*recall}{precision+recall} \tag{12}$$

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{13}$$

TP = Test Sample is positive and predicted positive.

TN = Test Sample is negative and predicted negative.

FP = Test Sample is negative and predicted positive.

FN = Test Sample is positive and predicted negative.

### VI. DATASET

For experimentation the images are downloaded from Flickr image sharing website using public API. The images are belongs to six different categories actor, clover, fish, autumn, butterfly and aeroplane. Each category consists of 300 images. The 70% of the images are used for training and 30% are used for testing purpose.

### VII. EXPERIMENTAL RESULTS

Tables I and II shows the precision, recall, F1-score and accuracy of different classification methods: SVM, traditional kNN, LMKNN, PNN, LMPNN and proposed method HM-WNN.

To implement SVM,  $m*(m-1)/2$  binary classifiers are composed where m is the no of classes/categories and  $m=6$  and one vs. all method is used. In LMKNN, k neighbors are determined for each category and calculated the mean vector of each class [6]. The class label of an input sample is predicted having minimum distance between mean vectors of each among all classes. In PNN, weights are assigned to each neighbor and assign the class of neighbor nearest to an input sample [7]. In LMPNN method, first local mean vector is determined for each category and weights are assigned to each local mean vector [8].

In F-WNN algorithm the features (color & texture) are combined. The nearest neighbors are determined for the combined features and harmonic mean vector of the combined features are calculated and finally predicted the class of an input image I. The HM-WNN shows the performance of proposed method where the nearest neighbors for color feature and texture feature vector are determined separately. We check the value of k between 1 and 50 to identify the correct value of k in order to determine no of neighbors. The value of k at which the highest accuracy is obtained on dataset is selected within the range.

TABLE I. PRECISION AND RECALL OF DIFFERENT CLASSIFICATION METHODS

Method	Precision (%)	Recall (%)
SVM	86.16	86.16
kNN	84.33	84.50
LMKNN	86.16	86.50
PNN	82.50	82.33
LMPNN	84.50	83.50
F-WNN	87.16	87.16
HM-WNN	87.73	88.66

TABLE II. F1-SCORE AND ACCURACY OF DIFFERENT CLASSIFICATION METHODS

Method	F1-Score (%)	Accuracy (%)
SVM	86	85.92
kNN	84	83.87
LMKNN	85.66	85.83
PNN	81.83	82.13
LMPNN	82.66	82.78
F-WNN	86.50	86.50
HM-WNN	88.00	88.01

From Tables I and II, it is observed that the proposed classifier based on harmonic mean vector determined from nearest neighbors of each features separately gives good results for classification of tagged images as compared to the SVM, traditional kNN, LMKNN, PNN, LMPNN classification algorithm. Also, classification error rate of the HM-WNN method is less than other classification algorithms as shown in Fig. 1.

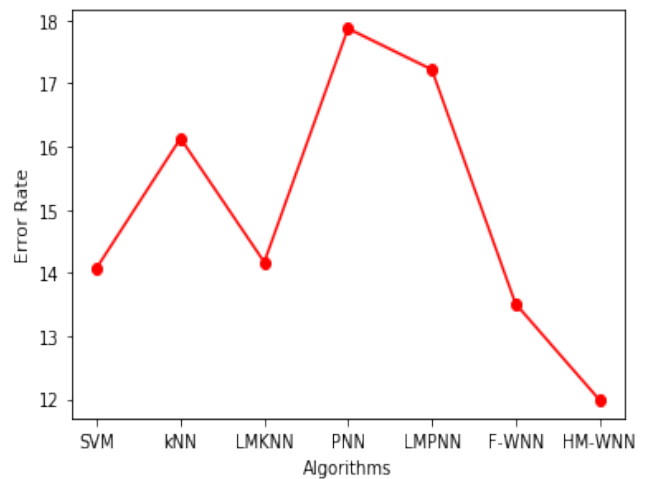


Fig. 1. Error Rate of different Classification Methods.

### VIII. CONCLUSION

In the paper, HM-WNN classifier is proposed for classification of the images. The main objective of the HM-WNN is to improve classification accuracy by creating a separate classifier for each feature and combining the

harmonic mean obtained by classifier designed for each feature. The proposed method also used the information of neighbors per category over the entire training samples. The experimental result shows that the HM-WNN provides good performance as compared to the existing kNN methods. In future we need to develop a method to assign tags to the new images using tags assigned to the visually similar images.

#### REFERENCES

- [1] Cover, Thomas, and Peter Hart. "Nearest neighbor pattern classification." *IEEE transactions on information theory* 13, no. 1 (1967): 21-27.
- [2] Zhang, Shichao, Xuelong Li, Ming Zong, Xiaofeng Zhu, and Ruili Wang. "Efficient knn classification with different numbers of nearest neighbors." *IEEE transactions on neural networks and learning systems* 29, no. 5 (2017): 1774-1785.
- [3] Zhang, Shichao, Xuelong Li, Ming Zong, Xiaofeng Zhu, and Debo Cheng. "Learning k for knn classification." *ACM Transactions on Intelligent Systems and Technology (TIST)* 8, no. 3 (2017): 1-19.
- [4] García-Pedrajas, Nicolás, Juan A. Romero del Castillo, and Gonzalo Cerruela-García. "A proposal for local k values for k -nearest neighbor rule." *IEEE transactions on neural networks and learning systems* 28, no. 2 (2015): 470-475.
- [5] Li, Ping, Jianping Gou, and Hebiao Yang. "The distance-weighted k-nearest centroid neighbor classification." *J. Intell. Inf. Hiding Multimedia Sig. Process* 8, no. 3 (2017): 611-622.
- [6] Gou, Jianping, Lan Du, Yuhong Zhang, and Taisong Xiong. "A new distance-weighted k-nearest neighbor classifier." *J. Inf. Comput. Sci* 9, no. 6 (2012): 1429-1436.
- [7] Dudani, Sahibsingh A. "The distance-weighted k-nearest-neighbor rule." *IEEE Transactions on Systems, Man, and Cybernetics* 4 (1976): 325-327.
- [8] Liu, Shenglan, Ping Zhu, and Sujuan Qin. "An improved weighted KNN algorithm for imbalanced data classification." In *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*, pp. 1814-1819. IEEE, 2018.
- [9] Fan, Guo-Feng, Yan-Hui Guo, Jia-Mei Zheng, and Wei-Chiang Hong. "Application of the weighted k-nearest neighbor algorithm for short-term load forecasting." *Energies* 12, no. 5 (2019): 916.
- [10] Mitani, Yoshihiro, and Yoshihiko Hamamoto. "A local mean-based nonparametric classifier." *Pattern Recognition Letters* 27, no. 10 (2006): 1151-1159.
- [11] Zeng, Yong, Yupu Yang, and Liang Zhao. "Pseudo nearest neighbor rule for pattern classification." *Expert Systems with Applications* 36, no. 2 (2009): 3587-3595.
- [12] Gou, Jianping, Yongzhao Zhan, Yunbo Rao, Xiangjun Shen, Xiaoming Wang, and Wu He. "Improved pseudo nearest neighbor classification." *Knowledge-Based Systems* 70 (2014): 361-375.
- [13] Pan, Zhibin, Yidi Wang, and Weiping Ku. "A new k-harmonic nearest neighbor classifier based on the multi-local means." *Expert Systems with Applications* 67 (2017): 115-125.
- [14] Gou, Jianping, Wenmo Qiu, Zhang Yi, Yong Xu, Qirong Mao, and Yongzhao Zhan. "A local mean representation-based K-nearest neighbor classifier." *ACM Transactions on Intelligent Systems and Technology (TIST)* 10, no. 3 (2019): 1-25.
- [15] Gou, Jianping, Hongxing Ma, Weihua Ou, Shaoning Zeng, Yunbo Rao, and Hebiao Yang. "A generalized mean distance-based k-nearest neighbor classifier." *Expert Systems with Applications* 115 (2019): 356-372.
- [16] Ji, Qian, Liyan Zhang, Xiangbo Shu, and Jinhui Tang. "Image annotation refinement via 2P-KNN based group sparse reconstruction." *Multimedia Tools and Applications* 78, no. 10 (2019): 13213-13225.
- [17] Hao, Zhangang, Hongwei Ge, and Long Wang. "Visual attention mechanism and support vector machine based automatic image annotation." *PLoS one* 13, no. 11 (2018): e0206971.
- [18] Cui, Chaoran, Jialie Shen, Jun Ma, and Tao Lian. "Social tag relevance learning via ranking-oriented neighbor voting." *Multimedia Tools and Applications* 76, no. 6 (2017): 8831-8857.
- [19] Qian, Xueming, Xian-Sheng Hua, Yuan Yan Tang, and Tao Mei. "Social image tagging with diverse semantics." *IEEE transactions on cybernetics* 44, no. 12 (2014): 2493-2508.
- [20] Wu, Wei, Jianyun Nie, and Guanglai Gao. "An improved SVM-based multiple features fusion method for image annotation." *Journal of Information & Computational Science* 11, no. 14 (2014): 4987-4997.
- [21] Qian, Xueming, Xiaoxiao Liu, Chao Zheng, Youtian Du, and Xingsong Hou. "Tagging photos using users' vocabularies." *Neurocomputing* 111 (2013): 144-153.
- [22] Li, Xirong, Cees GM Snoek, and Marcel Worring. "Learning social tag relevance by neighbor voting." *IEEE Transactions on Multimedia* 11, no. 7 (2009): 1310-1322.
- [23] Li, Xirong, and Cees GM Snoek. "Classifying tag relevance with relevant positive and negative examples." In *Proceedings of the 21<sup>st</sup> ACM international conference on Multimedia*, pp. 485-488. 2013.
- [24] Shen, Yi, and Jianping Fan. "Leveraging loosely-tagged images and inter-object correlations for tag recommendation." In *Proceedings of the 18<sup>th</sup> ACM international conference on Multimedia*, pp. 5-14. 2010.
- [25] Chen, Hong-Ming, Ming-Hsiu Chang, Ping-Chieh Chang, Ming-Chun Tien, Winston H. Hsu, and Ja-Ling Wu. "Sheepdog: group and tag recommendation for flickr photos by automatic search-based learning." In *Proceedings of the 16<sup>th</sup> ACM international conference on Multimedia*, pp. 737-740. 2008.
- [26] Fu, Hao, Qian Zhang, and Guoping Qiu. "Random forest for image annotation." In *European Conference on Computer Vision*, pp. 86-99. Springer, Berlin, Heidelberg, 2012.
- [27] Anupama D. Dondekar and Balwant A. Sonkamble, "Analysis of Flickr Images using Feature Extraction Techniques," *4<sup>th</sup> IEEE International Conference on Computer and Communication Systems (ICCCS 2019)*, Singapore, pp. 278-282.