Chapter 1

INTRONDUCTION

Artificial vision systems have fascinated humans since pre-historic times. The earliest mention of an artificial visually guided agent appears in classical method where a bronze giant named Talo was created by the ancient god Hephaestus and was given as a gift to King Minos of the Mediterranean island of Crete [24]. According to legendth robot served as a defender of the island from invaders by circling the island three times a day, while also making sure that the laws of the land were upheld by the island's inhabitants the fascination and interest for vision systems continues today unabated, not only due to purely intellectual reasons related to basic research, but also due to the potential of such automated vision systems to drastically increase the productive capacity of organizations. Typically, the most essential component of a practical visually guided agent is its object recognition module. General a object recognition system can be defined as a computer application or a system to identifying or verifying a object automatically from a digital image or a video frame from a video source.

1.1BACKGROUND

In Recent Years, Object recognition from single intensity images has received a significant attention. Object Recognition has considerable importance for industrial automation; a robot must recognize the different objects in a scene before being able to manipulate them in any useful manner. Furthermore, it is necessary to compute the position and orientation, or location, of the identified objects in order to perform tasks such as grasping, pick-and-place, and assembly. Apart from its engineering applications, object recognition is of great scientific interest. Human beings possess the ability to recognize objects naturally and effortlessly even when the objects are partially occluded or are represented only by their outlines. If the same task is presented to an artificial system, however, one soon begins to appreciate the enormous complexity of the problem.

Different perceptual modalities can be used to gather information from a scene of interest. Examples include laser range finders, ultrasonic sensors, infrared sensors, and charge-coupled device (CCD) cameras. In this thesis, single intensity images acquired with a CCD camera are used. Multiple images can also be used for object recognition in applications involving motion, such as tracking an object in a sequence of images, or stereo vision.

1.2 MOTIVATION

Object recognition has recently received a blooming attention and interest from the scientific community as well as from the general public. The general public has the interest mostly due to the unusual events like terrorist attack around the world, so this becomes our need for useful security systems .Object recognition applications are far from limited to security systems as in the field of face recognition in the community of the different persons.

1.3 OBJECT RECOGNITION SYSTEM

Despite the evident success of recognition systems that are tailored for specific tasks, robust solutions to the more general problem of recognizing complex object classes that are sensed under poorly controlled environments remain elusive. Furthermore, it is evident from the relevant literature on object recognition algorithms that there is no universal agreement on the definitions of various vision subtasks. Often encountered terms in the literature such as:

Detection, localization, recognition, understanding, classification, categorization, verification and identification, are often ill defined, leading to confusion and ambiguities.

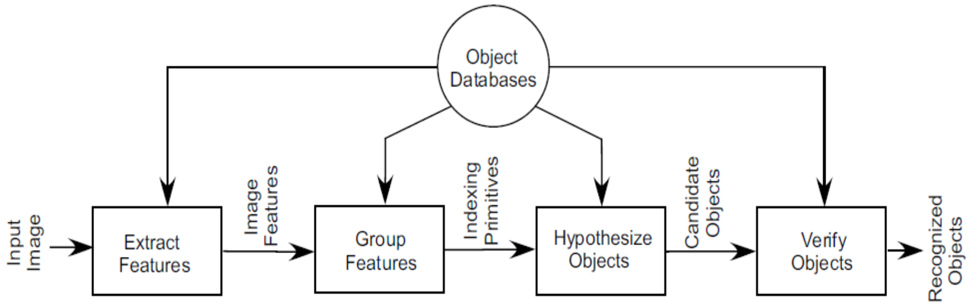


Figure 1.1: Basic block diagram of object recognition

Vision is popularly defined as the process of discovering from images what is present in the world and where it is [23]. Within the context of this report, we discuss four levels of tasks in the vision problem [24]:

Detection: Is a particular item present in the stimulus?

Detection and Localization: Given a complex image, decide if a particular exemplar object is located somewhere in this image, and provide accurate location information on this object.

Classification: Given an image patch, decide which of the multiple possible categories are present in that patch.

Naming: Given a large complex image (instead of an image patch as in the classification problem) determines the location and labels of the objects present in that image.

Description: Given a complex image, name all the objects present in the image, and describe the actions and relationships of the various objects within the context of this image. As the author indicates, this is also sometimes referred to as scene understanding.

Within the context of this thesis we will discern the detection, localization, recognition and understanding problems, as previously defined.

For relatively small object database sizes with small inter-object similarity, the problem of exemplar based object detection in unoccluded scenes, and under controlled illumination and sensing conditions, considered solved by the majority of the computer vision community. Great works have also been made towards solving the localization problem. Problems such as occlusion and variable lighting conditions still make the detection, localization and recognition problems a challenge. Tsotsos [34] and Dickinson [35] present the components used in a typical object recognition system: that is, feature extraction, followed by feature grouping, followed by object hypothesis generation, followed by an object verification stage (see Fig. 1). The advent popularity of machine learning approaches and bags-of-features types of approaches has blurred somewhat the distinction between the above mentioned components.

It is not uncommon today to come across popular recognition approaches which consist of a single feature extraction phase, followed by the application of cascades of one or more powerful classifiers. The definition of an object is somewhat ambiguous and task dependent, since it can change depending on whether we are dealing with the detection, localization, recognition or understanding problem. According to one definition [26], the simpler the problem is (i.e., the further away we are from the image understanding problem as defined above), the closer the definition of an object is to that of a set of templates defining the features that the object must possess under all viewpoints and conditions under which it can be sensed. As we begin dealing with more abstract problems (such as the object understanding problem) the definition of an object becomes more efficient and dependent on contextual knowledge, since it depends less on the existence of a finite set of feature templates. For example, the object class of toys is significantly abstract and depends on the context for a characterization of what might constitute a proper definition of an object. It is important to emphasize that there were multiple starting points that one can identify for early definitions of what constitutes an object, since this is highly dependent on the recognition system used. As previously discussed, one early starting point was work on the block-world system which led to definitions and generalizations involving 3D objects. However, there were also other significantly different early definitions, which emerged from early applications on character and chromosome.

1.4 Different approaches for object recognition

Object recognition techniques can be classified in many ways. One possible classification at high-level is:

1.4.1 Region based approach

These methods are more effective than holistic methods. They work on the principle of applying different processing methods to distinct object regions and hence filter out those regions that are mostly affected by expression changes or spurious elements. In other words they partition the object surface into regions and extract appropriate descriptors for each of them. But, these are also sensitive to object alignment which makes useful image regions hard to detect automatically. Also, local features and differences in resolution determine their performance.

1.4.2 Holistic approach

In holistic methods, the object as a whole is taken as input data. One of the main algorithms that come under this category is the Eigen vector method. Eigen vector method is based on the implementation of Principal Component Analysis (PCA). In this technique, we find the features of the studied images by looking for the maximum deviation of each image from the mean image. To obtain this variance we get the eigenvectors of the covariance matrix of all the images. The Eigen vector space can be obtained by applying the Eigen vector method to the training images. After that, we project the training images into the Eigen vector space. Later, we project the test image into this new space and the distance of the projected test image to the training images is used to classify the test image. Other examples of holistic methods are fisher faces and support vector machines.

1.4.3 Hybrid and multimodal approach

The idea of this method comes from how human vision system sees both object and local features. There are many examples in hybrid approach e.g. Modular Eigen vectors and component based methods. Even though there are wide range of algorithms available for both object detection and recognition. Matching these algorithms on to our embedded system will be a real challenge.

1.5 Scope of the work

In this work, we have focused on the problem of object recognition and have derived a novel way for recognizing object using Scale-invariant feature transform. We have a database of objects. In first part there is a test object in training image and Second part is test image. The SIFT features are generated for every training image and then k nearest neighbours classifier is used for matching scheme. The recognition results demonstrate its robust performance under different expression conditions, rotation variation, illumination changes and partial occlusion.

1.6 Organization of the thesis

The remaining part of this thesis is organized in the following chapters:

Chapter 2- Literature survey:

In this section, the critical points of current knowledge on object recognition have been reviewed. Substantive findings as well as theoretical and methodological contributions in the field of object recognition have been included. This section summarizes the work done in the field of object recognition using various techniques in recent times.

Chapter 3- Proposed object recognition system:

In this section, image pre processing has been discussed. Also, detailed description about the method of feature extraction used has been provided. The Scale-invariant feature transform is used for feature extraction. Along with this, the methods of classification also have been discussed. These include k nearest neighbour's classifier etc.

Chapter 4- Experiments and results:

All the experiments conducted along with the analysis of the results have been explained in detailed. The section is divided into two parts, feature extraction and classification. Figures and tables have been provided wherever necessary.

Chapter 5- Conclusion and Future Scope:

In this section we will discuss the conclusion of the thesis work and the future scope of the work.

References: This section gives the reference details of the thesis.

Appendix A: Abbreviations

Appendix B: Introduction to Image Processing in MATLAB Improved

Chapter 2

LITERATURE SURVEY

In Computer Vision, Object recognition is widely used for the purposes of inspection, registration, and manipulation. The current commercial systems for object recognition depend almost exclusively on correlation-based template matching. Template matching becomes computationally infeasible when object rotation, scale, illumination, scale and 3D pose are allowed to vary and also when dealing with partial visibility and large model databases. An alternative to search all image locations for matches is to extract features from the image that are at less partially invariant to the image formation process and matches only those features. Many candidate feature types have been proposed and explored, including line segments [1], groupings of edges [2, 3], and regions [4]. These features worked well for certain object classes but they are not quite stable to forma basis for reliable recognition.

One approach is to use a corner detector (more accurately, a detector of peaks in local image variation) to identify repeatable image locations by which local image properties could be measured. Zhang *et al.* [5] used the Harris corner detector to identify feature locations for epipolar alignment of images taken from differing viewpoints. Rather than attempting to correlate regions from one image against all possible regions in a second image, large savings in computation time were achieved by only matching regions centred at corner points in each image.

Schmid& Mohr [6] also used the Harris corner detector to identify interestpoints, and created a local image descriptor attract interest point from an orientation-invariant vector of derivative-of-Gaussian image measurements. These image descriptors were used for robust object recognition by looking for multiple matching descriptors that satisfied object based orientation and location constraints. This work was impressive both for the speed of recognition in a large database and the ability to handle cluttered images.

Other approaches to appearance-based recognition include Eigen space matching [7], colour histograms [8], and receptive field histograms [9]. All these approaches have been demonstrated successfully on isolated objects or pre-segmented images, but it has been difficult to extend them to cluttered and partially occluded images due to their more global features. Ohba & Ikeuchi [10] successfully applied the Eigen space approach to cluttered images by using many small local Eigen-windows, but an expensive search is required for each window in a new image, similar to template matching.

In the traditional formulation, a vision system captures multiple instances of an object from a set of object classes, and is asked to classify a new test image that may contain one or many known object classes. Successful method shave been demonstrated in the past, including pedestrian detection [11], general object detection [12,13] (e.g., vehicle sand animals), and scene annotation [14,15] (e.g., buildings, highways, and social events). These works have been based on analysis of certain local image patches that are robust invariant to image scaling, affine transformation, and visual occlusion, which are the common nuisances in image based object recognition. The local image patches are typically extracted by a viewpoint-invariant interest point detector [16] combined with a patch descriptor, e.g., SIFT (Scale-Invariant Feature Transform) [17,18].

Chapter 3

PROPOSED OBJECT RECOGNITION SYSTEM

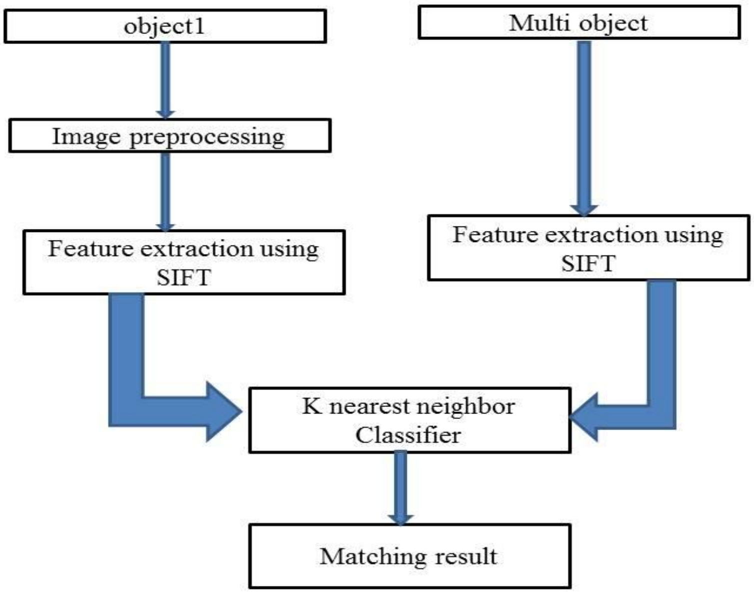


Figure3.1: flow diagram of proposed system

An image is extracted from database and the required pre processing is done on it. Pre processing helps in improving the results obtained from the feature extraction methods. Thereafter, features are extracted using the appropriate methods and feature vectors are formed. These feature vectors are then classified using various classifiers. All the above mentioned steps have been elaborated in this chapter.

3.1 Image pre processing

Image pre processing commonly comprises of a series of sequential operations including image enhancement or normalization, geometric correction, removal of noise etc. It is done before applying the feature extraction techniques. Using the above mentioned techniques, we alter the image pixel values permanently and use this improved data for later analyses.

3.1.1 Need for pre processing

Image pre processing more importantly is done for noise removal which is mainly added during image acquisition. Image acquisition is a very important step for the quality control since it provides the input data for the whole process. Digital image is acquired by an optical sensor which is always a video camera (with one line or a matrix of CCD). It provides accurate and noiseless image. Local illumination is also directly linked with the quality of image acquisition.

Sometimes there might be other reasons also. There may be a need to convert a colored image into gray scale. Image enhancement by various methods also comes under image pre processing.

Various image pre processing methods

(i) Noise removal

Images taken with both digital cameras and conventional film cameras pick up noise. It is the result of errors in the image acquisition process that result in pixel values that do not reflect the true intensities of the real scene from a variety of sources. For further use of these images, it is required that the noise should be removed. For removal of noise various filters can be applied on images, like high pass filter, Gaussian filter etc.

(ii) Contrast adjustment

The contrast of an image refers to the distribution of the dark and light pixels. A low-contrast image means there is a small difference between its light and dark pixel values. This in turn means that the histogram of a low-contrast image is narrow. Since the human eye is sensitive to contrast rather than absolute pixel intensities, a better image can be obtained by stretching the histogram of the image so that the full dynamic range of the image is used.

(iii) Intensity adjustment

Image enhancement techniques are used to improve an image. These techniques can be used to increase the signal-to-noise ratio or make certain features easier to be seen by modifying the colours or intensities. Adjustment of intensity is an image enhancement technique which maps the image intensity values to a new range.

(iv) Histogram Equalization

Histogram equalization distributes evenly the occurrence of pixel intensities so that the entire range of intensities is occupied. This method usually increases the contrast of images globally, especially when the required data of the image is represented by close contrast values. By adjusting the histogram, the intensities can be better distributed. Due to this, the areas of lower local contrast gain a higher contrast. Histogram equalization effectively spreads out the most frequent intensity values.

(v) Normalization

Image normalization is done in order to reduce the amount of computation so that the method is more efficient. The pixel values ranging from 1 to 255 are mapped to lie within 0 to 1.

3.2 Feature extraction

In image processing, feature extraction is a special form of dimensionality reduction. It simplifies the amount of resources required to describe a large set of data accurately. Image features can refer to the global properties of an image like average gray level, shape of intensity histogram etc. or the local properties of the image like shape of contours, elements composing textured region etc.

To choose the features to be extracted is critical and the following concerns should be kept in mind while selecting them.

1) Enough information about the image should be carried by the features and there should be no requirement of any domain specific knowledge for their extraction.

2) Their computation should be easy so that the approach is feasible for a large image database and rapid retrieval.

3) They should be meaningful i.e. they should be associated with interesting elements in the image formation process. They should be invariant to the image formation process like invariant to the viewpoint and illumination of the captured digital images.

4) They should relate well with the human perceptual characteristics since the end users will finally determine the suitability of the images retrieved.

5) The extracted features should be able to be detected or located from the images with the help of algorithms. Also they should be easily described by a feature vector.

In this work, the features are extracted from images using Scale invariant feature transform (SIFT) methods. The extracted features are then subjected to classification.

3.3 Scale Invariant Feature Transform

The feature extraction technique, done by Scale invariant Feature transform has such advantages as scale invariability, rotation invariance, affine invariance, which could reduce mismatching because of occlusion, confusion, noise, and keep high matching rate. The detailed step of this method is as follows.

In first step Extreme value or key point is detected in the different scale of the image, secondly, the position of key-point is located by the filter of the key-points; thirdly, the key-points gradient direction is determined by the key-point neighbourhood; fourthly, the feature descriptor is determined by the feature of the key-point .

3.3.1. Extreme Value Detected in Scale Space

In [44] SIFT, the scale transformation is realized by the Gaussian convolution kernel, and the description $L(x, y, \sigma)$ of input image $I(x, y)$ in different scale can be expressed by the

$$L(x,y,\sigma)=G(x,y,\sigma)*I(x,y)\text{……………………….}(1)$$

The 2D Gaussian convolution kernel $(G(x, y, \sigma))$ is described

$$G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{x^2+y^2 / 2\sigma^2} \text{......................}(2)$$

According to (1) and (2), the description of $I(x, y)$ in different scale is calculated by its convolution with a Gaussian filter, and the Gaussian function is only the kernel function. After the convolution, the calculated image is a Gaussian image. (x, y) represents the pixels coordinate of the input image; σ is the scale factor; $L(x, y, \sigma)$ is the spatial scale images. The image I(x, y) zoom with σ, and the smoothness of the image would change with de change of σ, and then a series of scale image could be obtained. According to those scale images, the extreme point (key- point) will be detected. The difference of Gaussian (DOG) image between scale $k\sigma$ and σ is as in

$DOG(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y)$

$= L(x, y, k\sigma) - L(x, y, \sigma)$ ……………….(3)

By (3), the feature of image, such as contour, corner, edges, can be detected in different scale, and the candidate key-points of image are chosen by the extreme value of the features. That is: by comparing every point with its neighbourhood which includes adjacent scale, the position and scale of local the extreme point are the key-points“. The schematic diagram is Fig.

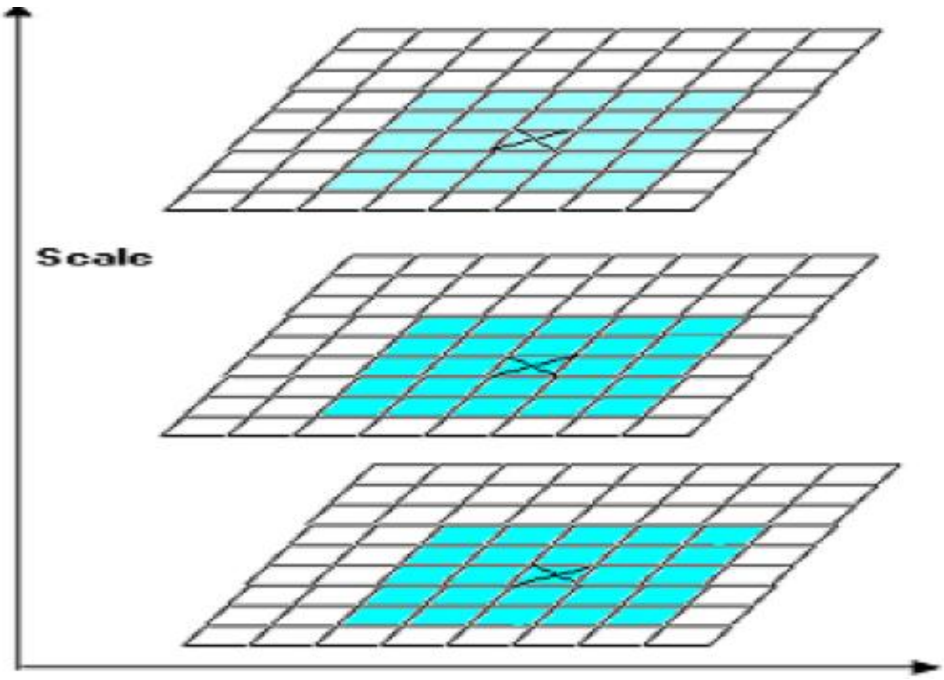


figure 3.2: key point detection

3.3.2. Location of Key-points

The location of key-point is considered to filter the key- points which are sensitive to noise or have no edge effect in this process. The reference shows that, according to Taylor quadratic expansion, *DOG(x,y,σ)* can delete the extreme points which have lower contrast, and the value of Hessian vector and the ratio of determinant can reduce the edge effect.

3.3.3. Interpolation of nearby data for accurate position

First, for each candidate key point, to determine its position accurately, Interpolation of nearby data is used. The initial approach was to just locate each key point at the location and scale of the candidate key point. The new approach calculates the interpolated location of the extremum to improve matching and stability. For interpolation, Quadratic Taylor expansion of the Difference-of-Gaussian scale-space function is applied, D(x, y, σ) with the candidate key point as the origin. This Taylor expansion is given below:

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}}\mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2}\mathbf{x} \quad \text{..............(4)}$$

Where D and its derivatives are evaluated at the candidate key point and X=(x, y, σ) is the offset from this point. By taking the derivative of this function with respect to X and setting it to zero The location of the extremum, , is determined. If the offset is larger than in any dimension, it indicates that the extremum lies closer to another candidate key point. In this case, the candidate key point is changed and the interpolation performed instead about that point. Otherwise the offset is added to its candidate key point to get the interpolated estimate for the location of the extremum. A similar sub pixel determination of the locations of scale-space extrema is performed in the real-time implementation based on hybrid pyramids developed by Lindeberg and his co-workers.

3.3.4. Discarding low-contrast key points

The value of the second-order Taylor expansion D(X) is computed at the offset to remove the key points with low contrast, If this value is less than , the candidate key point is removed. Otherwise it is kept, with final location and scale , where y is the original location of the key point at scale .

3.3.5. Eliminating edge responses

Even if the candidate key point is not robust to small amounts of noise, the DoG function will have strong responses along edges. Therefore, in order to increase stability, we have to eliminate the key points that have poorly determined locations but have high edge responses.

For poorly defined peaks in the DOG function, the principal curvature across the edge would be much larger than the principal curvature along it. To Find these principal curvatures amounts to solving for the eigen values of the second-order Hessian matrix,

$$\mathbf{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$$

The eigen values of **H** are proportional to the principal curvatures of D. It is the ratio of the two eigen values let suppose is the larger one α, and the smaller one β, with ratio α/β , is sufficient for SIFT's purposes. The trace of **H**, i.e. $D_{xx} + D_{yy}$ is the sum of the two eigen values, while its determinant, i.e $D_{xx}D_{yy} - D_{xy}^2$, is the product. The ratio $\mathrm{R} = \mathrm{Tr}\,(\mathbf{H})^2 \,/\, \mathrm{Det}\,(\mathbf{H})$ equal to $(r+1)^2/r$, which depends only on the ratio of the eigen values rather than their individual values.

R should be minimum when the eigen values are equal to each other. Therefore If the absolute difference between the two eigen values is high which is equivalent to a higher absolute difference between the two principal curvatures of D, the higher the value of R. It follows that, for some threshold eigen value ratio, if R for a candidate key point is larger than $(r_{\text{th}} + 1)^2 / r_{\text{th}}$, that key point is poorly localized and hence rejected. The new approach uses $r_{\text{th}} = 10$.

This processing step for suppressing responses at edges is a transfer of a corresponding approach in the Harris operator for corner detection. The difference is that the measure for thresholding is computed from the Hessian matrix instead of a second-moment matrix (see structure tensor).

3.3.6. Direction of Key-point

After the position and scale of the key-point is determined, the next step is key-points direction, which can ensure the features rotation invariance. The direction is calculated by the image information of key-points neighborhood. Firstly, we give the expression of the module value (4) and phase (5).

m(x,y)=$\sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$ .....

$$\theta\,(x\,,\,y) = \arctan\left[\frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)}\right] \ldots\ldots\ldots\ldots(6)$$

Where, *L* is the scale of feature, and the calculating step is as follows:

• Choose a neighbourhood ***M*** by the center of the key-point;

• Calculate the directions of points in ***M*** by (6);

• Obtain the direction distribution, and draw the statistical histogram; . The direction of key-point is the maximal component in the statistical histogram.

Since the result of (6) is a real number, we divide the real interval ([0,2π]) into 36 portions by π/18 so as to be convenient in obtaining the direction distribution, then the direction statistical histogram contains 36 phases, and the statistical rule is principle of proximity. Fig. 3.3 shows that the direction is determined by the statistical histogram, and the maximal component (π/2) is the key-points direction.

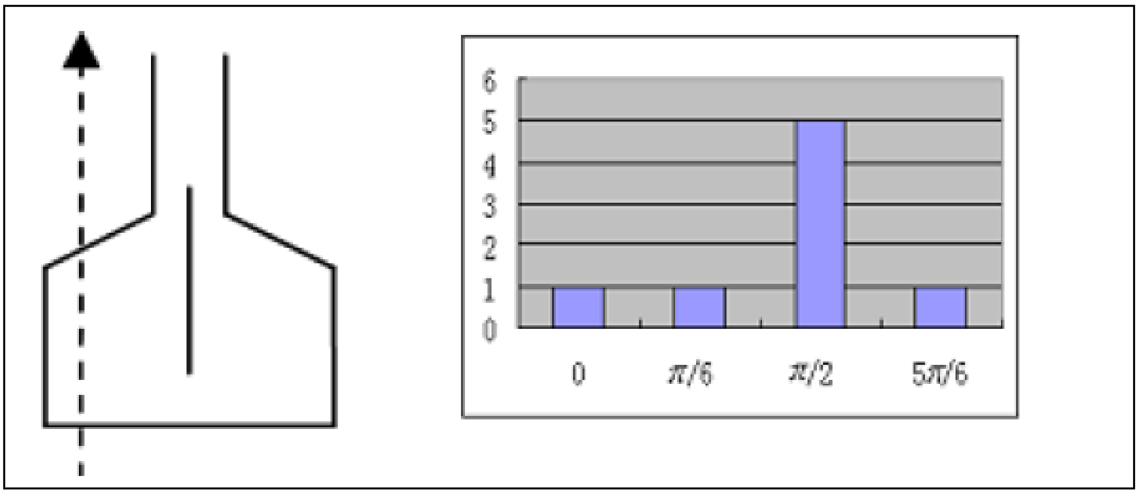


Figure 3.3: Direction of key-point (a) Original image (b) Statistical histogram

3.2.8. Feature Description of Key-points

The position, scale and direction of key-point can only ensure the 2D geometric invariance, but can’t ensure the lightness and view transformation invariance. So SIFT introduces feature description to solve those problems. The step is as follows:

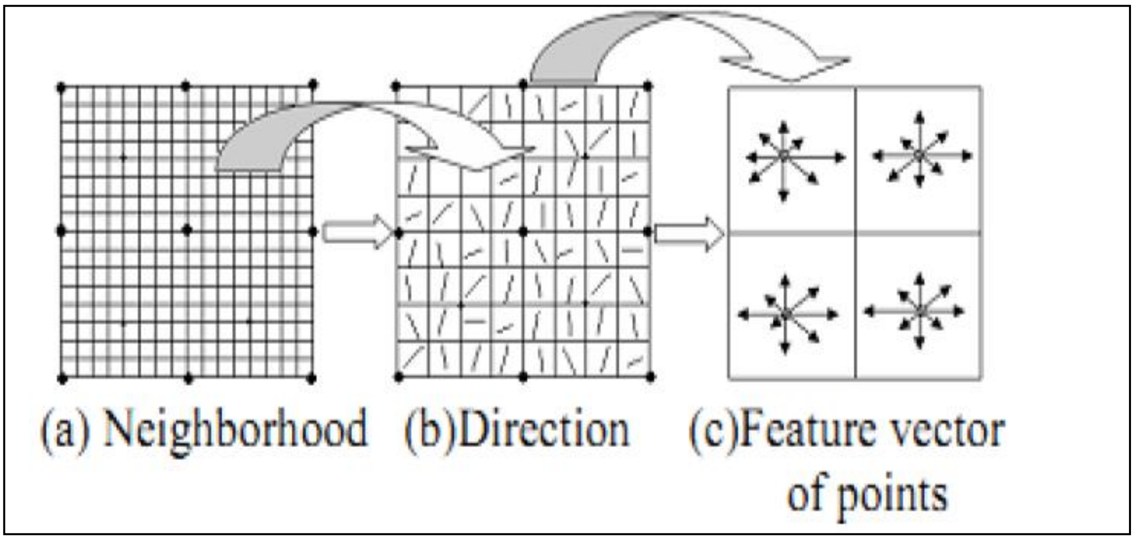


figure 3.4 :feature description of key points

• Select 16×16 pixel fields ***M*** in the neighbourhood of key-point, and divide ***M*** into 16 subfields by 4×4;

• According to (5) and (6), calculate the direction and amplitude in every subfield, and then the direction distribution in the ranges of [0, 45, 90, 135, 180, 225, 270, 315] could be obtained.

• According to the direction statistical histogram of subfields, 8 direction descriptions will be calculated by the amplitude and Gaussian function;

• The feature description is obtained by connecting the direction descriptions of all subfields; the total of the direction descriptions is 16, so the length of the feature description is 128=16×8; • In order to ensure the illumination invariance, the feature description should be normalized.

3.4 Feature Matching

The image matching is the feature matching. Firstly, the Euclidean distance between key-points from two pictures is calculated, and the two key-points with minimum distance is the candidate matching key-point pair. Secondly, calculate the ratio of the minimum distance: the second smallest distance; and if the ratio is greater than the preset threshold,

The candidate matching key-point pair is a good key-point pair. Thirdly, if the number of matching key-point is greater than the preset threshold, the two images are matched.

3.5 Classification

Image classification involves analyzing the numerical properties of various image features and organizing the data into categories. Usually, two phases of classification algorithms are employed: training and testing.

In the initial training phase, characteristic properties of typical image features are identified and, a unique description based on this of each classification category, i.e.

training class, is created. The description of training classes is an extremely important component of the classification process.

The criteria for constructing training classes are that they are:

1. independent, i.e. a change in the description of one training class should not change the value of another,

2. discriminatory, i.e. different image features should have significantly different descriptions, and

3. Reliable, i.e. all image features within a training group should share the common definitive description of that group.

In the next phase i.e. testing, accuracy of the classifier is measured. The accuracy can be determined by applying the classifier to an independent training set of objects with known classifications. Knowledge of the accuracy is necessary both in the application of the classifier and also in comparison of different classifiers.

A convenient way of building a parametric description of this sort is via a feature vector, where „n" is the number of attributes which describe each image feature and training class. This allows us to consider each image feature is occupying a point, and a sub-space (i.e. a representative point surrounded by some spread, or deviation) is occupying by each training class, within the n-dimensional classification space. The classification problem is for determining to which sub-space class each feature vector belongs.

Classification can be linear or nonlinear. Linear classifiers are usually the fastest classifiers. It classifies the data on the basis of a linear combination of the characteristics based on which the classification is done. The classes are divided by a linear separator in the feature space. If the feature space is two dimensional, the separator is a line, it three dimensional the separator is a plane and if „p" dimensional then the linear separator is a (p-1) dimensional hyper plane. Nonlinear classification is required when the class boundaries cannot be approximated well with linear hyper planes. Example of nonlinear classifier is K Nearest Neighbour.

3.5.1 Methods of classification

There are mainly two ways of classification, supervised and unsupervised. The difference between them lies in the fact that how the data is classified.

3.5.1.1 Supervised classification

In supervised classification, there are predetermined classes. Statistical processes (i.e. based on an a priori knowledge of probability distribution functions) or distribution free processes can be used to extract class descriptors. These classes can be regarded as a previously decided finite set. After classification, certain segment of data will be labeled with these classes. The task of the algorithm is to search for patterns and construct mathematical models. These models are then used to find out the measure of variance in the data and classify it. The examples of supervised classification are decision tree induction, naïve bayes classifier etc.

3.5.1.2 Unsupervised classification

Unsupervised classification relies on clustering algorithms to automatically segment the training data into prototype classes. In this type of classification, the classes are not pre decided. The basic task of the classifier is to develop the classes or the labels automatically. The algorithm is not told how the data is to be grouped; it is something it has to arrive at by itself. This is a difficult decision to make. The classifier looks for similarities between data and then determines which of these can form a group and can be classified under one label. The classes are also called clusters. The example of unsupervised classification is K-means classification. In K-means, the classifier is told in advance, the no of clusters have to be formed.

In unsupervised classification, since there is no defined strategy, the algorithm starts from one point and performs iterative repetitions to reach a stable configuration that makes sense. The results can vary widely and depend largely on the first few steps taken.

3.5.2 Examples of classifiers

The most common classifiers used are:

3.5.2.1 Support vector machine (SVM)

Support vector machine (SVM)[19] is an example of supervised learning classification. It is a binary linear classifier which takes an input and decides to which of the two classes it belongs. The classifier s first trained using a set of training examples. The training examples are pre marked as belonging to one of the two categories and based on these examples, the SVM classifier builds a model that assigns new examples (data to be tested) their suitable classes. The training examples are represented as points in space and are mapped such that there is a clear gap which divides the examples belonging to separate classes. The new examples are then mapped into the same space by analyzing to which of the two classes they suit better.

Apart from classification, a support vector machine can also be used for regression etc. It constructs a hyper plane that separates the two classes with gap as wide as possible. A hyper plane is regarded as „good" if it has the largest distance with the nearest training data point of any class. This reduces the error of the classifier.

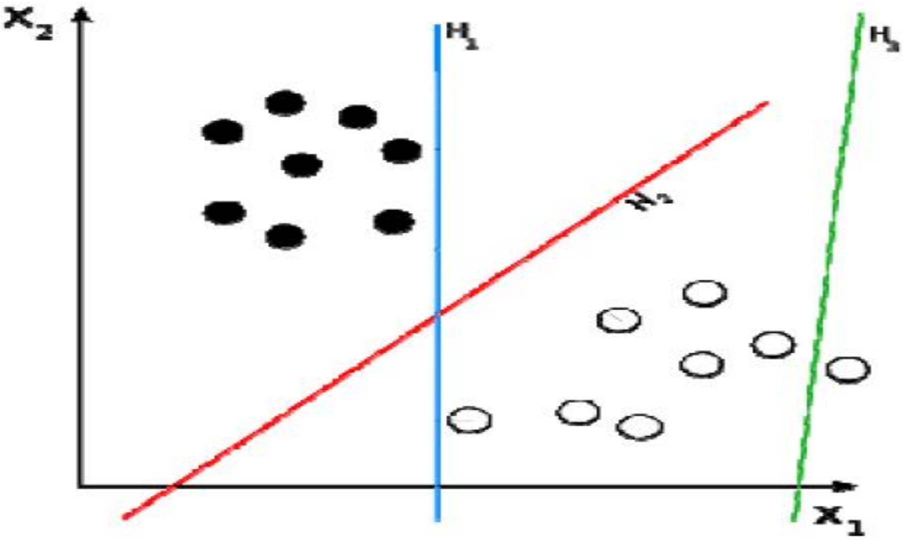


Figure 3.5: Hyper planes separating two classes

In the above figure, we see that although red and blue hyper planes both are separating the two classes entirely, the red hyper plane is doing it such that its distance from the nearest points of the two classes is maximum. Hence the red hyper plane is the most optimum hyper plane. The data points which are closest to the hyper plane are called support vectors.

Nonlinear SVM

Nonlinear SVM is applied in cases where the data sets have nonlinear decision boundaries. The trick applied here is to transform the data from its original coordinate space to a new space where a linear decision boundary can be used to separate the instances in the transformed space. Transformation of space may suffer from dimensionality problem which is associated with high dimensional data. Moreover, it is not always easy to find out what mapping must be used to ensure that a linear decision boundary can be constructed in the transformed space. This problem can be solved by „Kernel" trick. It is a method for computing similarity in the transformed space using the original attribute set. The measure of similarity used is the dot product of the two input vectors. The similarity function computed in the original attribute space is called the „kernel function‟. The kernel trick eliminates the need to know the exact form of the mapping function.

Multi class SVM

Multi class SVM classifies data into more than one class. For multiclass case, we transform the problem into multiple binary classification problems.

3.5.2.2 Minimum distance classifiers

The minimum distance classifiers[20] as the name suggests, classify the data on the basis of the distance between the data points in the feature space. It classifies in such a manner, so as to minimize the distance between the data and the class in multi-feature space. The index of similarity here is the distance so that the minimum distance is identical to the maximum similarity. The most common distances often used in this procedure are:

Euclidian distance

The Euclidean distance is also known as the Euclidean metric that is distance between two points that is measured with a ruler, and is given by the Pythagoras formula. By using this formula as distance, Euclidean space or any other inner product space becomes a metric space. The norm of the Euclidian distance is called the Euclidean norm.

It is used in those problems where the variances of the population classes are different from each other. Theoretically the Euclidian distance is identical to the similarity index. The Euclidean distance between two points „p‟ and „q‟ is the length of the line segment connecting them.

If p = ($p1$, $p2$,..., pn) and q = ($q1$, $q2$,..., qn) are two points in Euclidean n dimensional space, then the Euclidian distance from p to q is given by:

$$d(p, q) = d(q, p) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 \ldots \ldots \ldots \ldots + (p_n - q_n)^2}$$
$$= \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \ldots\ldots\ldots\ldots (7)$$

The position of a point in a Euclidean *n*-space is shown by a Euclidean vector. So, Euclidean vectors p and q, starting from the origin of the space. The Euclidean length, or Euclidean norm, or magnitude of a vector measures the length of this vector and is given by

$$x(t), \frac{1}{\sqrt{A}} y\left(\frac{t-\tau}{A}\right), z(t) \ldots\ldots$$

Here the last equation involves the dot product.

Mahalanobis distance

Mahalanobis distance was introduced by P. C. Mahalanobis in 1936. Its basis is the correlations between variables by which different patterns can be identified and analyzed. It measures the similarity of an unknown sample set to a known one. It differs from Euclidean distance in the manner that it takes into account the correlations of the data set and is scale-invariant. To put in other words, it is a multivariate effect size.

The formal definition of the Mahalanobis distance of a multivariate vector is

$$x = (x_1, x_2, \ldots\ldots\ldots\ldots, x_n)^T$$

$$D_M(x) = \sqrt{(x - \mu)^T}$$

$$x = (x_1, x_2, \ldots \ldots \ldots, x_n)^T$$

$$D_M(x) = \sqrt{(x - \mu)^T S^{-1} (x - \mu)} \ldots\ldots$$

Where, μ is the mean vector represented as $\mu = (\mu_1, \mu_2)^T$ and S is the covariance matrix.

Mahalanobis distance also called "generalized squared interpoint distance". Its squared value is defined as a dissimilarity measure between two random vectors x and y of the same distribution with the covariance matrix S as:

$$d(\bar{x},\bar{y}) = \sqrt{(\bar{x}-\bar{y})^T S^{-1}(\bar{x}-\bar{y})} \ldots$$

If the covariance matrix is the identity matrix, the Mahalanobis distance becomes same as the Euclidean distance. On the other hand, if the covariance matrix is diagonal, the resulting distance measure is called the normalized Euclidean distance.

3.5.2.3 Artificial Neural networks (ANN)

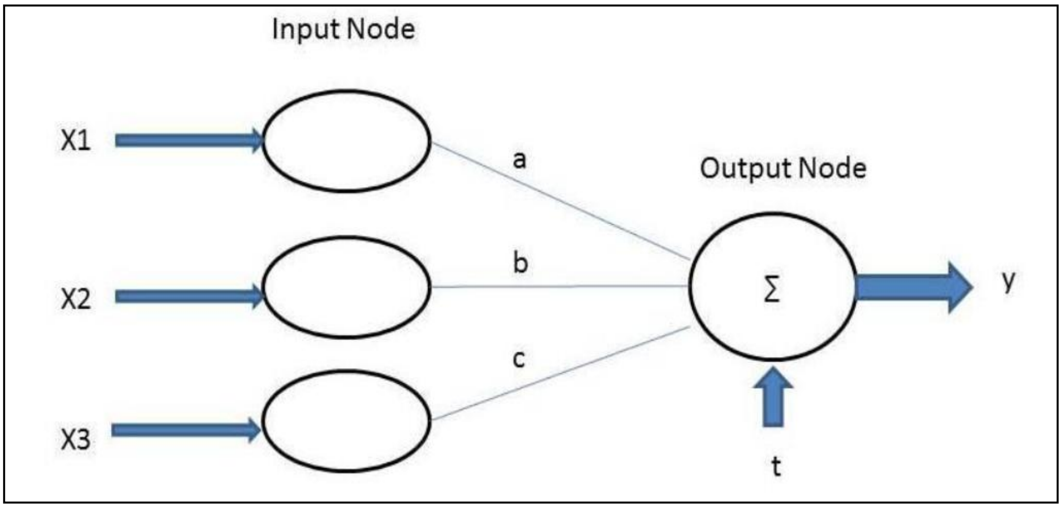


Figure 3.6: Model of a perceptron

Artificial neural network classifiers[21] are inspired by biological neural systems. The nerve cells in human brain are called neurons. These are linked with other neurons via strands of fiber called axons. Whenever the neurons are stimulated, axons transmit nerve impulses from one neuron to another. Extensions from the cell body of the neuron are called dendrites. Dendrites connect one neuron to the axons of other neurons. The connection between a dendrite and an axon is called a synapse. It has been discovered that the human brain learns by changing the strength of the synaptic connection between neurons when stimulated repeatedly by the same impulse.

The ANN has a structure analogous to the human brain. It is composed of an interconnected assembly of nodes and direct links. These models can be trained for the purpose of classification. The simplest model is called perceptron.

The perceptron consists of two kinds of nodes, the input nodes and the output node. Input nodes are used to represent the input attributes and the output node represents the model output. The nodes in the neural network are called neurons are units. In a perceptron, each of the input nodes is connected to the output node via a weighted link. The weight of the link is used to emulate the strength of the synaptic connection between neurons. The perceptron computes the output value by calculating a weighted sum of the inputs, subtracting a bias factor „t" from the sum and then examine the sign of the result. Training of the perceptron network involves changing and adapting the weight of the links until they fit the input output relationship of the training data i.e. the outputs of the perceptron become consistent with the true outputs of training data.

In an ANN model, an input node simply transmits the value it receives to the outgoing link without any transformation. On the other hand, the output node is a mathematical device which computes the weighted sum of its inputs subtracts the bias term and produces an output that depends on the sign of the result. The sign function is an activation function for the output neuron, it value is +1 if positive and -1 if negative. Once the model is trained, it can now be used to test new examples and classify them. The perceptron learning algorithm converges to an optimum solution for linearly separable classification problems. If the problem is not linearly separable the algorithm fails to converge and in that case, multilayer artificial neural network is needed.

A multilayer neural network has a more complex structure than a perceptron. It contains several intermediate layers between its input and output layers. These intermediate layers are called hidden layers and the nodes in these layers are called hidden nodes. There are two types of networks, feed forward and recurrent networks. In feed forward networks, the nodes in one layer are connected only to the nodes in the next layer while in a recurrent network; the links may connect the nodes in the same layer or in the previous layers. The activation function may be other than sign function also like linear or sigmoid function. These activation functions allow the hidden and the output nodes to produce nonlinear outputs. This type of complex structure can classify problems which are not linearly separable and have nonlinear solutions. This model will converge to the right solution when sufficient training data is provided.

K Nearest neighbor (KNN) classifier

In pattern recognition and classification, the k-nearest neighbor algorithm (k-NN) [22], is an algorithm for classifying data objects based on closest training examples in the feature space. The KNN is a type of instance based or lazy learning, where the function is only approximated locally and all computation is postponed till classification. The k-nearest neighbor algorithm is simplest among all machine learning algorithms: an object is classified by a majority vote of its neighbouring data, and the test object is assigned to the class which is most common amongst its k nearest neighbours (k is a small positive integer). If „k‟ is chosen to be 1, then the test object is simply assigned to the class to which its nearest neighbour belongs.

Although there is no explicit need for training in this algorithm, the neighbours can be regarded as training examples and are chosen from a set of objects for which the correct classification is known. The k-nearest neighbor algorithm is affected by the local structure of the data. The nearest neighbour rules effectively compute the decision boundary in an implicit way.

The training examples can be regarded as vectors in a multidimensional feature space, each belonging to a class. So, in the training phase of the algorithm, it is required to only store the feature vectors of the training samples along with their class.

In the classification phase, k is a constant decided by the user and a test vector (can also be called query or a test point) is classified by assigning it the label which is most common among the k training data nearest to that particular test point.

Euclidean distance is most commonly used as the distance measure; however this only applies to the continuous variables. For other types of classification like text classification, other measures such as the overlap metric(Hamming distance) can also be used. The accuracy of the k nearest neighbour algorithm can be improved significantly by using special algorithms such as Large Margin Nearest Neighbour or Neighbourhood components analysis.

The concept of „majority voting‟ for classification has one drawback. If more examples in the training sample belong to one class, it tends to dominate the prediction of test sample, since their chances of being present in the k nearest neighbours is high owing to their large numbers. So the training examples have to be chosen very carefully. This problem can also be overcome by taking into consideration the distance from the test vector to each of the k nearest neighbours. Choosing the value of „k‟ also depends on the data. Larger values of „k‟ help in reducing the effect of noise by averaging it out. But it makes the boundaries between the classes less distinct. There are various techniques for selecting the optimum value of „k‟ like cross validation. The special case when the value of „k‟ is „1‟, i.e. the testing example is assigned the class to which its nearest neighbour belongs, is called the nearest neighbour algorithm.

Chapter4

EXPERIMENTS AND RESULTS

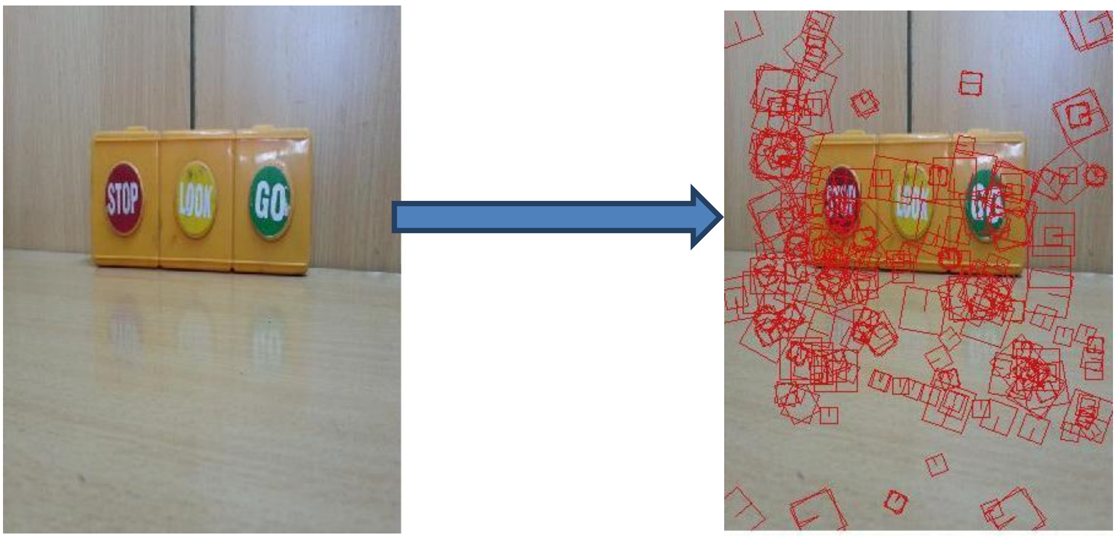
4.1 Image pre-processing

Image normalization is done in order to reduce computational overhead and improve the performance of the method. The images are gray scale images which contain intensity values ranging from 0 to 255. The pixel values of all the images are divided by the maximum pixel value of the image. The value of all the pixels lies between 0 and 1.

4.2 Feature extraction

Feature extraction of all the images is done using the SIFT. Some feature images are shown below:

OBJECT:1



OBJECT:2

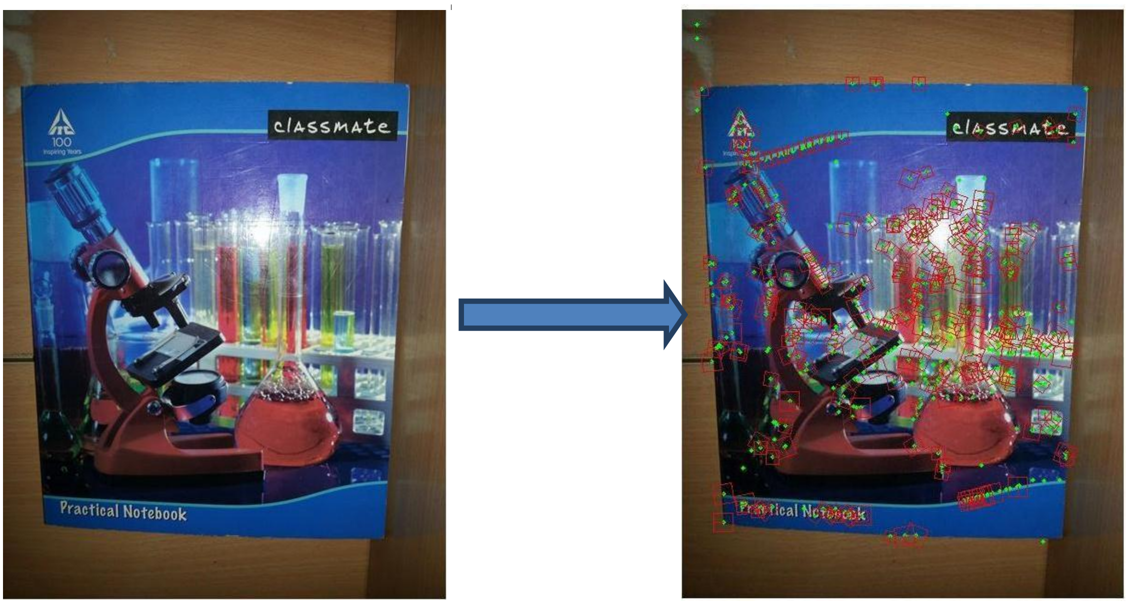


Figure 4.1 : detection of feature points

4.3 Result:

Matching with subject 1

Matching with subject 2

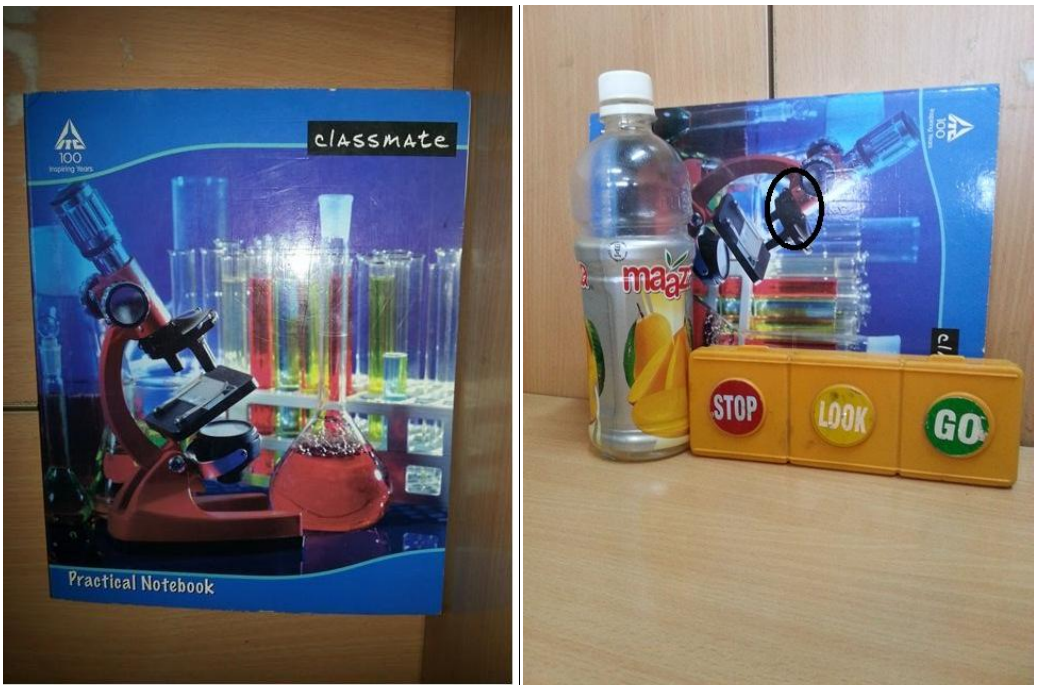


Figure 4.2: recognition of test object

We have taken the key features of test object using scale invariant feature transform. After this we match the key features in a occluded image using k- nearest neighbour's classifier .and we found the test object in a occluded image.

Chapter 5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

We have presented a critical overview of the object recognition literature, pointed out some of the major challenges facing the community and emphasized some of the characteristic approaches attempted over the years for solving the recognition problem. We began the survey by discussing how the needs of industry led to some of the earliest industrial inspection and character recognition systems. It is pleasantly surprising to note that despite severe limitations in CPU speeds and sensor quality, such early systems were astoundingly accurate, thus contributing to the creation of the field of computer vision, with object recognition assuming a central role. We pointed out that recognition systems perform well in controlled environments but have been unable to generalize in less controlled environments. Throughout the survey we have discussed various proposals set forth by the community on possible causes and solutions to this Problem .We continued by surveying some of the characteristic classical approaches for solving the problem. We continued the survey by discussing some common testing strategies and fallacies that are associated with recognition systems. We concluded by discussing in some depth of successful recognition system that have been openly tested in various object recognition challenges.

So in the presented work, a novel method for object recognition using SIFT (Scale invariant Fourier transform) has been proposed. This method is invariant to scaling contrast, illumination and partial occlusion. In our result we have observed the Equal error rate is 0.80, obtained by using SIFT. The Recognition rate is found 97.91% that show Robustness of this method. It shows all the matching between occluded images and test images. We have done the matching technique by using k nearest neighbours' classifier. We found that object Recognition using SIFT technique is robust and invariant to scaling, contrast, illumination and partial occlusion.

5.2 Future scope

As one of the most successful applications of image analysis and understanding, At present object recognition has gained so much attention. it has become a popular area of research in computer vision and one of the most successful applications of image analysis and understanding in last decade.

Object recognition is a technology just reaching sufficient maturity for it to experience a rapid growth in its practical applications. Much research effort around the world is being applied for expanding the accuracy and capabilities of this domain, with a consequent broadening of its application in the near future. Verification systems for physical and electronic access security are available today, but the future holds the promise and the threat of passive customization and automated surveillance systems enabled by object recognition. The latest example in face recognition systems in the world operates in the U.S. Department of State for visa processing. It consists of over 75 million photographs. There can be several other uses of object recognition in future. Some of them can be:

1. object recognition can be used to prevent ATM frauds. A database having unique codes of all ATM customers are being prepared with the banks. High resolution cameras will have to be deployed at all ATM and face recognition software has to be used. Now whenever a user will enter the ATM, his photograph will be taken and access to the ATM machine will be permitted only after the photo is matched with the database.

2. Similarly, passport and visa verifications can also be done using object recognition technology .Driving license verification can also be performed using object recognition technology as mentioned earlier.

4. object recognition technology will be the best choice in India as compared with other biometric technologies since other technologies cannot be helpful in crowds.

Along with all the advantages, object recognition technology has its own problems. With the wide spread installation of security cameras and the increasing financial and technological feasibility of automating this surveillance, public fear shave also increased about the potential for invasion of privacy that this technology can bring about.

REFERENCES

1. Grimson, Eric, and Thom´as Lozano-P´erez, “Localizing overlapping parts by searching the interpretation tree,” IEEE Trans. on Pattern Analysis and Machine Intelligence, **9** (1987), pp. 469–482.
2. Lowe, David G., “Three-dimensional object recognition from single two-dimensional images,” Artificial Intelligence, **31,** 3 (1987), pp. 355–395.
3. Nelson, Randal C., and Andrea Selinger, “Large-scale tests of a keyed, appearance-based 3-D object recognition system,” Vision Research, **38,** 15 (1998), pp. 2469–88.
4. Basri, Ronen, and David.W. Jacobs, “Recognition using region correspondences,” International Journal of Computer Vision, **25**, 2 (1996), pp. 141–162.
5. Zhang, Z., R. Deriche, O. Faugeras, Q.T. Luong, “A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry,” Artificial Intelligence, **78,** (1995), pp. 87-119.
6. Schmid, C., and R. Mohr, “Local grayvalue invariants for image retrieval,” IEEE PAMI, **19,** 5 (1997), pp. 530–534.
7. Murase, Hiroshi, and Shree K. Nayar, “Visual learning and recognition of 3-D objects from appearance,” International Journal of Computer Vision, **14,** 1 (1995), pp. 5–24
8. Swain, M., and D. Ballard, “Colour indexing,” International Journal of Computer Vision, **7,** 1 (1991), pp. 11–32.
9. Schiele, Bernt, and James L. Crowley, “Object recognition using multidimensional receptive field histograms,” Fourth European Conference on Computer Vision, Cambridge, UK (1996), pp. 610–619.

10. Ohba, Kohtaro, and Katsushi Ikeuchi, "Detectability, uniqueness, and reliability of eigen windows for stable verification of partially occluded objects," IEEE Trans. On Pattern Analysis and Machine Intelligence, **19,** 9 (1997), pp. 1043–48.
11. B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowdedscenes. In CVPR, pages 878– 885, 2005
12. S. Agarwal and D. Roth. Learning a sparse representation for objectdetection. In ECCV, 2002.
13. A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. VanGool. Towards multi-view object class detection. In CVPR, 2006.
14. C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. Freeman. SIFT flow: densecorrespondence across different scenes. In ECCV, 2008.
15. A. Quattoni, M. Collins, and T. Darrell. Transfer learning for imageclassification with sparse prototype representations. In CVPR, 2008.
16. K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas,F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affineregion detectors. IJCV, 65(1–2):43–72, 2005.
17. D. Lowe. Object recognition from local scale-invariant features. InICCV, 1999.
18. H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. SURF: Speeded uprobust features. CVIU, 110(3):346–359, 2008.
19. Xiao-Lei(Celina) Xia and Kang Li,"A Sparse Multi-class Least-Squares Support VectorMachine'', IEEE 2008.
20. H. Lin and A. N. Venetsanopoulos,"A Weighted Minimum Distance Classifier for Pattern Recognition'', IEEE 1993.
21. Dar-Shyang Lee and Sargur N. Srihari,"A Theory of Classifier Combination: The Neural Network Approach'',IEEE 1995.

22. Padraig Cunningham and Sarah Jane De lany, “k -Nearest Neighbor’s Classifier’’,Technical Report UCD-CSI-2007-4 March 27, 2007

23. Lichun Zhang, Junwei Chen, Yue Lu, and Patrick Wang, “ Face Recognition Using Scale Invariant Feature Transform and Support Vector Machine’’, Department of Computer Science & Technology East China Normal University, College of Computer and Information Science, Northeastern University Boston, MA 02115, USA

24. R. Graves, The Greek Myths: Complete Edition, Penguin, 1993.

25. A. Andreopoulos, J. K. Tsotsos, A Theory of Active Object Localization, in: Proc. IEEE Int. Conf. on Computer Vision, 2009

26. L. Valiant, A theory of the learnable, Communications of the ACM 27 (11) (1984) 1134–1142.

27. L. Valiant, Learning disjunctions of conjunctions, in: Proc. 9th International Joint Conference on Artificial Intelligence, 1985.

28. S. Dickinson, D. Wilkes, J. Tsotsos, A Computational Model of View Degeneracy, IEEE Trans. Patt. Anal. Mach. Intell. 21 (8) (1999) 673–689.

29.E. Dickmanns, Dynamic Vision for Perception and Control of Motion, Springer-Verlag, London, 2007.

30.S. J. Dickinson, A. Leonardis, B. Schiele, M. J. Tarr (Eds.), Object Categorization: Computer and Human Vision Perspectives, Cambridge University Press, 2009.

31. A. Pinz, Object Categorization, Foundations and Trends in Computer Graphics and Vision 1 (4)..

32.M. Tarr, Q. Vuong, Steven’s Handbook of Experimental Psychology (3rd ed.), Vol. 1: Sensation and Perception, chap. Visual object recognition,John Wiley & Sons, 287–314, 2002.]

33. M. Zerroug, R. Nevatia, Three-dimensional descriptions based on the analysis of the invariant and quasi-invariant properties of some curvedaxisgeneralized cylinders, IEEE Transactions on pattern Analysis and Machine Intelligence 18 (3) (1996) 237–

34.J. Tsotsos, The Encyclopedia of Artificial Intelligence, chap. Image Understanding, John Wiley and Sons, 641–663, 1992.

34. S. Dickinson, What is Cognitive Science?, chap. Object Representation and Recognition, Basil Blackwell publishers, 172–207, 1999.

35. D. Marr, Vision: A Computational Investigation into the Human Representation and Processing of Visual Information, W. H. Freeman and
Company, 1982.

36. A. Andreopoulos, S. Hasler, H.Wersing, H. Janssen, J. K. Tsotsos, E. K¨orner, Active 3D Object Localization using a humanoid robot, IEEE
Transactions on Robotics 27 (1) (2011) 47–64.

37. P. Perona, Object Categorization: Computer and Human Perspectives, chap. Visual Recognition circa 2008, Cambridge University Press,
55–68, 2009

Appendix A

Abbreviations

1) ANN - Artificial Neural Network

2) HMM - Hidden Markov Model

3) KNN - K nearest Neighbor

4) CCD- Charge coupled device

5) RGB - Red Green Blue

6) SVM - Support Vector Machine

7)DOG- Difference of Gaussian

Appendix B

An Introduction to Image Processing in Matlab

MATLAB is a high performance language for technical computing. It integrates computation, visualization and programming in an easy-to-use environment where problems and solutions are expresses in familiar mathematical notation. It is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations in a fraction of the time it would take to write a program in a scalar non interactive language such as C, C++ or JAVA.

Image formats supported by Matlab

The following image formats are supported by Matlab:

- BMP
- HDF
- JPEG
- PCX
- TIFF
- XWB

Working formats in Matlab

If an image is stored as a JPEG-image on your disc we first read it into Matlab. However, in order to start working with an image, for example perform a wavelet transform on the image, we must convert it into a different format.

Intensity image (gray scale image)

It represents an image as a matrix where every element has a value corresponding to how bright/dark the pixel at the corresponding position should be colored. There are two ways to represent the number that represents the brightness of the pixel: The double class (or data type). This assigns a floating number ("a number with decimals") between 0 and 1 to each pixel. The value 0 corresponds to black and the value 1 corresponds to white. The other class is called uint8 which assigns an integer between 0 and 255 to represent the brightness of a pixel.

Binary image

This image format also stores an image as a matrix but can only color a pixel black on white (and nothing in between). It assigns a 0 for black and a 1 for white.

Indexed image

This is a practical way of representing color images. An indexed image stores an image as two matrices. The first matrix has the same size as the image and one number for each pixel. The second matrix is called the color map and its size may be different from the image. The numbers in the first matrix is an instruction of what number to use in the color map matrix.

RGB image

This is another format for color images. It represents an image with three matrices of sizes matching the image format. Each matrix corresponds to one of the colors red, green or blue and gives an instruction of how much of each of these colors a certain pixel should use.

Reading Image Files

The command to read an image from file filename and store it in matrix variable p is:

p = imread('filename');

The filename may be an absolute pathname or a relative pathname from the current working directory. Omitting the ';' at the end of the command causes the value to be printed to the command window.

Writing Image Files

The command to write an image from variable p and store it in file filename is:

imwrite (p, 'filename');

The filename may be an absolute pathname or a relative pathname from the current working directory.

Displaying Image Files

The command to display the image from variable p to a figure window is:

Imshow (p);

An additional parameter may be used to set the number of display levels or set the range of display levels. Various controls, such as dynamic display of index and value for the cursor position, are available in the image display tool.

The figure command can be used to create a new current figure for the display:

figure, imshow (p);

Standard Arrays

MATLAB has standard arrays for creating arrays of a defined size with zeros, ones, true, false, or random values. For example: p = zeros(M,N);

Major built in Functions (used in this work) are:

1) Classify

Its format is:

class = classify (sample, training, group)

It classifies each row of the data in sample into one of the groups in training.

Group is a grouping variable for training. „Class" indicates which group each row of SAMPLE has been assigned to, and is of the same type as „group".

2) Factorial Its format is: factorial (n)

This command calculates the factorial on „n" which is the product of all the integers from 1 to n.

3) Bispeci

It calculates the bi spectrum using the indirect method. This command is in „HOSA" toolbox.

Its format is:

[Bspec,waxis] = bispeci (y,nlag,segsamp,overlap,flag,nfft, wind)

Here, „y" is the data vector or time-series, „nlag" is the number of lags to compute, „segsamp" is the number of samples per segment, „overlap" is the percentage of overlap, „flag" stands for „biased' or 'unbiased' , „ nfft" is the FFT length to use, „wind" is the window function to apply.

The output „Bspec" is the estimated bispectrum and it is an nfft x nfft array with origin at the center, and axes pointing down and to the right. „Waxis" is the frequency-domain axis associated with the bispectrum.

4) Size

Its format is:

[M, N] = SIZE(X)

It calculates the size of matrix X and returns the number of rows and columns in X as separate output variables.

5) Sort

Its format is:

Y = SORT(X, DIM, MODE)

This command sorts in ascending or descending order. „Dim" selects a dimension along which to sort. „Mode" selects the direction of the sort, 'ascend' results in ascending order and 'descend' results in descending order. The result is in Y which has the same shape and type as X.