EEE3032 – Computer Vision and Pattern Recognition

Coursework Assignment

Visual Search of an Image Collection

URN: 6654308

1. Abstract

Visual search is a search system which uses multiple set of algorithms to find semantically relevant images as per the input image. In this assignment, I have explored similar problem using MSRCV2 dataset of 591 images. This assignment is based on image processing techniques of extracting features from images and excludes state of the art deep learning techniques. Problem uses an input image and finds similar looking images based on global color histogram, spatial color, texture and combination of spatial color and texture. Descriptors calculated by mentioned algorithms are compared with search query using distance metrics such as L1 norm, L2 (Euclidean), cosine and Mahalanobis distance. Spatial color and texture capture more distinguishable features and outperforms other feature descriptors by achieving a precision of 0.25. Principal Component Analysis (PCA) is used for dimensionality reduction of feature descriptors to achieve compactness of features by preserving feature importance. Experiments show that PCA with mahalanobis distance on spatial color and texture increases the precision further but it doesn't consistently improve performance of other feature descriptors. Also, object classification using Support Vector Machines (SVM) is performed on similar dataset. 70-30% split train and test dataset shows maximum F1 score by avoiding overfitting. SVM achieves F1 score of 0.49 on spatial color and texture feature descriptors.

Contents

2. Introduction

Visual search or content-based image retrieval is one of the widely used applications of Computer vision. Text retrieval is easily achievable but visual search field is still in research phase. State of the art techniques [1] in visual research are based on deep learning and similarity metrics where a generalized feature extractors based on zero/few shot learning learns features for each image and compares it with similarity metrics such as cosine similarity, Euclidean distance, etc.

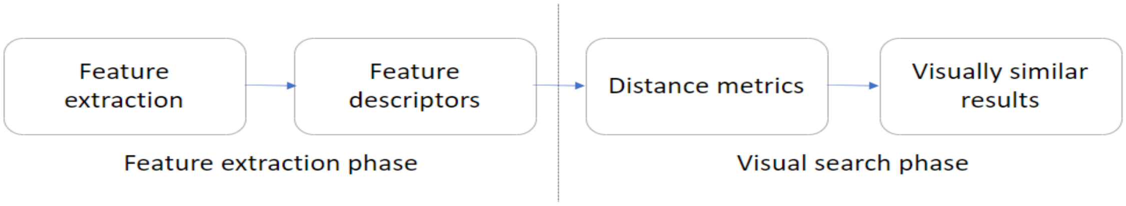


Figure 1: Process flow of visual search techniques

In this assignment, I focus on computer vision techniques to perform visual search and exclude recent developments in deep learning based visual comparison. Typical visual search techniques consist of feature extraction stage and visual comparison stage.

2.1. Feature extraction

Given a dataset, feature extraction techniques calculate features from each image. For each image, a feature descriptor is produced which is a vector of N dimensions. The goal of this feature descriptor is to capture semantically distinguishable features with as low dimension as possible. Feature descriptors of all images are stored in different files or data structure. Feature extraction techniques are based on global and spatial color distribution, texture analysis, etc.

2.2. Visual comparison

Visual comparison is a method of comparing a query image with images in dataset to retrieve similar looking images. In this assignment, one random query image is selected from dataset with its calculated feature descriptor. It is compared with all feature descriptors of dataset images using distance metrics. The distances are sorted in ascending order and top K images are shown as visually similar images. L1 norm, L2 norm, Mahalanobis and cosine similarity metrics are used to compare the performance.

Below sections shows analysis of different feature descriptors, distance metrics, analysis with different grid configurations. Section 3 details technical description of experiments performed. Section 4 is based on performance analysis of each algorithm. Precision-recall metrics, F1 score and confusion matrix are used for performance analysis of the techniques. Section 6 details classification of images using SVM.

3. Experiments

This section focuses on technical description, experimental usage, limitations, and advantages of techniques in visual search.

3.1. Average RGB color descriptor

One of the basic descriptors is average RGB color descriptor. It splits image into red, green, and blue color channels and calculates mean of each pixel intensities of respective channel. Each image is represented with a feature descriptor of 1x3 dimensional vector.

Visual search is based on semantic contents in image and these exceptionally low dimensional vector do not capture any distinguishable features. It is unable to capture shape or color distribution in image and shows irrelevant search suggestions. Hence, average RGB color descriptors are not ideal to use in practical visual search techniques.

3.2. Global color histogram

Global color histogram is based on color representations of an image. Each image consists of red, green, and blue channels and has different resolution. If these intensities are presented as descriptors, dimensions will change with respect to image resolution. Hence, descriptors should be restricted to specified bins and invariant to resolution of image.

In this method, RGB color space of image is quantized into indexed bins using quantization factor (*q*). Each pixel has *R = [0, 255], G = [0, 255]* and *B = [0, 255]* component. Pixel intensity is quantized into a specified bin using following formula on each color component.

*Color bin value = floor ((color value * q)/256)*

For each color component, a respective bin value is calculated. Floor function is used to round the division to nearest integer bin. Each pixel is now represented with 3 color bin values for 3 color components. To further reduce the dimensions, each pixel should be given one bin so that global histogram is formed. This is implemented by linear quadratic equation of 2 color components with coefficient i.e., quantization factor (*q*).

$$Bin = q^2 * (red\ color\ bin) + q * (green\ color\ bin) + 1 * (blue\ color\ bin)$$

Hence, each pixel range is now transformed into bin in the range of $[0, q^3 - 1]$. Now each bin will have some number of pixels lying inside its distribution. A histogram is calculated based on number of pixels in each bin using MATLAB’s histogram function.

As average RGB color descriptor only captures global color, global color histogram can capture distribution of every color in an image. This analyzes color distribution in the whole image and captures color ranges of all the objects in an image. However, global color histogram does not capture any spatial or texture information. For example, if two images have different background and two same objects with distinct colors, descriptor will determine both images to be different.

3.3. Spatial color

Spatial color is another type of feature descriptor which focuses on spatial components of an image. Above approaches did not preserve any positional features and were dependent on global distribution. However, considering local features approximates the objects more accurately. Multiple local descriptors are combined to form a global descriptor for an image. Image is divided into grids with rows and columns to be configurational parameters. For each grid in image, red, green, and blue components are averaged to form a local descriptor. To preserve the compactness of descriptor, all local descriptors are appended in global descriptor. Size of global descriptor is given as *3WH*, where *W* is the horizontal number of grids and *H* is the vertical grids.

Color histogram can be used in each cell to capture spatial color distribution, but it increases the size of descriptor-based number of grids. If the number of grids is high, the size of the descriptor becomes too large and increases time and space complexity.

Spatial color descriptor captures the region color intensity rather than complete count of pixel intensities in image. If a person is in front of camera, it would capture the distribution of face color and background color with respect to its position and will compare similar images in search space. However, global color histogram will just capture amount of color present, which also can be present in altogether different image.

One disadvantage of spatial color is its dependency on positional color intensity. Two different objects with same color distribution might be similar based on spatial color descriptors.

3.4. Spatial texture

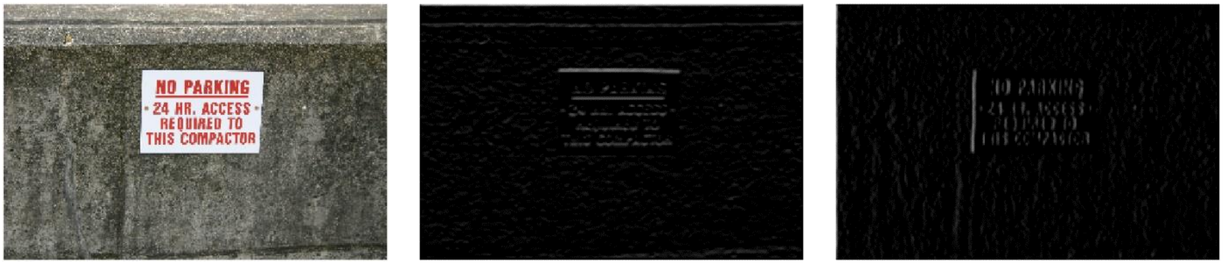
As above methods fail miserably to two different objects with same color intensity, it is required to capture the local texture features as well to increase semantic level of features. Instead of calculating color distribution in each grid, this descriptor analyzes texture based on edge magnitude and direction. Edge detection is performed in the X and Y direction with the help of Sobel operator. It calculates first order derivative with image using S_x and S_y filter and highlights areas of changing intensities to be edges.

Sx=

1	2	1
0	0	0
-1	-2	-1

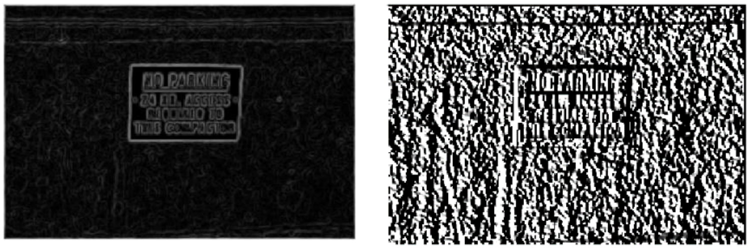
Sy=

1	0	-1
2	0	-2
1	0	-1



(a) (b) (c)

Figure 2: Image (a) shows original image. Image (b) is the horizontal edge detections with X direction Sobel filter and image (c) is vertical edge detections with Y direction Sobel filter



(a) (b)

Figure 3: Image (a) show the magnitude of edges detected in Figure 2 (a) and Image (b) shows theta angle of all the edges to define texture details.

Edge magnitudes and slope of edges are calculated using the formula below.

$$\text{Edge magnitude} = \text{sqrt}\ (Dx^2 + Dy^2)$$

$$\text{Theta} = \tan^{-1}(Dy / Dx)$$

Where, Dx and Dy are horizonal and vertical edge magnitudes, respectively. Slope of edges is calculated by angle between vertical and horizontal edge components.

Given a specific threshold, strong edge magnitudes are selected. Theta is normalized by dividing it with *2*pi* and further multiplied with number of bins. Histogram is computed based on count of theta angles with respect to bins.

3.5. Spatial color and texture

Spatial color and texture capture the best from both the world i.e spatial color and spatial texture. It captures color distribution and texture for each grid and learns positional as well as color distribution features.

For each grid, spatial color and spatial texture features are calculated and appended to form a local descriptor. All local descriptors combined represent a global descriptor for an image.

3.6. Dimensionality reduction by PCA

Most feature descriptors face ‘Curse of dimensionality’, a phenomenon of dealing with large dimensions of features which increases time and space complexity. Hence, feature descriptors should preserve the prominent features of images and should be as low dimensional as possible. Increase in dimensions linearly increases the processing exponentially. Also, most dimensions in descriptors have no or very little variation and no impact in the feature representation.

Principal Component Analysis (PCA) is a technique of reducing the dimensionality of descriptors based on low variation in a dataset. It not only increases the interpretability but also minimizes the information loss. It identifies directions and magnitude of variation in dataset which is also called as the covariance. The covariance is defined by eigen model with eigen values and eigen vectors which captures magnitude and direction in each dimension of descriptors, respectively. It removes the low variation components with the help of low magnitude eigen values. The dataset is fit into lower dimension which reduces the complexity and curse of dimensionality.

3.7. Similarity measures

After computing the feature descriptors of all images in dataset, a query image is selected at random and compared with all images to find *topK* similar matching images. L1 norm, L2 norm, Mahalanobis distance and cosine metrics is used for the similarity metrics.

- L1 norm

L1 norm or Manhattan distance is calculated by magnitudes of distances along vector components. L1 norm of a point in coordinate system $x = (i, j)$ with respect to origin is represented as follows.

$$D(x) = |i| + |j|$$

- L2 norm

L2 norm or Euclidean distance is the shortest straight-line distance between two vectors. Euclidean distance of a vector $x = (i, j)$ from $y = (p, q)$ is represented as follows. Sqrt is a square root function.

$$D(x) = sqrt\ ((p-i)^2 + (q-j)^2)$$

- Mahalanobis distance

Mahalanobis distance is a multi variate distance metrics which calculates the distance from point to a distribution in effective way than L1 and L2. The components of eigen model are used to compute mahalanobis distance. It is calculated by subtracting a point (x) from mean (u) which is further multiplied by inverse of eigen values (U) and eigen vectors (V) respectively.

$$D(x)^2 = |V^{-1}\ U^T\ (x-u)|$$

- Cosine similarity

Cosine of the angle is found between two N dimensional vectors (*x and y*) in cosine similarity [2]. Cosine distance is calculated by dot product of two vectors which is further divided by product of magnitudes of vectors.

$$Similarity(x,y) = (x.y)\ /\ (|x|*|y|)$$

4. Performance comparison

4.1. Dataset

In this assignment, I have used MSRC v2 dataset by Microsoft. It consists of 591 images of 20 categories. The dataset includes classes such as cows, bicycles, flowers, etc. Some of the classes have overlap between the features such as cows and sheep in green background. For evaluation purposes, all classes are independent even though some classes visually look similar to distinguish the robustness of model.

4.2. Evaluation strategy

To evaluate model performance, randomly 50 query images are selected. Each query image is compared with all images in the dataset. The distance for each image is sorted in ascending order so that the most relevant image is ranked first and so on. Precision and recall values for each comparison in iteration are stored to calculate precision recall curve. Mean average precision (mAP) is used to measure the performance of a model. Standard deviation shows variation in precisions as each class has different precision.

4.3. Precision and recall

Precision and recall metrics are common metrics used for the evaluation of visual search systems. Precision defines how many relevant images are selected for class out of all incoming instances. However, recall defines the correctness of instances of class out of class predictions. Precision and recall are defined as follows.

Precision = (Correct classification / total images)

Recall = Correct classification / total image of the same class

In visual search, Precision is calculated as number of relevant images up to top N candidate images. Higher performance indicates high precision value. Ideal system should select all the relevant documents in the complete dataset to achieve precision value of 1.

Recall is defined as the number of relevant images returned out of N relevant images. Recall starts at zero and increases gradually over the complete iteration in dataset. Higher recall indicates that model recalls relevant images faster and measures how long it takes to fetch all relevant images.

If a system has high recall and low precision, the system retrieves all relevant documents with the possibility of selecting false positives. A system with high precision and low recall shows that system is confident but results in some false negative results.

4.4. F1 score

F1 [3] score is the harmonic mean of precision and recall. It provides better results by considering false negative categories compared to Accuracy. F1 score is crucial when impact of false positives and false negatives is to be considered. Accuracy measures the impact of true positive and true negatives.

*F1 score = (2 * recall * precision) / (precision + recall)*

As imbalanced data distribution is used in training and testing SVM, F1 score is used to compare the performance of object classification.

5. Experimental results

5.1. Average RGB color descriptor

Below table shows the mean average precision achieved by average RGB color descriptor. Results of L1, Euclidean and cosine similarity distance metrics show that Euclidean distance outperforms other metrics. As RGB color descriptor does not capture any texture or spatial color space; it is not the ideal descriptor for feature extraction.

Distance metrics	mAP	Standard deviation
L1	0.1347	0.0769
Euclidean distance	0.1763	0.0644
Cosine similarity	0.1654	0.0483

Table 1: mAP performance of Average RGB color descriptor on 50 random query images.

5.2. Global color histogram

Feature descriptors of all images are calculated with varying size of bins. Figure 4 shows the analysis of distinct size of bins and their mAP impact over different distance metrics. Method follows the same evaluation strategy mentioned above. L1 norm achieves higher precision over multiple tests runs and performs better than L2 and cosine distance metrics. Precision of L1 norm increases as we keep increasing the number of bins. Precision recall curve is displayed in Appendix 9.2.

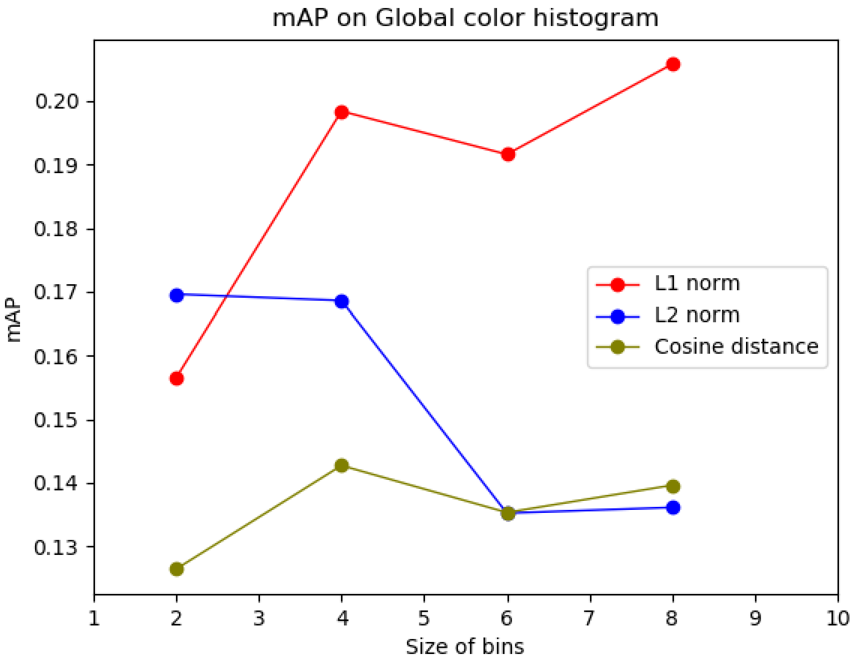


Figure 4: MAP analysis on global color histogram. Figure shows the analysis based on different distance metrics and size of bins. Image is generated by Matplotlib

5.3. Spatial color

Spatial color information divides image into grids and calculates average color representation for each grid. The image is divided into number of rows and columns which define grids. The mesh graph below shows the effect of number rows and columns on mAP values. L1 norm achieves 0.28 mAP on 16x16 grids whereas L2 norm achieves 0.26 on 16x15 grids. The details of spatial color experiments and precision recall curve can be found in Appendix 9.1.

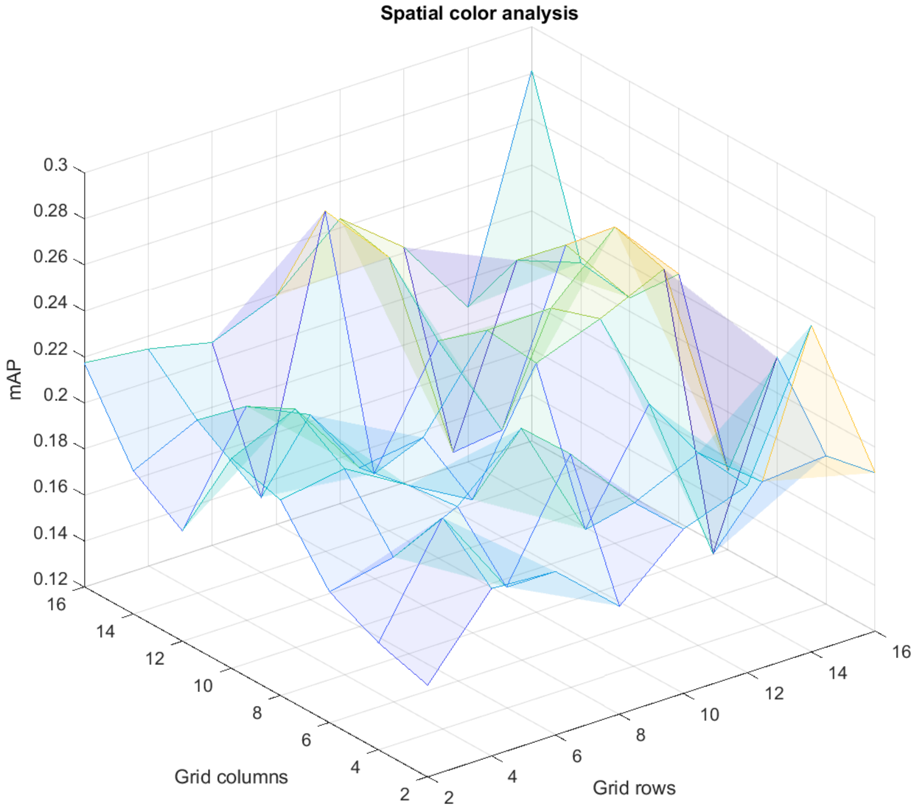


Figure 5: Mesh graph of mAP values by spatial color information using L1 norm

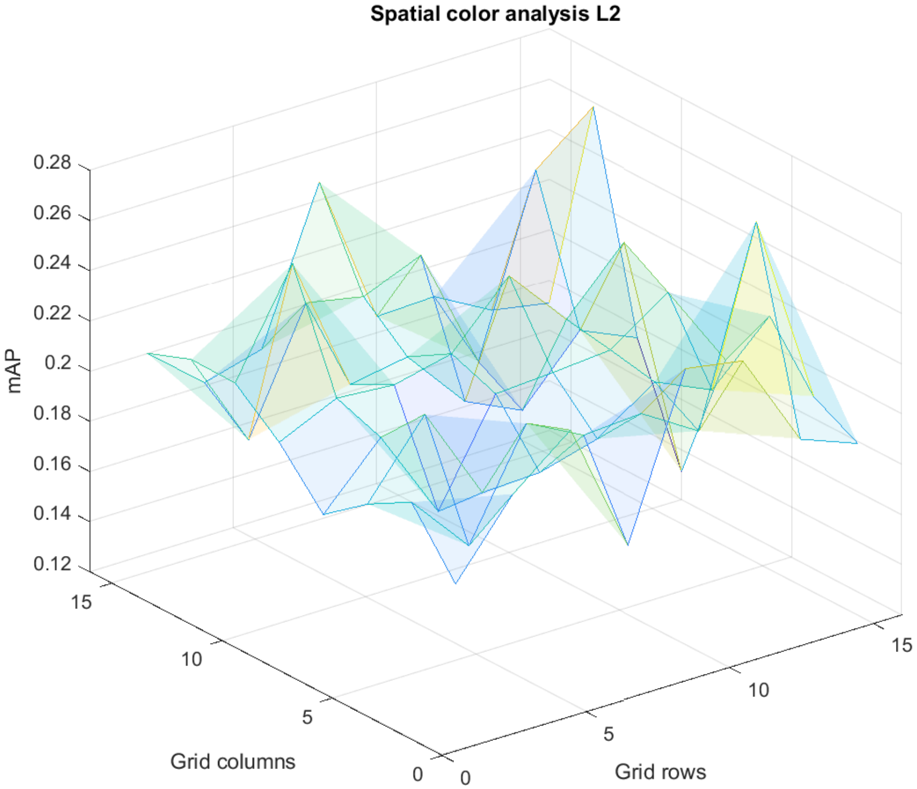


Figure 6: Mesh graph of mAP values by spatial color information by L2 norm

5.4. Spatial texture

Spatial texture analyzes positional texture features based on Sobel edge detection. It identifies the strong edges based on number of grids. Histogram is generated by defining 8 number of bins, edge magnitude and edge direction. By observing the results on different grid sizes, it is found that 6 and 8 horizontal and vertical grids achieve better performance than other configurations. Analysis of precision recall curve is shown in Appendix 9.3.

Distance measure	mAP	Std
L1 norm	0.2106	0.1206
L2 norm	0.1956	0.0956
Cosine distance	0.1793	0.0142

Table 2: mAP and standard deviation of spatial texture descriptor on different distance metrics

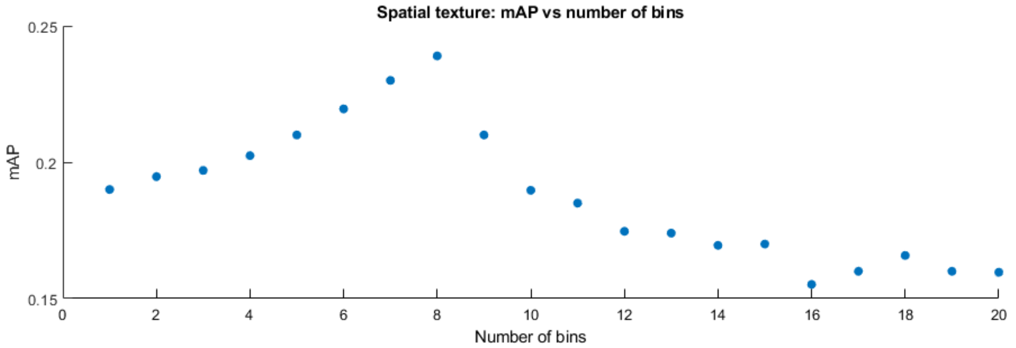


Figure 7: Plot shows the analysis of mAP vs different number of bins. Experiment is performed on 6x8 grids and shows better performance on 8 number of bins.

5.5. Spatial color and texture

Spatial color and texture combine both the features from color representations and texture analysis over multiple grids. Below table shows that L1 norm with 6 horizontal and vertical grids achieves 0.25 mAP and outperforms L2 norm and cosine distance metrics.

Distance measure	mAP	Std
L1 norm	0.2450	0.2430
L2 norm	0.2316	0.1571
Cosine distance	0.1644	0.1205

Table 3: mAP and standard deviation of spatial texture and color descriptor on different distance metrics

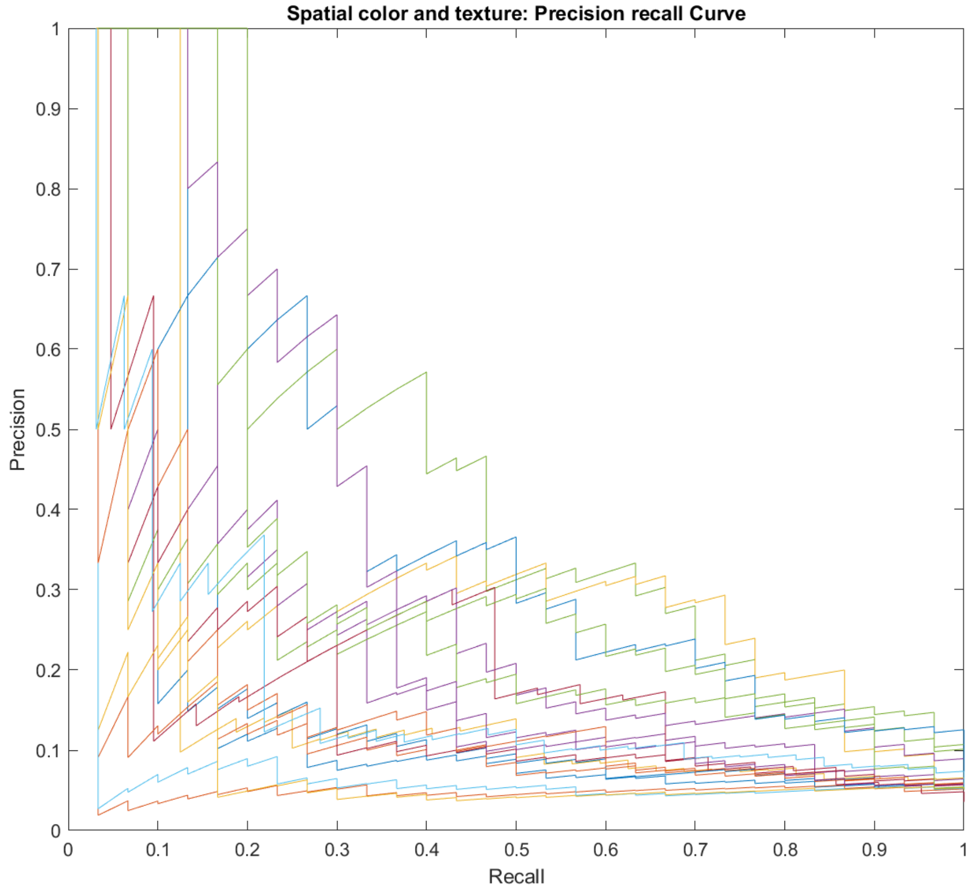


Figure 8: Precision recall curve of each image in 20 categories. Query image is compared with 591 test images. As recall reaches one, precision is in 0.05 to 0.2 range of all classes.

5.6. Principal Component Analysis

PCA is used to reduce the dimensionality of feature descriptors by using eigen model. Low variations in feature dimensions are reduced to derive compact feature descriptors. It is used with Mahalanobis distance to verify if performance gain is observed. The best performing configurations with L1 distance are used to compare PCA performance. Even though performance gain is not consistent, PCA improves mAP on Spatial color and spatial color with texture features.

Model	MAP without PCA	MAP with PCA
Global color histogram	0.1927	0.1456
Spatial color	0.1923	0.2417
Spatial texture	0.2112	0.1531
Spatial color and texture	0.2450	0.2508

Table 4: mAP comparison of different descriptors with and without PCA

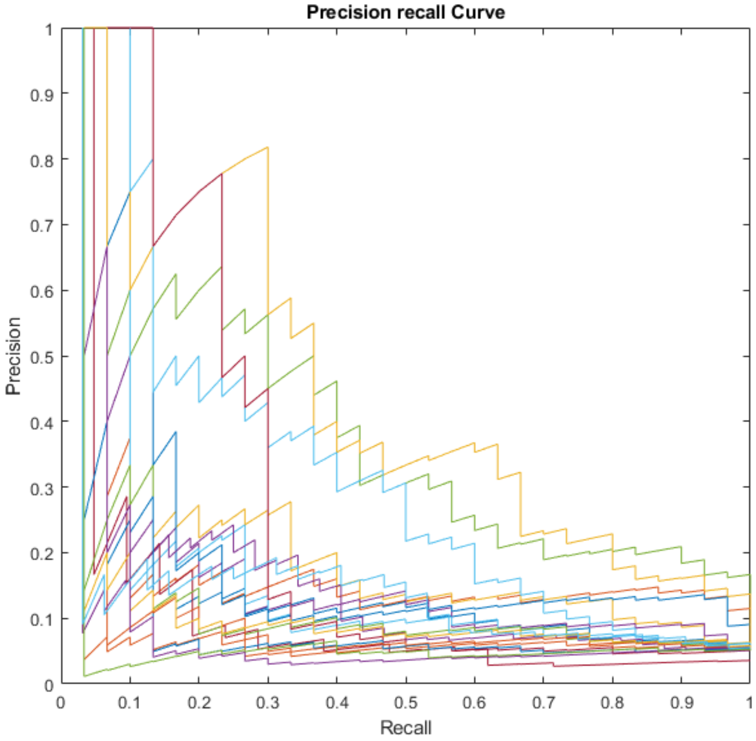


Figure 9: Precision recall curve of each image on 20 categories using Spatial texture descriptor. Query image is compared with 591 test images. As recall reaches one, precision is in 0.05 to 0.21 range of all classes.

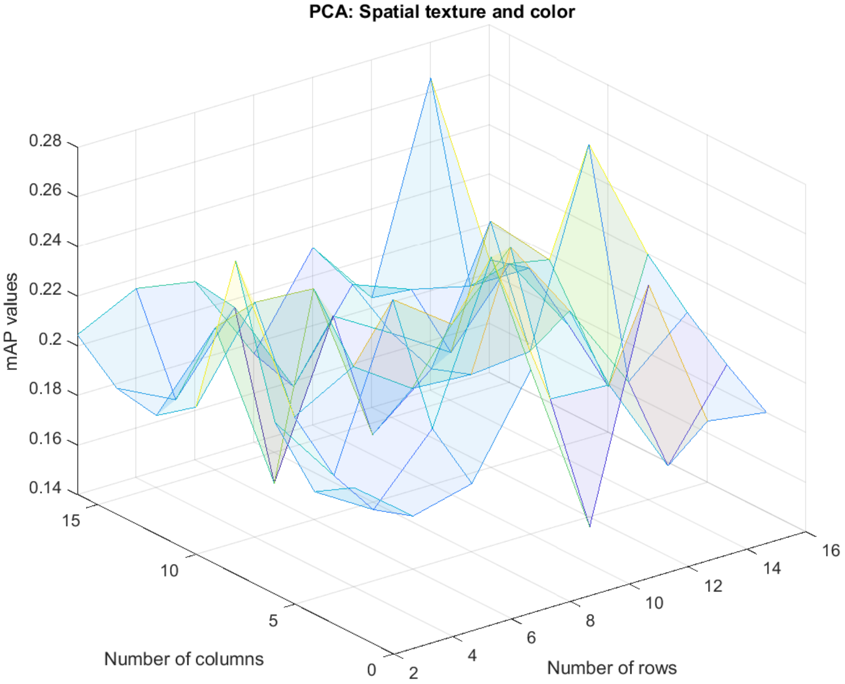


Figure 10: Mesh graph of mAP values over multiple row and column configuration. Experiment is performed on PCA with mahalanobis distance. Spatial color and texture is used as feature descriptor. Maximum mAP value is 0.27 on 12x15 grid size.

5.7. Comparative analysis

The comparative analysis of different feature descriptors is shown the following graph. Spatial color and texture features outperform other descriptors. Significant improvement is observed in spatial features when compared with global color histograms. Spatial information is a crucial factor in visual comparison. Global color histogram does not add intricate features and performs better when the same intensity color images of classes are shown. On the other hand, spatial color and texture adds more semantic details to the features by analyzing colors and textures in the subsection of images.

Color features fail to distinguish two distinct categories when color distributions are same. If a bicycle and sheep is on green grass, descriptors are less distinguishable and fails to separate both classes. Shape features add more variations in the descriptor as both bicycle and sheep has different shape.

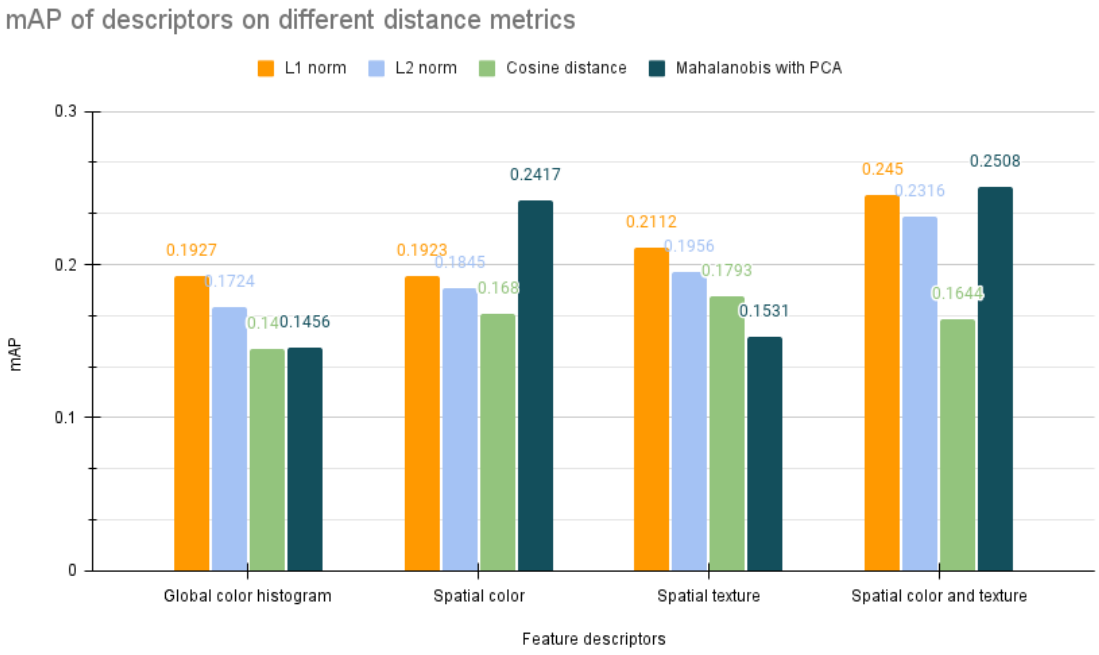


Figure 11: Comparative analysis of different descriptors. It shows spatial color and texture with PCA achieves better mAP results.

As shown in above image, L1 norm is an effective measure when comparing two different descriptors. The experiment of reducing dimensionality using Eigen models and Mahalanobis distance increases the precision in most of the cases. Cosine distance is proved to be not effective distance metrics in this scenario. Spatial color and texture with PCA achieves the best mAP of 0.2508.

6. Object classification using SVM

Support vector machines (SVM) is a widely used algorithm in supervised classification and regression analysis. SVM is a robust algorithm which achieves significant accuracy gains with less computational power. The main objective of SVM is to find a hyperplane between two classes which creates a decision boundary in feature latent space. In case of multiple classes, SVM creates multiple hyperplanes to distinguish classes from each other. Multiclass SVM is achieved by decomposing multiple SVMs into binary classification problem where the relationship of one-versus-all or one-versus-one is found between class labels.

In this assignment, object classification using SVM is performed in MATLAB. Feature descriptors of all the images in a dataset are saved in respective folders for each feature descriptors mentioned above. The features with class labels are loaded to form a complete dataset. It is further divided into training and testing dataset with varying number of data proportion. The model achieves the best results at 70-30% train and test split, respectively. Analysis of data proportion provides an estimate of sufficient data for the SVM model to avoid overfitting or underfitting.

Further, training data is used to train the model and test data is used for performance comparisons. Confusion matrix, precision-recall curve and F1 score is used to compare accuracy on various shape descriptors.

6.1. Analysis of data proportion

To avoid overfitting and underfitting of SVM model, experiments are performed with different train and test splits. Below table shows the accuracy and F1 score on test dataset. This analysis is based on selecting global color descriptor. Table shows train-test split of 70-30% achieves better accuracy and F1 score. Hence, 70-30% split is used for further experiments.

Train dataset (%)	Test dataset (%)	Accuracy	F1 score
50	50	0.3567	0.3194
60	40	0.3212	0.3433
70	30	0.4188	0.3812
80	20	0.3743	0.3786
90	10	0.3950	0.3745

Table 5: Train and test split strategy for object classification using SVM

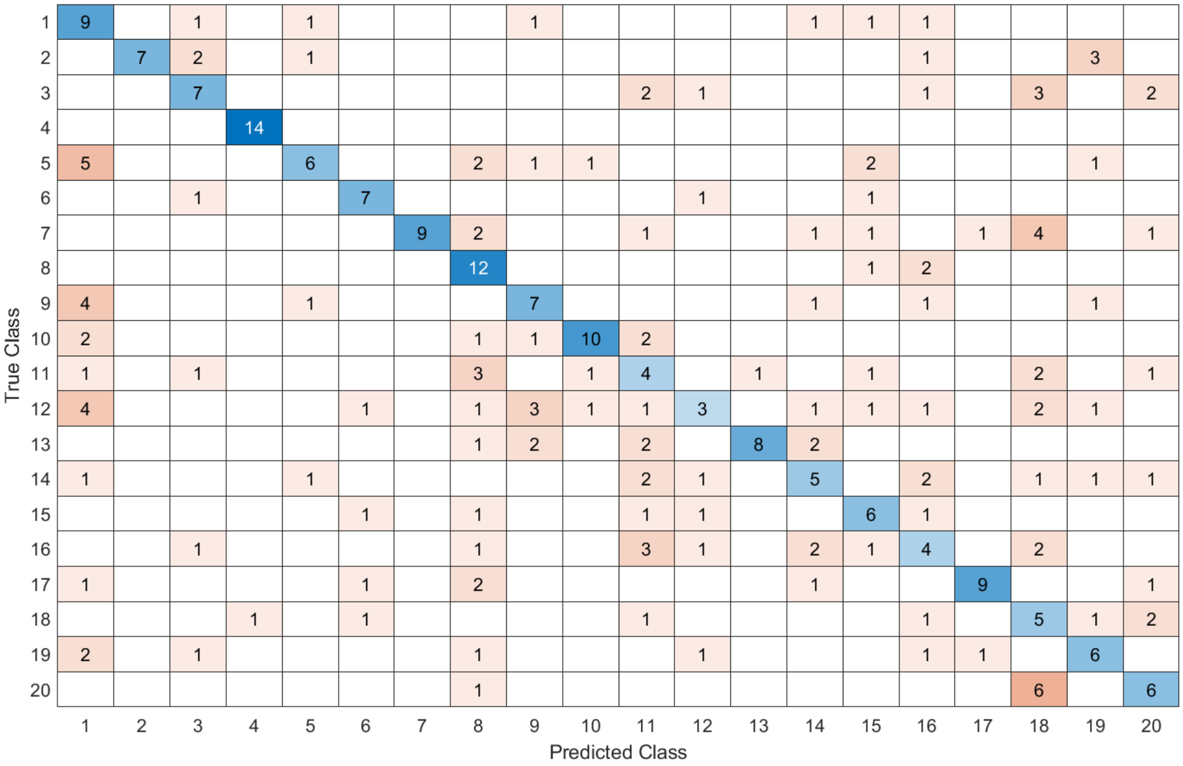
6.2. Classification performance

Based on the above dataset splits, different feature descriptors are used for feature extraction. Below table shows performance of object classification via SVM with various descriptors.

Feature descriptor	Accuracy	F1 score
Global color histogram	0.4188	0.3812
Spatial Color	0.4211	0.4032
Spatial texture	0.3960	0.3876
Spatial color and texture	0.4881	0.4982

Table 6: Accuracy and F1 scores of different descriptors with SVM. Experiments are performed on 60-40% split strategy

Figure 12 shows the confusion matrix of SVM classification. It is calculated on spatial color and texture feature descriptors as it achieves better F1 score of 0.4982. Following image shows that model learns robust and distinguishable features for class 2, 4, 7, 10, 13 and 17.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	9		1		1				1					1	1	1				
2		7	2		1											1			3	
3			7								2	1				1		3		2
4				14																
5	5				6			2	1	1					2				1	
6			1			7						1			1					
7							9	2			1			1	1		1	4		1
8								12							1	2				
9	4				1				7					1		1			1	
10	2							1	1	10	2									
11	1		1					3		1	4		1		1			2		1
12	4					1		1	3	1	1	3		1	1	1		2	1	
13								1	2		2		8	2						
14	1				1						2	1		5		2		1	1	1
15						1		1			1	1			6	1				
16			1					1			3	1		2	1	4		2		
17	1					1		2						1			9			1
18				1		1					1					1		5	1	2
19	2		1					1				1				1	1		6	
20								1										6		6

Figure 12: Image shows the confusion matrix of spatial color and texture feature descriptors with SVM. It is calculated on 60-40% train-test split.

7. Conclusion

Finding distinguishable semantic features in an image is a crucial factor in Visual search. Selecting feature descriptor is dependent on the usage of different applications and it's important to define the use case before selecting appropriate feature descriptor. Parameters of image processing feature descriptors also are dependent on different configurations and datasets. The best configuration can be achieved by different experimental setting. Spatial texture features perform better than global color histograms as it learns more representative features. Experiments showed that spatial color with texture outperforms other feature descriptors using L1 norm. Also, dimensionality reduction with PCA increases the mAP further but not by noticeable margin. Feature descriptors of each experiment are used to perform object classification using SVM. As spatial color and texture features are more distinguishable, it achieves F1 score of 0.49 and outperforms other feature descriptors. In summary, feature descriptors should be more distinguishable and capture intricate features or styles in images to provide better generalization.

8. References

[1] Wiggers, K.L., Britto, A.S., Heutte, L., Koerich, A.L. and Oliveira, L.S., 2019, July. Image retrieval and pattern spotting using siamese neural network. In *2019 International Joint Conference on Neural Networks (IJCNN)* (pp. 1-8). IEEE.

[2] Lahitani, A.R., Permanasari, A.E. and Setiawan, N.A., 2016, April. Cosine similarity to determine similarity measure: Study case in online essay assessment. In *2016 4th International Conference on Cyber and IT Service Management* (pp. 1-6). IEEE.

[3] Bénédict, G., Koops, V., Odijk, D. and de Rijke, M., 2021. sigmoidF1: A Smooth F1 Score Surrogate Loss for Multilabel Classification. *arXiv preprint arXiv:2108.10566*.

[4] Noble, W.S., 2006. What is a support vector machine?. *Nature biotechnology*, *24*(12), pp.1565-1567.

9. Appendix

9.1. Spatial color analysis

Distance metrics	mAP	Standard deviation
L1	0.1923	0.0568
Euclidean distance	0.1845	0.0622
Cosine similarity	0.168	0.0532

Table 7: mAP and standard deviation of spatial color analysis. L1 norm outperforms other distance metrics in visual search

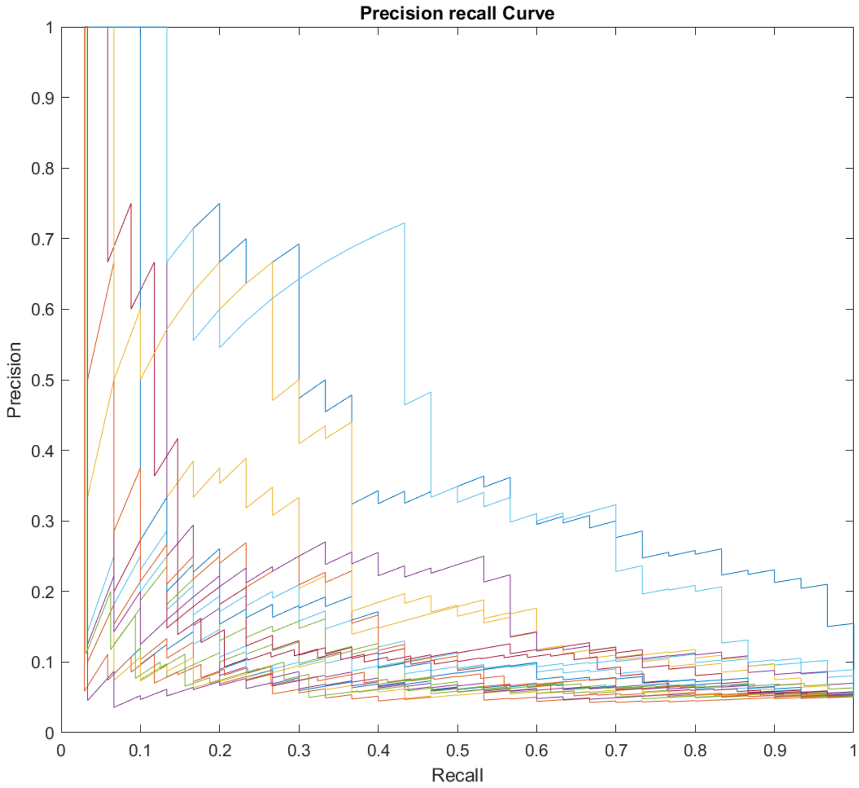


Figure 13: Precision recall curve of each image in 20 categories. Query image is compared with 591 test images. As recall reaches one, precision is in 0.08 to 0.21 range of all classes.

9.2. Global color histogram

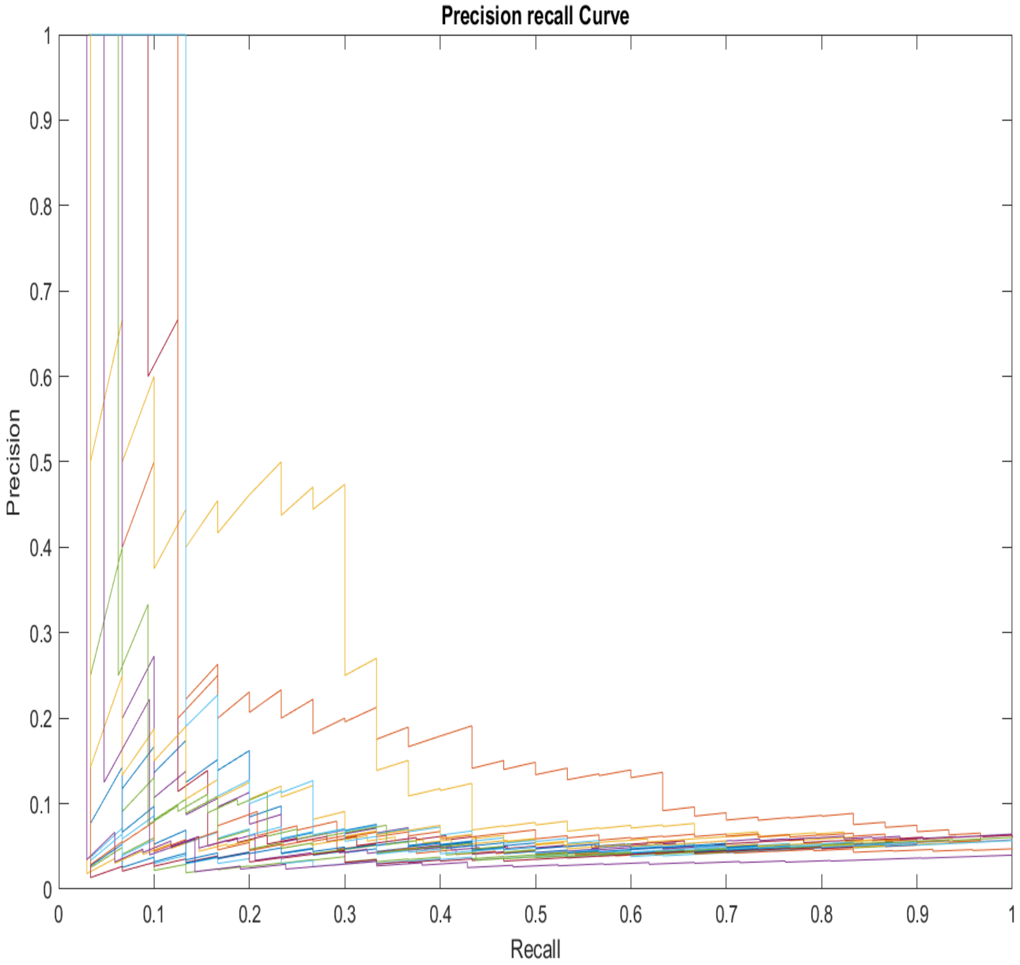


Figure 14: Precision recall curve of each image in 20 categories. Query image is compared with 591 test images. As recall reaches one, precision is in 0.05 to 0.1 range of all classes.

9.3. Spatial texture

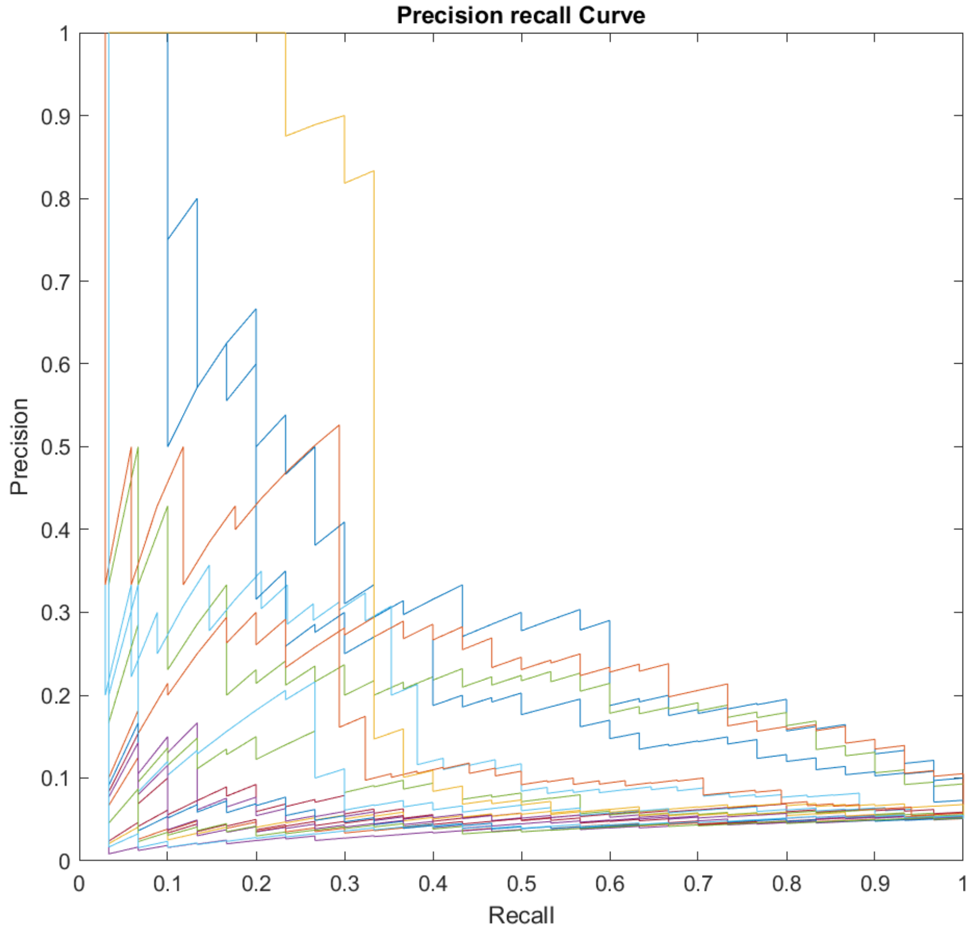


Figure 15: Precision recall curve of each image in 20 categories. Query image is compared with 591 test images. As recall reaches one, precision is in 0.05 to 0.18 range of all classes.