PERCEPTUAL SHAPE FEATURE BASED IMAGE CODING FOR VISUAL CONTENT CLASSIFICATION AND OBJECT RECOGNITION

by

Elham Etemad

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

at

Dalhousie University
Halifax, Nova Scotia
November 2018

Table of Contents

List of Tables

List of Figures

Abstract

The most essential technique in creating agents with ability to process and understand the content of visual data is object recognition, which includes image content classification, and object localization. Deep convolutional neural networks' (CNNs) performance gain in computer vision, there still are application scenarios with limited training data and computing power for which using deep CNN based methods is not feasible. On the other hand, the human engineered image representations require less training data and computing power and can be enhanced by importing domain specific knowledge. These representations may also benefit from the human vision characteristics in reducing the gap between computed image representations and human vision perception. In this thesis we have proposed four methods to improve image classification and object localization. All these methods utilize the perceptual shape features of image since it is proved that the human vision perception on objects mostly relies on shape features of the objects, while color and texture are utilized as extra sources to complete this perception. In the first method, we have created a static dictionary of perceptual shape features based on N-gram model and used that in combination with spatial pyramid matching to represent images. In the second method, a dynamic dictionary from image edge segments is formed where these segments are obtained from an octave of image in different scales. The third method considers the curve partitioning points as descriptive features of the image and created a dynamic dictionary from descriptors of these points. The proposed object localization method utilizes the perceptual shape features of the image to improve the location of objects determined by object recognition module. The initial location may be obtained by any object recognition method, then the proposed method iteratively merges the edge segments with the detected object using a best first search strategy. These proposed methods have been evaluated on different benchmark image datasets. Judging on the overall performance of the proposed method, it is expected that the proposed methods would bring some useful alternatives to support efficient tool development for applications lacking training data or no training data at all.

Chapter 1

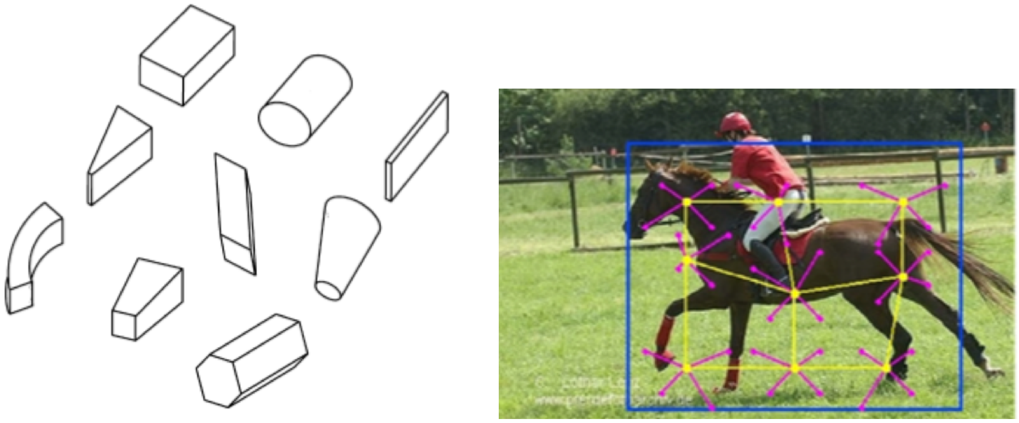
Introduction

Because of the invention of digital cameras, huge repositories of images and videos become available. The production rate of these multimedia data increased substantially by embedding digital cameras into the cellphones. A huge part from this amount of data is now accessible for everyone through the internet. Since an image is worth more than a thousand words, it is a vast amount of hidden information for experts in computer vision and image processing domains to put much efforts towards its extraction and representation. Much research has been done in computer vision to create agents that extract information from image and video contents while the most essential technique in creating this kind of agent is object recognition [7].

1.1 Background

Investigation in the literature of computer vision shows that there is ambiguty and confusion in defining subtasks of computer vision. Terms such as detection, recognition, localization and understanding are generally defined in various ways which creates the impression that there is no universal agreement in their definitions and usages [7]. To be clear about these terms, the definition of Andreopoulos is used in this thesis [6]:

- **Detection**: Whether a single object exists in the image or not?
- **Localization**: Follow detection by finding the accurate location of the detected object in the image.
- **Recognition**: Localization of all the objects present in the image.
- **Understanding**: Recognition of all the objects plus finding the role of each object in the context of the image



(a) (b)

Figure 1.1: Definition of object in extremes of object definition spectrum. (a) Object's feature is defined as a set of templates [20]. (b) Object is defined in the context of the image [157].

In this definition, localization consists of finding the location of the object, x and y position in the image's coordinate system, in addition to detecting the object in the image. Recognition generalizes the localization task to all the presented objects. The understanding task consists of recognition plus the ability to find the role of each object in the context of the image [7].

Another ambiguous and confusing term in object recognition module is the definition of the object by itself, since it depends on the task; detection, localization, recognition, and understanding; that we are dealing with [7].

Considering the definition of [8], when we are dealing with simpler tasks, the definition of the object is closer to a set of templates that define the features of object in different conditions and viewpoints. On the other hand, when the problem turns out to be more abstract, the object is defined by the contextual knowledge and is less dependent of the existence of a set of features in the image. Two different impressions of objects in the extremes of this spectrum are shown in Figure 1.1.

In this thesis, an object is defined as a region in the image whose visual characteristics are already learned by the computer considering both the context and visual features of the training instances (Figure 1.1.(b)). In this definition, the object recognition does not solely rely on visual features of object which are helpful for recognizing

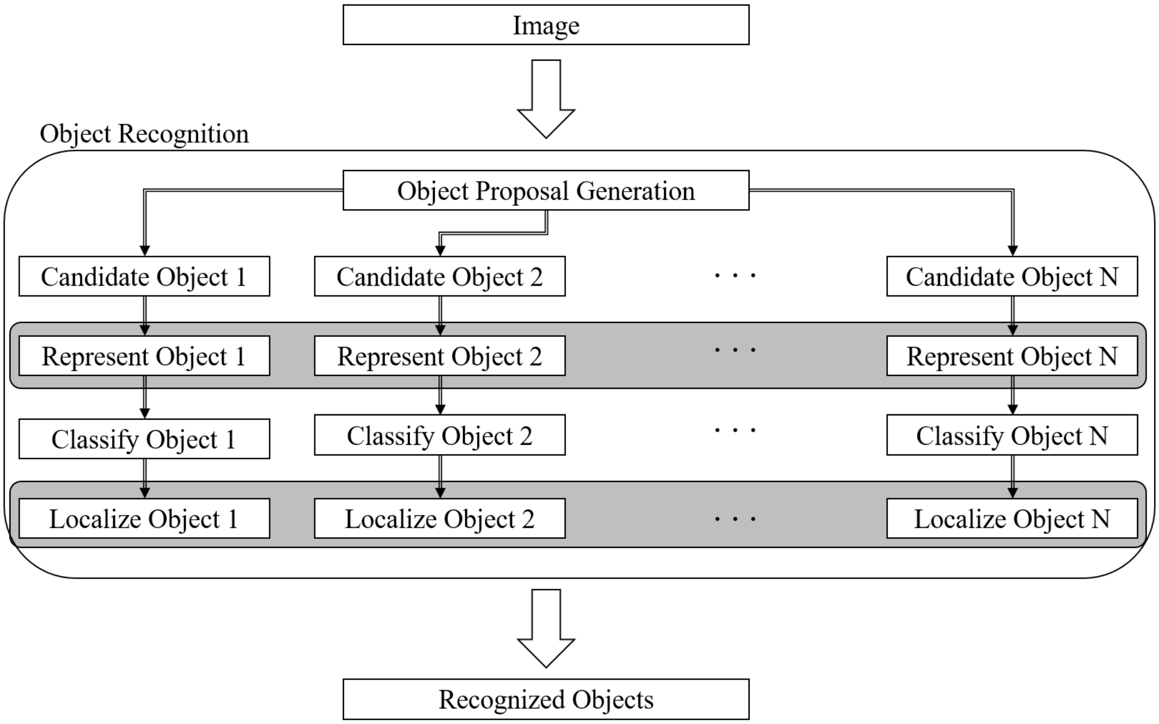


Figure 1.2: The main pipeline for many object recognition methods

occluded objects.

1.2 Problem Statement

The object recognition module consists of three major sub-tasks of object proposal generation, object detection, and object localization [8]. A review on the techniques in each sub-task is represented in the Chapter 2.

In object proposal generation, the candidate regions in the image are found as possible locations that an object may exist. These techniques mostly rely on visual characteristics of the image such as its color [137], texture [150], edge segments [150] and so on. After finding the candidate locations, the part of the image in each candidate area is represented and classified to detect the correct objects among those candidate regions and reject the regions with no object. Object detection, similar to many other vision-based applications such as Image classification, and vision-based Augmented Reality (AR) applications, relies on having an efficient representation for the image. A more descriptive representation results in an improved computer vision based application. There are many different ways of representing an image; locally, globally or learning the representation.

Although the deep image representation methods using Convolutional Neural Networks (CNN) have performed similarly to humans [134], they require a huge amount of training data, access to pre-trained models in cases with few training examples, and powerful computing devices. There are many cases, such as Augmented Reality applications, where any or all these requirements are not feasible. On the other hand, global and local methods require less training data, and computing power besides having more applicability of the prior knowledge, obtained from the current application, in the representation. As an example, mobile Augmented Reality applications have small application-specific training data and limited computing power. These applications mostly rely on handcrafted features for image representation [66].

When the objects in the image are detected, a localization module is applied to better estimate their location in the image. The localization module may use visual information of the image such as color segments [30], or be a trained regression from the ground truth area in the training set [59].

1.3 Research Contributions

In this research we are focusing on the object representation, and object localization tasks which are highlighted in Figure 1.2 to improve the quality of object recognition methods. In order to represent the image, we proposed three image representation methods using the perceptual characteristics of the human vision system.

Based on the fact that human vision perception mostly relies on the object's shape, the boundaries of objects are more descriptive characteristics of the image. On the other hand, according to the Gestalt Laws of grouping, the human vision perception tends to group objects that are close (*proximity*) or similar (*similarity*) to each other and are continuing one another (*continuity*) [28]. These characteristics of the human vision system have inspired us to propose perceptual image representation methods which improve the performance of computer vision applications.

First Image Representation Method: This method describes the image by extracting their Generic Edge Tokens (GETs) and describing traces in the image using the N-gram notation where its visual words are extracted GETs. In this method a static dictionary of visual words is created in which the words are the N-grams of the image. The Bag of Words technique has been used for describing the image using this statically-generated dictionary. By introducing the Shape Pyramid made of the flat dictionaries of the N-grams, we created a hierarchical dictionary for our bag of visual words. For further improvement, we applied a hierarchical structure on selection of the local patches of the image by using the Spatial Pyramid structure. By combining these two pyramids, we came up with a Spatio-Shape Pyramid structure in which each level of the Shape pyramid is associated with a specific level in the Spatial pyramid. The experimental results show performance of this image representation on some benchmark datasets. The detailed discussion on this proposed image representation method is presented in Chapter 3 and [46, 45].

Second Image Representation Method: This method extracts perceptual structure-based edge segments from the image's edge map, describes the region around them, and clusters those segments to find edge tokens. Each image will be encoded using these edge tokens learned from the training set. In this method, we have considered octaves of images, versions of the original image smoothed with various values for the smoothing parameter, and applied different smoothing filters to each of them to extract edge maps. This smoothing using different parameters creates a hierarchy of edge segments, in which the edges obtained from the most smoothed image are coarser and less noisy and representative of objects' boundaries, and edges obtained from the least smoothed images are finer and represent smaller objects and textures in the image. We utilize the Canny Edge detection algorithm [22] along with the Hough transform [54] to find edge segments of the image in each level of smoothing, and each octave. The feature vectors for these segments are created by applying a local descriptor to the area around them. These feature vectors are clustered using K-means algorithm to find edge tokens from the training set. These tokens are utilized by the proposed method to find an encoding for each image. Our proposed method has been tested on the multi-class multi-label image classification problem and its performance comparison is elaborated in the experimental results section. For more information on this proposed method, please review Chapter 4 and [44].

Third Image Representation Method: The N-gram based image representation method utilizes PCPG [72] which is an edge tracker module based on the human vision perception to extract N-grams of the image. This method improves PCPG by generalizing the joint detection module and applying the laws of Gestalt to group perceptual structure-based edge segments. In this method, we have considered edge's direction changes happening as a result of sign or magnitude change in the edge's pixels' gradient to identify the Generic Edge Tokens (GETs). We have grouped these GETs based on their proximities, and their slope and curvature similarities, while preserving the continuity of the edge traces and found higher level Curve Partitioning Points (CPPs) which are utilized as descriptive points for the image. These CPPs are described and clustered to create a Bag of CPPs (BoC) which contains the representatives for different groups of similar CPPs in our training set. Each image is encoded according to this BoC by calculating its Normalized Curve Histogram in all levels of the Spatial Pyramid Matching [92]. The detail of this proposed method is discussed in Chapter 5 and [42].

Object Localization Method: We proposed an object localization method by relying on the fact that each object has boundaries which can be captured by the edge tracker algorithms. We have used the PCPG [56] edge tracking method in order to find the required edge map. Then we applied a Best-First search [122] among the obtained edge segments and optimized the output score of the deep convolutional neural network for each recognized object. This score is the confidence returned by SVM when the representation of the cropped image obtained by CNN is fed for classification. In our proposed method we tried different sets of edge segments from the edge map, calling Trace, Generic Edge Tokens (GETs) and their combination. The AlexNet model [90] has been used for conducting our experiments, although our method is independent of the underlying convolutional network as far as they generate scores for each input image to be in each class. The RCNN object detection module [59] are used as the base method for object detection, that we applied our model for improving their localization. The proposed method is also independent of the object detection module. You may find a complete illustration of this proposed method in Chapter 6 and [43, 41].

1.4 Thesis Organization

This report is organized in the following order: a brief review of the current research in object recognition is presented in Chapter 2. The proposed methods are discussed and elaborated in Chapters 3, 4, 5, and 6. This thesis is concluded in Chapter 7 along with some possible areas of future work and the limitations of the proposed method.

Chapter 2

Background

Nowadays, due to availability of digital cameras for everyone and everywhere, visual data has a major share of digital content which is increasing exponentially. This amount of data creates the need to process and understand the contents in this visual digital world to facilitate humans' life. Computer Vision is the procedure of finding what exists in an image and where is it located [103]. Since 1960s, much research has been done in computer vision to create agents that extract information from image and video contents. The most essential task in creating this kind of agent is object recognition [7] which relies on the quality of the utilized image or video representation. In this chapter, we provide a brief survey on the existing image representation methods in Section 2.1 and complete the review by covering techniques to improve object recognition in Section 2.2. In the end of this chapter, we have provided a brief discussion on some of the techniques and concepts we have utilized in our proposed methods in Section 2.3. The organization of this chapter is depicted in Figure 2.1.

2.1 Image Representation

Each image consists of a number of pixels with wide range of intensity values or colors depending on the image color scale. The very first representation of the image is its pixel map which describes the image by a set of numbers, one for each pixel. However, this representation by itself is not suitable for complex computer vision tasks (e.g. object recognition), since the intensity values of the pixels are susceptible to environmental changes and the changes in the position of the objects in the scene; it is the base for most of the existing image representation methods which describe the image based on the texture, shape and color obtained from its pixel map.

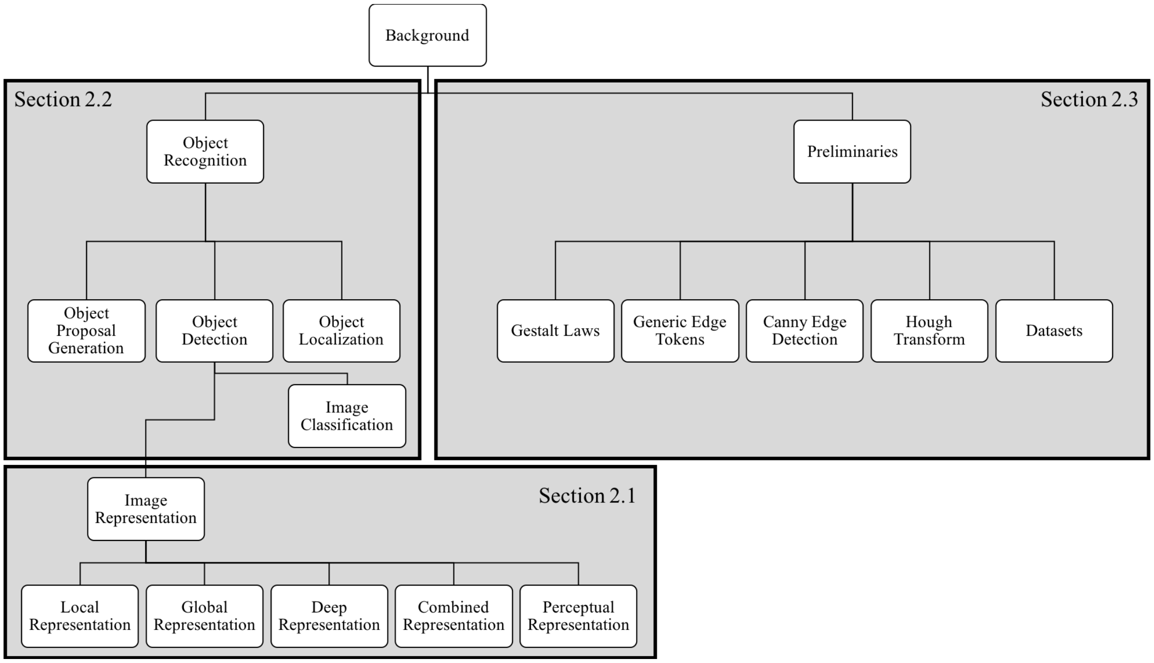


Figure 2.1: General organisation of the literature review chapter.

There are many different methods for representing images which can be categorized into four groups of local image representation (2.1.1), global image representation (2.1.2), deep image representation (2.1.4), and combined methods (2.1.5). The last category utilizes a combination of the other three for representing images. Besides, there are many image representation methods that consider the perceptual characteristics of the human vision (2.1.3). In this research our focus is on combining perceptual characteristics of the human vision and the local image representation techniques to create a more accurate representation. This representation can also be used for bridging the semantic gap of image representation between human perception and program interpretation.

2.1.1 Image Representation Using Local Features

The image representation methods using local features extract several interest points in the image and produce descriptors for those interest points which we call keypoints in the following context. These descriptors are used by the Bag of Feature (BoF) technique to represent an image. The general processing pipeline for BoF is represented in Figure 2.2. The major processes in BoF technique are Keypoint Detection, Local

Region Description, and Feature Association [9]. In the following sub sections, we are going to survey popular methods in each and every step of this pipeline.

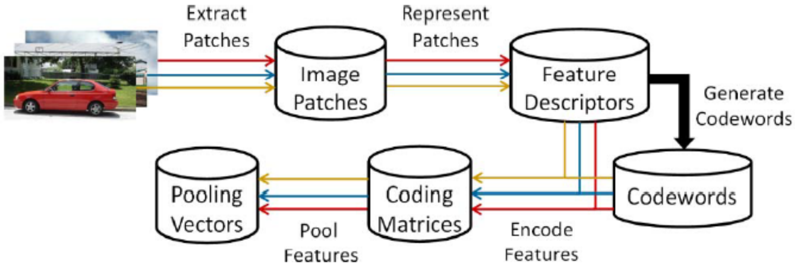


Figure 2.2: General diagram of Bag of Features [74].

Keypoint Detection

There are several methods for detecting Keypoints which are mostly corners of objects in the image. One group of the methods finds edges of the image, then finds corners by tracing edges. Another group finds changes in direction for places with a larger gradient. The other group tests small patches of the image to see if it is a corner candidate or not [119].

The extracted keypoints must have special characteristics to be considered as reliable keypoints. They have to be easy to extract and robust to rotation, scaling, change in illumination and viewing direction. These keypoints have to be repeatable, distinctive and robust to noise as well [10]. Various methods are investigated and introduced for extracting keypoints of images, some of which are discussed briefly.

In 2004, Lowe proposed a novel keypoint detection method called SIFT [101]. They used the Scale Space Pyramid [147] technique to extract keypoints. They calculate Difference-of-Gaussian (DoG) by using adjacent image scales and extract points which are local maxima between three scales and nine neighbors in each scale. They fit a quadratic function to the extracted points and solve it to localize keypoints. They also find orientation histograms for candidate keypoints and assign orientations to them.

SIFT is invariant to scale and rotation while it is robust against noise, illumination change and change in viewpoint [101]. However, this method requires heavy computation and is not suitable for real-time applications running on the current mobile devices [119]. To improve the SIFT algorithm, Ke proposed PCA-SIFT method later in 2004 [85]. In this method, they extract a huge number of patches from a diverse collection of images. They project the patches gradient to vector using their Eigen space matrix. By applying PCA [83] to the covariance matrix of these vectors, the size of the feature vector is reduced. By this technique, they improved SIFT execution time for the matching phase, while the representation phase became slower than the original SIFT [85].

FAST (Feature from Accelerated Segment Test) keypoint detector was introduced by Rosten et al. [119]. This method extracts corners from a set of images by applying Segment Test criterion. Then they create a Decision Tree to classify corners. For extracting corners, they consider a circle area around candidate corners and examine the number of pixels in that area to confirm if it is a corner or not. This method is great for real-time applications [96], although its performance for large-scale features is weak [119].

Speeded-Up Robust Feature (SURF) utilizes the Scale Space Pyramid for its keypoint extraction, but its innovation replaces image resizing with image smoothing to create the pyramid [10]. They apply non-maximum suppression to three scales and nine neighbors in each scale and extract the maxima as a keypoint candidate. They assign orientation to these keypoints by calculating Haar-Wavelet response of the candidate keypoint in both directions.

Later in 2008, an improvement on SURF was proposed by applying some modifications on Interest-point interpolation and Orientation Estimation to solve issues of SURF such as computation time and the accuracy [9].

ORB is an enhanced combination of the Fast Keypoint extraction method with the BRIEF keypoint descriptor [120]. In this method, they applied the Scale Space Pyramid for their keypoint detection method while they find FAST keypoints in each level of the pyramid as candidate keypoints. Later on these keypoints are sorted according to their Harris Corner Measure [67], and a predefined number of them are selected based on this measurement.

BRISK, introduced in 2011 by Leutenegger, is a feature detector and descriptor [96]. It uses the Scale Space Pyramid for keypoint detection. It uses the FAST method to select candidate keypoints. They use the FAST score for keypoint neighbors in current scale and also in a scale above and a scale below that. Finally, they extract refined saliency maxima for each candidate keypoint.

Feature Description

When keypoints have been extracted, the patches around those keypoints should be described in a stable and compact way. This representation should be robust to scale, rotation, affine transforms and noise [4]. Several studies have shown that the accuracy of the image representation mostly depends on the feature description method rather than keypoint extraction [85, 120]. On the one hand, these descriptors have to be distinctive, concise and robust to changes in the viewing condition and errors of keypoint detectors [85]. On the other hand, having a local descriptor which is fast to compute and memory-efficient to use is a critical requirement [21].

The SIFT descriptor selects a patch around the keypoint, then it rotates the patch using keypoint orientation information. It applies a Gaussian function to weigh gradient magnitude around the keypoint. These values are calculated on an orientation histogram around the keypoint to create the feature descriptor [101]. The dimension of the feature vector is a drawback of the SIFT method since it results in high computation time and storage space demand [10, 21]. Later on PCA-SIFT was introduced to solve problems with SIFT. It projects gradient images by using Eigen space which helps it to describe the patch with a more compact feature vector [85].

SURF, introduced in 2006, is much faster than SIFT but its representation still requires 256-bin-dimensional vector for each descriptor which is high when the number of keypoints increases [4, 21]. SURF selects a patch around the keypoint and rotates it to the orientation of the keypoint. It divides the patch to a number of cells and for each cell, it calculates Haar-Wavelet responses in both x and y directions [10].

CHoG, which is 10 times faster than SIFT, devides the patch into localized cells and local image gradients are computed for each cell [26]. This method applies Vector Quantization to encode gradient distribution to a small set of bins. Feature descriptor is created from these bins all over the patch.

BRIEF descriptor, however, utilizes a different method for feature description where a set of tests are scattered through the keypoint's patch. The descriptor vector is created based on the results of these tests on the patch. Each test compares the intensity value of two points and its result is either 1 or 0 based on the result of comparison [21]. This method is efficient for computation and storing in the memory, and as a result it is a satisfactory method for real-time applications [21, 96]. However, BRIEF is not reliable and robust to distortions and transformations [96].

ORB which is a modification of BRIEF descriptor was proposed in 2011 [120]. It uses the same method as BRIEF but on a steered version of the test matrix. This steered version is obtained by rotating the matrix according to orientation of the keypoint which is extracted using Oriented version of FAST keypoint detector. This method is robust to rotation and noise, while it is efficient in terms of time [4].

Continuing the trend that was started by BRIEF, a method called BRISK is proposed in which they select a number of test points in a circle around the keypoint [96]. These points are classified to short-distance and long-distance pairs. By using long-distance pairs, the direction pattern of the keypoint is extracted. The feature vector is created by comparing intensities of pairs in the short-distance category.

FREAK uses a retina sampling grid [111] to describe the keypoint area. In this model, they considered higher number of points near the keypoints while the density of points in farther area is less. The feature vector is a binary string that is formed by one-bit Difference of Gaussian (DoG) between points in the area around the keypoint [4]. This method is faster in terms of computation and more efficient in terms of memory space. It is also more robust to changes compared with SIFT, SURF, and BRISK [4].

Image Encoding

Refering to the pipeline of BoF framework (Figure 2.2), there are five major approaches for finding a global representation for the image by using its local features, keypoint detection, feature description, dictionary generation, feature encoding and feature pooling [74]. The first two approaches are discussed in the previous sub sections, and in this section the focus is on the latter three for finding the global representation.

In the dictionary generation phase, a dictionary created from the local features will be generated. There are many ways for creating dictionary from the local features. Many methods use a clustering algorithm such as K-means on the local features from the training set. The centroids of the obtained clusters are the words for the dictionary. While various dictionary learning methods have been proposed, the experiments conducted by [118, 34] demonstrate that the dictionary learning method has less significance on the performance of the global image representation. As a result of this observation, in this section we will focus only on the feature encoding methods which are categorized into reconstruction-based methods [74].

The main approach in feature encoding is solving an optimization problem of Equation 2.1. The solution for this equation is a vector of coefficients that minimizes the reconstruction error while some constraints are applied. In this equation, $B = [b_1, b_2, ..., b_m] \in \Re^{d \times m}$ is the dictionary, $X = [x_1, x_2, ..., x_d] \in \Re^d$ is the local feature, and $C = [c_1, c_2, ..., c_m] \in \Re^m$ is the coefficient which is the solution of this optimization problem. In fact, by minimizing the reconstruction error, we are looking for coefficients that map the local features to the words in the dictionary with the minimum loss. This optimization problem can be solved using a wide variety of methods such as [19, 149, 153, 124].

$$C = arg \min_{C} \|X - BC\|_2^2 \quad s.t. \quad Constraints. \tag{2.1}$$

Most of the time there are two phases in solving the previously defined optimization problem. In the first phase, a prebuilt dictionary exists which, in most cases, is obtained by applying a clustering algorithm on the local features of the images in the entire training set. By assuming that this dictionary is fixed, the optimization algorithm will find the coefficients which are able to map the local features to the current dictionary properly [142].

In the next phase, the assumption is that the trained coefficients obtained from the first phase are good enough for encoding the local descriptors, and the focus is on finding a better dictionary by solving the above mentioned optimization problem this time by targetting the dictionary [142] (see Equation 2.2).

$$B = arg \min_{B} \|X - BC\|_2^2 \quad s.t. \quad Constraints \tag{2.2}$$

Considering the optimization problem for minimizing the reconstruction error, researchers applied different constraints on this problem and proposed a wide variety of encoding techniques some of which we summarize.

A very basic encoding method is the Vector Quantization (VQ) or hard coding technique. In this method, the optimization constraint forces to have only one non-zero segment in the coefficient matrix for the current input. Looking at Figure 2.3 for VQ, each input x is mapped to a single word in the codebook. The coefficient in this case is the weight associated to this single connection. The VQ method suffers from quantization loss and ignores the relationship between different words in the dictionary by encoding each descriptor to a single word in the dictionary [61].

The sparse coding (SC) technique is introduced by adding the sparsity regularization term to the optimization constraints [95]. By this term, the quantization error of VQ is reduced and the salient patterns of the local descriptor is taken into account as well. This term helps the optimization problem to have only one unique solution. In this method the regularization term is not smooth and it loses the correlation between words of the dictionary. Each input descriptor x is encoded using some words in the codebook, usually more than one word, and these words are selected sparsely for each input (Figure 2.3).

The label constrained sparse coding method is similar to the sparse coding technique except that it takes the value of labels for each class into account. This method uses label consistency constraints in combination with other constraints of sparse coding. In this method, they create a visual similarity matrix for visual words and compute the label similarity of local features based on that [98].

By adding the distance of the local features with words in the dictionary, the Locality Constraint Linear Coding (LLC) is introduced [144]. By applying this constraint on the optimization problem, each descriptor is represented by multiple bases accurately. This method considers the correlation between similar descriptors and ensures that similar patches will be encoded with similar words. In the LLC representation when two input descriptors are similar to each other, there are more common words in the codebook for their encoding, and these words are also similar to each other (Figure 2.3).

While the discussed methods consider encoding a single local descriptor with a set of words in the dictionary, there are some methods which consider encoding a

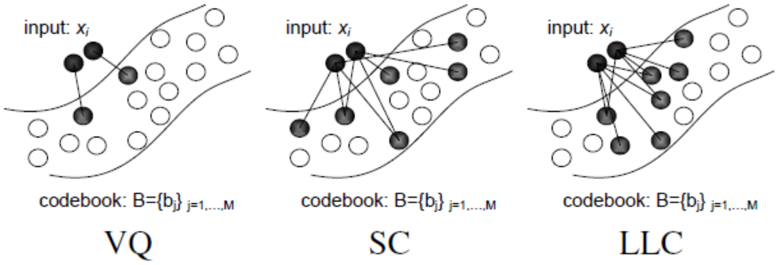

input: x_i
input: x_i
input: x_i
codebook: B={b_j} j=1,...,M
codebook: B={b_j} j=1,...,M
codebook: B={b_j} j=1,...,M
VQ
SC
LLC

Figure 2.3: Schematic comparison between three different coding methods [144].

group of words with a set of words in the dictionary, called group sparse coding. In this method, they regularized the trade-off between reconstruction error and suitable mixed norm for reconstruction weights. By having a set of training groups, a good dictionary estimation is done in this method [12].

The group sparse coding method works for the situations where we have knowledge about the existing groups in the local features. If this information is not available, the automatic group sparse coding can be used. The goal of this method is finding the hidden groups of data and training a dictionary over different groups. This method quantizes the data space by using different dictionaries. It minimizes the quantization error of a sample with a dictionary and finds the dictionary whose produced reconstruction error is minimum [142].

The Spatial Pyramid Matching technique is proposed to add a hierarchical structure to the encoding methods and has resulted in huge performance gain [92]. This method divides the image into gradually smaller non-overlapping blocks and creates the encoding which is a concatenation of the representation for each of these blocks. The size of the blocks for each level is smaller than the previous level's blocks' size. The spatial pyramid matching technique is combined with other encoding methods [152], and deep learning image representation networks [68] and has shown performance improvement on the benchmark datasets.

2.1.2 Image Representation Using Global Features

The global image representation methods describe an image as a whole by considering different characteristics of the image such as its color, texture, shape, or perceptual features of human vision. In this section, we are going to briefly summarize efforts which describe an image globally.

The use of color and texture features of the image as its representation is a common practice for global representation method in which the color feature can be defined based on the color space of the image [130]. The most common color spaces are RGB and HSI color spaces [57]. In terms of texture there is no exact definition for it, but the IEEE institute defined it as *"an attribute representing the spatial arrangement of the gray levels of the pixels in a region or image"* [109].

To capture the color feature, they used the color histogram of the image in HSI color space. Color histograms represent the distribution of colors in an image. Each histogram has a set of bins each of which corresponds to a color in the color space. The value of each bin is the total number of pixels in the image whose color is the color of the bin. To extract the texture of the image, they applied wavelet transform to the image. The wavelet transform decomposes image to its frequency bands [130].

The orientation of the edges are detected by using the color space of the image where the color difference histogram is utilized to find the edge orientation of the image [97]. The idea behind this technique is a psychological fact that human vision is sensitive to the color and edge orientation [84, 99].

An image representation method based on the fusion of color and texture features is introduced in which they have selected a number of colors from HSV color space as an image representation. They also find the co-occurrence matrix of the gray-scale image in four directions and calculate contrast, capacity, entropy and relevance of an image as its texture features [154].

An image representation by using three major visual characteristics of the image named color, texture and shape is introduced by Iqbal et al. [77]. To represent the color of the image, they create three color histograms for different channels of RGB as well as an intensity histogram based on the gray-scale image. To extract the texture information of the image, they applied the Gabor wavelet algorithm. They applied Hu moment invariant algorithm [73] to find shapes of the image [115]. They concatenate these feature vectors to create the global image representation.

Another method that describes the image by considering its color, texture and shape features is proposed by Wang et al. [145]. To find the color feature of the image, they used RGB color space and quantized each channel to a number of colors.

They apply an optimization problem to find which pixel of the image belongs to which quantized color. To represent the color information, they used the fuzzy color histogram. They applied the steerable filter to the image to find representation for the texture of the image [79]. The Pseudo-Zernike moments is used to describe the shape of the image in this method [86].

An image representation was introduced in 2015 to classify apple disease [39] which considers three major factors of color, texture and shape to create the image representation. To describe the color of the image, they used the color histogram along with the color coherence vector [114]. Color coherence is a kind of color histogram whose segments are super pixels (which create coherent areas) instead of pixels. In this method, they applied Local Binary Patterns (LBP) [110] and complete LBP [65] to extract texture information of the image. As the last part, they used Zernike moments of the image as its shape descriptor.

2.1.3 Image Representation Using Perceptual Features

As a result of studies on human visual system, many researchers in computer vision domain have utilized the perceptual characteristics of the human vision for describing images [75]. In one hand the human visual system is capable of grouping elements from complex scenes to simplify the image description, and on the other hand perceived elements from natural scenes belonging to a single object are often grouped in the human visual system [108].

Various characteristics of the human visual system such as attention system [102], and color [105] or shape [156] characteristics of images are considered by many researchers. Since our proposed method has focused on perceptual shape features, a brief review of some of the existing methods are provided here.

Gestalt Laws is utilized for retrieving the human-made objects where it relies on the fact that these objects usually have solid edges and corners [78]. In this method, strong evidences of existance of an object which are obtained from relationships among edge segments are extracted as image features. These features are classified using K-nearest neighbour algorithm and the images that contain human-made objects are retrieved.

The perceptual shape descriptors are also utilized for image retrieval where various tokens are extracted from each shape each of which corresponds to a salient attribute of that shape [14]. These tokens are described according to its orienation in the image space. These tokens are arranged to create an $M - tree$ structure which is utilized for image indexing.

The perceptual edge segments obtained from an image's edge map are utilized for its representation in the context of image retrieval [156]. They have divided the image into non-overlapping blocks, varied sizes for edge segments and noises. For each block, they have calculated the frequency and length of each edge segments and by concatenating these features, they have described each image and used it for image retrieval.

By importing the N-gram notation from natural language processing into the task of image representation using perceptual features, Mukanova et al. have introduced N-grams of shape by grouping the connected edge segments together and describing each image using those N-grams [107].

2.1.4 Image Representation Using Deep Features

The very first Convolutional Neural Network is introduced in 1998 for classifying handwirtten digits. LeNet with 2 convolution layers each followed with a pooling layer is proposed by Lecun et al. [94] is shown in Figure 2.4. The outcome of the second pooling layer is flattened and connected to a fully connected layer for classification. This network has shown promising result in character recognition, but it could not be applied on larger images with higher resolutions at that time because of the limitations in computing power.

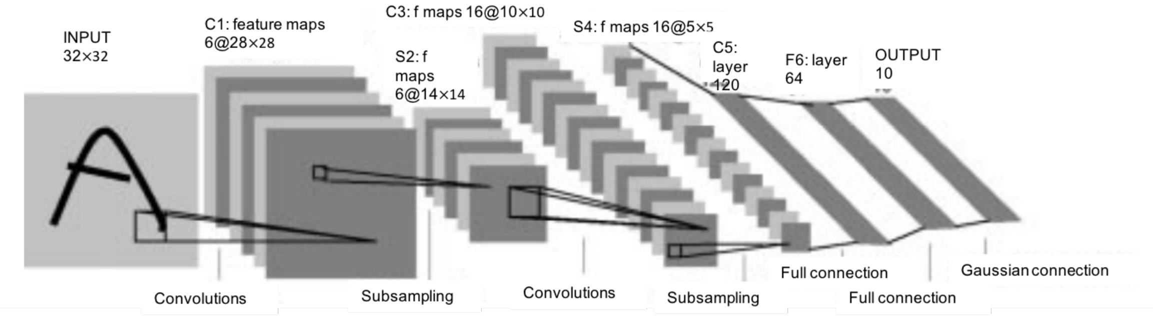


Figure 2.4: The LeNet architecture proposed in [94].

A deep convolutional neural network for classification of images in the huge dataset of Imagenet [38] is introduced in 2012 [90]. In this method they introduced a deep network with eight layers five of which are convolutional layers and the other three are fully connected layers. This network consists of 60 million parameters and 650.000 neurons. Each neuron uses ReLU nonlinearity function which is much faster than the other *tanh* functions [90]. The overall architecture of this network is presented in Figure 2.5.

The convolutional layers of this network convolve the input image with a number of kernels whose sizes are different for each layer. For instance, the first convolutional layer filters the input image of size $224 \times 224 \times 3$ with 96 kernels of size $11 \times 11 \times 3$ [90]. This structure improves the classification result by reducing the error rate of the top-1 test about 10 percent.

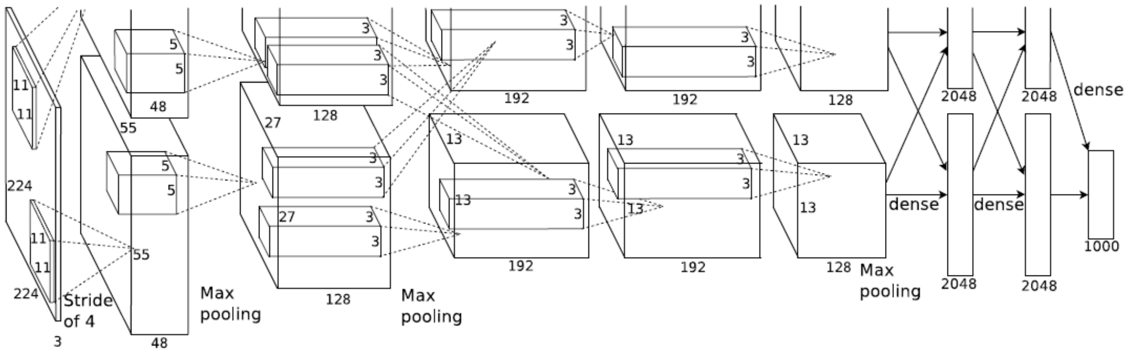
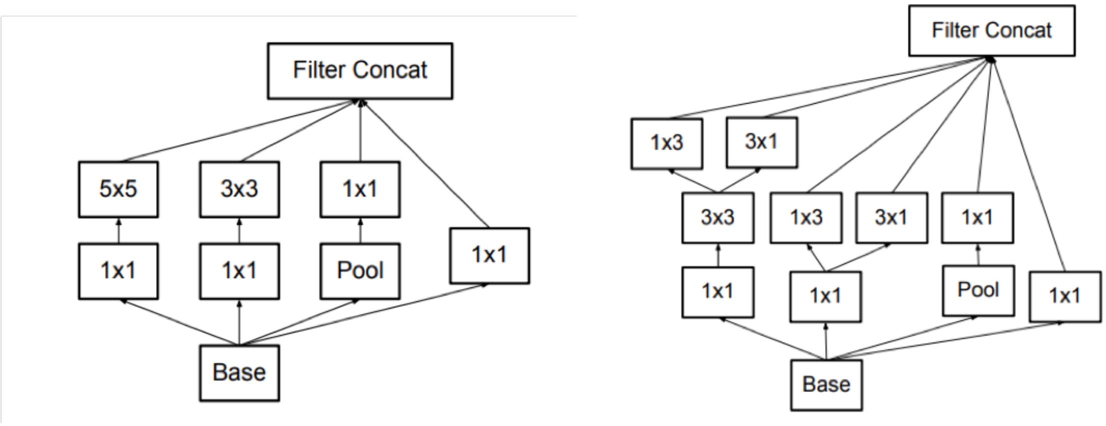


Figure 2.5: The CNN architecture proposed in [90].

The winner of ImageNet competition in 2013 has introduced a modification of the Alexnet network which is called ZF-Net [155]. There are two main contribution in this paper, modification of Alexnet, and visualizing the intermediate features. They have changed the size and number of filters for the convolutional layers which resulted in their superior performance in that competition. They reduced the size of the filter in the first convolutional layer to $7 \times 7 \times 3$ with a decreased stride value, the number of pixels the filter slides over the image. This modification helped them to retain most of the information in the image's pixels. Their proposed visualization for the intermediate features, deconv-net, also opened a way for the researchers to understand the middle layers of the deep network.

GoogleNet [135] introduced by Google has won the ILSVRC competition in 2014 by achieving a performance very close to human. This network introduces the inception modules and utilized batch normalization, and RMSprop. Using inception modules, this model could reduce the number of network parameters drastically. To introduce inception layers, they have utilized Hebbian principles to move from fully connected convolution layers to sparsely connected ones. In this architecture, they utilized 1 × 1 convolutions to reduce dimension and to use rectified linear activations [116]. They have improved this inception modules and introduced modules with coarser filters in [136].



(a)

(b)

Figure 2.6: The Inception modules introduced in (a) [135] and (b) [136].

VGG-Net proposed by researchers of Oxford university in 2014 has 19 layers with the filter size of 3 × 3 and stride of 1 [129]. Their reasoning for using a smaller filter size and a deeper network is the fact that combining two 3 × 3 convolution layers has an effective receptive field of 5 × 5. This is similar to having a larger filter while keeping the benefits of smaller filter sizes. Other benefits are a decreased number of parameters, and using two ReLU layers, one for each convolution layer, instead of one.

Sometimes the source task for training is different from the target task for the test. This situation may occur due to using a pre trained model of deep learning. A major problem in this situation is different labeling of the images in two separate tasks. To solve this issue, the 8th layer of the architecture in Figure 2.5 is replaced with two modified and adopted fully connected layers in the pre trained model and the exact values of the trained parameters in the network are carried over to the testing phase [112].

Using the input feature to the first fully connected layers of the network as the image representation is investigated by Razavian et al. [117]. In their proposed method, they applied the network similar to Figure 2.5 to extract image representation. They proposed two different settings for image classification. In the first one, they simply input the extracted representation to the SVM classifier. In the second setting, they add some cropped and rotated versions of the training images to the training set and classify the test images. Their experiments show that the second setting improves the classification accuracy [117].

ResNet, or Residual Neural Network, utilizes gated recurrent units and heavy batch normalization as layers of the deep network which are shown in Figure 2.6. This network which contains 152 layers has lower complexity than VGGNet and its performance is better than human. Not only these residual networks allow the continuous information flow by using parameterless identity-mapping shortcuts, but also they can have more than thousands layers without losing any performance [70, 69].

2.1.5 Image Representation Using Combined Features

However deep learning representations have achieved promising performances, there are some research that have combined handcrafted local or global features with these machine learned deep features for further improvement on the performance.

Combining the bag of words framework with learning deep features is investigated in 2014 where the proposed architecture borrows the strength of both techniques [60]. In each level of this method SIFT local descriptors [101] are encoded using the spatial aggregating restricted Boltzmann machines (RBM) [131]. They stacked several layers to create a Deep Belief Networks (DBN) [71] to finalize their hybrid architecture.

By combining the handcrafted and deep feature, the performance of object recognition has been improved using three dimensional data [81]. In this method, they extracted the SIFT local descriptors [101] and encoded them using LLC [144]. At the end, they utilized Spatial Pyramid Matching technique [93] to create the handcrafted representation. In Parallel, they have calculated deep representation using the AlexNet convolutional network [90]. These two representations are combined and fed into an SVM classifier for determining the type of the object.

A network with two main parallel representations is devised by Wu et al. [148] which is inspired from [81]. For deep representation, they devised their own CNN network consisting of five convolutional layers. This network generates an image representation with the length of 4096. For finding the handcrafted features, they calculated RGB, HSV, and YCbCr [82] histograms for representing the color feature. They also utilized 8 Gabor filters [55] and 13 Schmid filters [123] to calculate texture features of the image. At the end, they find a representation vector with the length of 4096 from concatenating these local features. They use a fusion layer to combine them and generate their final representation which is fed into a Softmax layer for classification.

2.2 Object Recognition

Object recognition is the task of finding the locations and types of all the objects in an image [6]. This task is a generalization of object detection and object localization for targeting multiple objects instead of a single one. There are many researches that have been conducted to improve the performance of object recognition as a whole, or improve one of its sub-tasks which at the end results in the overall improvement. The main sub-tasks which are involved in the object recognition are object proposal generation, object detection, and object localization.

The goal of object proposal generation is finding the areas in the image where the possibility of object existence is higher than the other areas. This possibility is mostly determined using some heuristics on the visual characteristics of the image. For instance regions with similar colors possibly belong to a single object. Some other method utilize boundary information, texture information, or a combination of them. A review of some of the works in this area is presented in 2.2.1.

After finding the candidate object using the object proposal generation, an object detection technique applies on the area for the candidate object to determine the type of the object there. Researchers have used many different methods such as template matching, knowledge, OBIA (Object Based Image Analysis), and machine learning based methods for detecting the type of the object [31], among which we have chose machine learning based methods. The machine learning based methods consists of two major modules for finding the type of the object which are image representation, and image classification. A survey of the machine learning based methods for object detection is given in 2.2.2.

So far, the type of the objects in the image are determined, but their position in the image is not yet regularized. The next step for object recognition, is finding a better location for the detected objects in the image. This task is called object localization and usually utilizes image characteristics such as color, edge, and texture for improving the performance of localization by solving an optimization problem. A survey of these methods is presented in 2.2.3.

As it was mentioned earlier, there are some methods that consider object recognition as a single problem to target. A review of some of these technique is given in 2.2.4. These techniques generally have modifications in more than a task.

2.2.1 Object Proposal Generation

The very first method for finding object proposals in the image is sliding a window all over the image [5] which is used in many recognition methods such as [36, 63, 139]. A method using sliding window and applying linear SVM on HOG feature of each window to find objects and their parts in the image is proposed in 2010 [52]. Despite their impressive performance, their method is costly because of its exhaustive search all over the image.

To mitigate the cost of finding object proposals, the Selective Search method is proposed in 2013 [137]. This method utilizes Felzenszwalb's method [53] for finding initial regions in the image. A buttom-up grouping is performed on these initial regions where in each step the most similar regions are combined to create a larger region. This grouping continues until there is only one group which contains all the regions of the image. For measuring similarity, a diverse combination of color, texture, size and distance metrics is utilized.

Although the hierarchical grouping of Selective Search [137] has performed well, a hierarchical segmentation for proposing candidate objects is also investigated [141]. They proposed a multi-branch algorithm where each branch classifies image regions using a binary SVM classifier. For each class, the new branches are generated and this routine continues. At the end, the remaining segments are greedily merged to create a single segment. The main contributions are training the binary classifiers for each branch of the tree, and using linear SVM with weighted loss for each branch. These weights are determined using the wrongly classified regions in the sibling of this branch, to balance the weights of positive and negative regions. The classification threshold for each branch is also determined using the previous levels classifiers.

However, having an efficient grouping strategy to combine initial regions of the image is important, the effective distance metric has its significance for finding object proposals. By defining a distance metric that considers the amount of complexity in each group of superpixels, the object proposal generation research is advanced [150]. This method calculates color and texture distance between groups of superpixels by using the color histogram and Gaussian derivatives. The Floyd Warshal algorithm [1] is also used to calculate the distance of superpixels from a connectivity graph with the intuition that superpixels belong to an object are mostly close to each other. The level of complexity between two groups is determined by calculating their minimum and maximum color and texture distance. A small maximum distance means that both of them are similar and the complexity is low. On the other hand, a small minimum means that there are at least two similar superpixels in these groups which means that the complexity is high. By defining a complexity metric, a bottom-up merging algorithm has been utilized to group superpixels.

Instead of having a fixed similarity metric for the regions in the image, Chen et al. [29] have extracted the local regions by using the method presented in [53]. They utilized Fast R-CNN [58] for extracting the features for each of these local regions. Then they designed a Recursive Neural Network (RNN) for grouping these local regions and assigning an objectness score for their corresponding proposals.

2.2.2 Object Detection

After finding a set of candidate objects in the image, the type of each candidate object should be determined. Over the years researchers have proposed many different methods for detecting objects in the image which can be categorized into four main groups of template matching based methods [20, 76, 13, 100], knowledge based methods [133, 25, 132], OBIA based methods, and machine learning based methods [31].

Our focus is on machine learning based methods for detecting the type of the object in an image. These methods contain two major parts of representing the image in a way that computer can understand, and apply image classification on those representations [31]. Since an extensive study on various image representation methods is provided in Section 2.1, the classification techniques utilized in this report are described in this subsection.

Support Vector Machine (SVM) [138] is one of the most famous machine learning algorithm for object detection. The original SVM is a binary classifier which finds a decision line with the maximum margin distance from instances of both classes. There are techniques to make SVM suitable for multi-class classification problems among which the one-versus-rest is the most popular one. In this technique, everytime SVM choses one of the classes as positive and all others as negative and iterate this routine for all classes to find all the decision boundaries. Some of the object detection methods that have utilized SVM as their classifier are [37, 64].

Artificial Neural Networks (ANNs) [91] or specifically Multi Layer Perceptron networks are among the widely used classifiers in object detection domain. ANNs contain three main layers of input, hidden, and output. Each of these are made up from sets of processing nodes which perform linear operation on their input signal, and add some nonelinearity to create their output. Each node (neuron) may be connected to all others in the next layer, or to a subset of them. A schematic diagram of a simple ANN is displayed in Figure 2.7 where X and W are input features and networks weights for the first layer respectively and f is the non-linearity function which can be either of Sigmoid, Tanh or ReLU. In the training phase, each of these connections are weighted with random numbers which will be tuned based on the training set. This classifier is utilized in many object detection methods such as [3, 35, 128].

2.2.3 Object Localization

Finding a set of bounding boxes using one of the object proposal generation methods and trying to improve its localization is a practice in object localization [30]. To capture the localization, they used superpixel tightness as a measure. In their method, at first they aligned the bounding box to be the tightest bounding box around the object. To do so, at first they limited the bounding box to the area which is covered

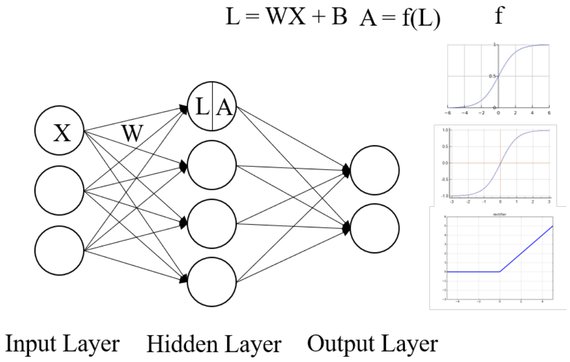


Figure 2.7: Samples of Gestalt psychology laws obtained from: [140, 15]

by all superpixels which are fully laid in the bounding box. Then they add the superpixels that have overlap with the bounding box gradually, until they get to a box close to the current box but tighter. In order to improve their localization, they used straddling expansion with multiple thresholds. They select a set of five thresholds, and they add all the superpixels whose overlaps with the bounding box is greater than each threshold. At the end, they used a non-maximum suppression to come up with a single bounding box per object.

By extracting the candidate objects using the Selective Search algorithm [137], and describing them either by using a combination of SIFT and Fisher Vectors, or a deep representation, a weakly supervised object localization is proposed [33]. For finding the deep representation, they utilized the RCNN approach [59] using the AlexNet[90] network. They proposed an approach for object localization, referred to as Multiple Instance Learning (MIL) approach, which iteratively selects the highest scoring detections as the positive training examples and trains the detection models.

Considering the observation that weakly supervised object localization methods work better images with bigger objects, [127] proposed an object localization module using MIL approach which devide the training images based on the estimated size of the objects. In this method, the object proposal generation module [159] has been used to find bounding boxes in the image. These areas are described using deep learning network of AlexNet [90]. They used the Kernel Ridge Regressor (KRR) [126] to estimate the size of the objects, and divide their training samples to some batches based on the object size. For training on each batch, all of the samples whose objects' size is greater than the threshold for that batch are used for finetuning the trained model obtained from the samples with the bigger objects.

2.2.4 Object Recognition

By adopting three main phases of generating object proposals, representing them, and training a linear SVM classifier for each object class, Girshik et al. have proposed their object recognition method [59]. They extract a predefined number of object proposals from each image using the selective search algorithm and warp them into a bounding box suitable for Caffe [80]. They treat the object proposals that their IoU overlap with a ground truth example is higher than a threshold as positive samples and the others as negative samples. They optimize a linear SVM for each class of the objects using standard hard negative mining method.

After extracting outputs of convolutional layers of CNN, a max-pooling and deconvolution upsampling have been performed to create features with the same size. Then a convolution layer has been applied to these features for capturing more semantics. These features are merged using local response normalization for creating a hyper feature map. This feature map has been used by a convolutional network to find object proposals. They added ROI pooling, followed by a convolution and a fully connected layer into their designed network. The output of this layer, which are bounding boxes and their scores, are fed into another network for object recognition which has a convolutional layer, and two fully connected layers. The final output of their designed network are object bounding boxes and their scores regarding each class [89] .

DeepID-Net has used the selective search algorithm to find some bounding boxes in the image [113]. It has used RCNN to reject boxes which are located in the background of the image and defined and utilized the DeepID-Net network and introduced the def-pooling layers. The output of this network is scores for different object classes. It also got the results of the image classification deep network and used them as a contextual information for refining their scores using linear SVM. At the end, they perform model averaging to improve the performance of their method. The final boxes are fed into a bounding box regression from RCNN to find a better box around the image.

Learning an object detector which can determine the location of the object as well as the type of the object is the aim of much research [17]. For this purpose, they used Selective Search algorithm to find a number of bounding boxes inside each training image. They defined an optimization problem which learns the parameters of the detector, and an optimization problem to predict the location and type of the object in an image. In their objective function, they used CNN features for each box of the image as its feature vector for calculating the similarity of two boxes. For the margin loss, they used the soft-max latent SVM to be able to find boxes for multiple instances of an object, and it makes their objective function less sensible to box initialization.

The latent SVM in combination with the DeCAF features of the CNN network is utilized in order to detect objects in an image [16]. Their key contribution is in applying two constraints on the loss function.The additional constraints are based on two hypotheses: **(a)** If a box contains an object, its horizontally mirrored version also contains the same object. **(b)** It is not possible to have two different objects in a single spot of the image.

In order to do object detection, an energy function is defined with three sub modules that utilize the information in the superpixels of the image [151]. The goal is finding a set of labels for superpixels in the image that minimizes this energy function. **(a)** One module is the appearance of the superpixel which is calculated using RCNN scoring for a number of regions in the image. The score for each superpixel is the summation over the score of all regions that have that superpixel. **(b)** The second module is a smoothness term which is defined using the idea that the neighboring superpixels should have similar labels and in the same time their appearance should be similar. **(c)** The third module optimizes the number of labels in the image, since there is a preferance for having concise and precise labels for each area of the image.

The use of image segmentation information for improving object detection performance is studied by Zhu et al. [158]. They used RCNN in order to get the candidate boxes and their scores. In order to have the segment information of the images, they used CPMC method [24]. They defined an optimization algorithm which considers the appearance of each candidate box, the features of the segment which the candidate box lays in, and the context information of the candidate box. For appearance, they used the same feature vector as RCNN. For segment features, they calculate histograms for the number of segment pixels inside and outside the candidate box, the number of background pixels inside or outside of the candidate box, IoU of the candidate box and the box surrounding the segment, and scores of O2P classifiers for each bounding box and segment [23]. To capture the context information, they enlarged the bounding boxes (Ground truth for training and candidate box for test) by a predefined factor. Then they trained network using these bounding boxes, whose labels are the labels of the box that we already considered. The feature of last fully connected layer of CNN is the context information. They used latent SVM in order to find the best set of candidate segments in images. In order to add negative samples, they applied hard negative mining technique. As a post-processing step, they used the bounding box regression method of RCNN in an iterative fashion. In each iteration, they modify scores based the regressed boxes, if a box has changed more than 20%.

For obtaining coarse sparselets for encoding parts of the objects and learning a dictionary of sparselets, Cheng et al. used a single layer autoencoder [32]. They used these coarse sparselets as initial value for a single layer neural network, where the last layer is a softmax layer. By training this neural network, they find the fine sparselets and the activation vectors. Using these parameters, they can find the response of each feature vector for different part models.

2.3 Preliminaries

So far in this chapter, we have talked about existing methods for image representation and object localization which provided the base for our studies in the following chapters. In this section, a brief review of some of the techniques that are adapted in our proposed methods such as Gestalt laws of grouping (2.3.1), Generic Edge Tokens (2.3.2), Canny edge detection (2.3.3), Hough transform (2.3.4), and the details of the utilized datasets (2.3.5) are provided.

2.3.1 Gestalt Laws

Gestalt psychology school of thoughts has began by the research on the perception of pure motion by Wertheimer et al. [146]. Since then many researches have been performed on various aspects of human visual psychology for perceptual grouping and enriched the Gestalt Laws. Among the vast amount of grouping rules introduced, we focus on the six perceptual grouping rules for contour integration.

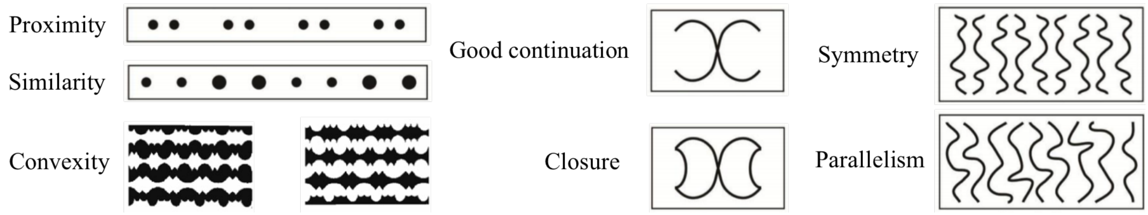


Figure 2.8: Samples of Gestalt psychology laws obtained from: [140, 15]

One of the tasks of human vision is grouping parts of the projected image from an object, which is simplified to contour integration whenever the object's boundary is a closed curve. Transferring to computer vision, this task becomes more difficult because of objects' occlusion and contour loss due to the poor contrast in the image. As a result, the computer will perceive fragmented curves each of which is a candidate for continuing the current contour. Researchers in computer vision have considered oriented edges of the image as primitives to apply grouping laws and find the object's boundary [140]. This is the list of Gestalt laws for contour integration with their definitions:

- **Proximity**: The closer elements to each other are stronger for grouping. The proximity example in Figure 2.8 shows the grouping of dots that are closer to each other.

- **Similarity**: The similar elements are more likely to be grouped together. The similarity example in Figure 2.8 shows that dots with similar size are grouped together.

- **Good continuation**: The elements tend to be grouped to form smoothed contours. In the example for good continuation (Figure 2.8), instead of observing to "c" characters, we observe two lines that are crossing each other.

- **Convexity**: The occluded contours that can be completed to create a convex shape are stronger candidates for grouping comparing to those which may result in concave shapes. In the first convexity example of Figure 2.8, we group the black shapes together, while in the second one which has the same shape with alternate color we group the white ones. This happens because of the convex edges of these areas in these examples.

- **Closure**: This law can be categorized as part of the *good continuation* law, but with determination of the final perception of the elements. We see two moon shapes in the closure example of Figure 2.8 because of their formation of a full shape.

- **Symmetry and parallelism**: While symmetry is considered as a subrule of *good continuation* and *convexity* for creatinng a good shape, parallelism determines the perceptual simplicity of lines. Our vision groups symmetric lines and parallel lines together in the Figure 2.8's examples provided for symmetry and parallelism.

2.3.2 Generic Edge Tokens

Generic Edge Tokens (GETs) are perceptual segments of the image which are extracted using the PCPG (Perceptual Curve Partitioning and Grouping) package [56]. A set of psychological studies inspired the GET extraction procedure which is called Gestalt Laws [140]. These laws describe the human vision system characteristics in understanding the objects and are categorized into six major laws of continuity, symmetry, simplicity, closure, similarity, and proximity.

For extracting the GETs, the gray scale image is scanned horizontally and vertically according to Figure 2.9to find its objects edges. For horizontal scanning, some of the rows in the pixel map of the image are selected to be processed for finding the edge. The number of skipped rows is a parameter of the PCPG package which is adjustable. In each row, the pixels where their values are different from their neighbors are selected as an edge pixel. For each of these pixels, its 8-neighbor pixels are investigated to find the trace which this pixel belongs to. This procedure continues to find all of the pixels in the found trace. The same routine applies to all of the pixels in that row. For vertical scanning, a similar method is used for the selected columns of the pixel map. In the end of this procedure, the edge traces of the objects in the image are found.

These traces are investigated to find the points where the curvature of the edge

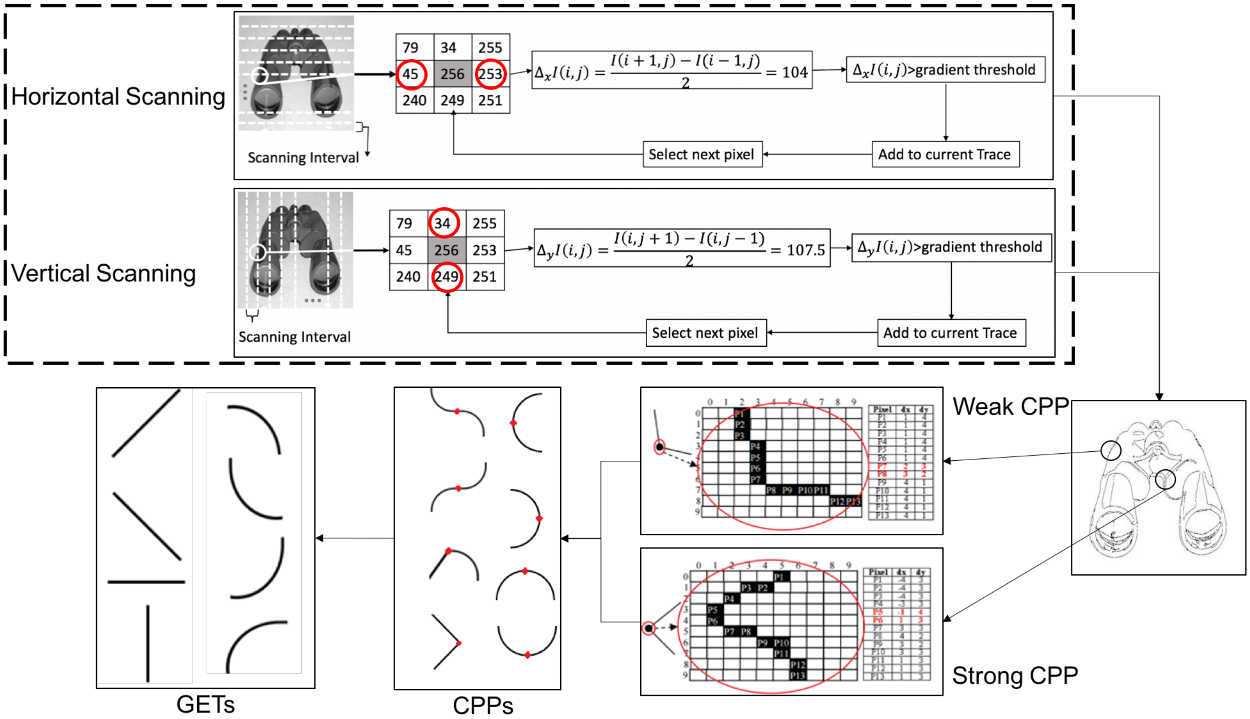


Figure 2.9: The main procedure of PCPG package for finding, the edge map, the Generic Edge Tokens and the Curve Partitioning Points.

changes, Curve Partitioning Points (CPPs). There are two types of CPPs categorized into Strong CPPs and Weak CPPs which are represented in Figure 2.9. The strong CPPs can be found by just comparing the sign of derivatives, dx and dy, for the curve in left side of the point to the curve in the right (i.e. zero-crossing), while for detecting weak CPPs the Order Preserving Arctangent Bin Sequence (OPABS) technique is used [72]. Using OPABS a histogram of arctangent values of derivatives in each pixel is obtained. To create this histogram, the arctangent values are classified into eight categories, each of which corresponds to a bin in histogram. This histogram signifies the evidence for presence of a weak CPP.

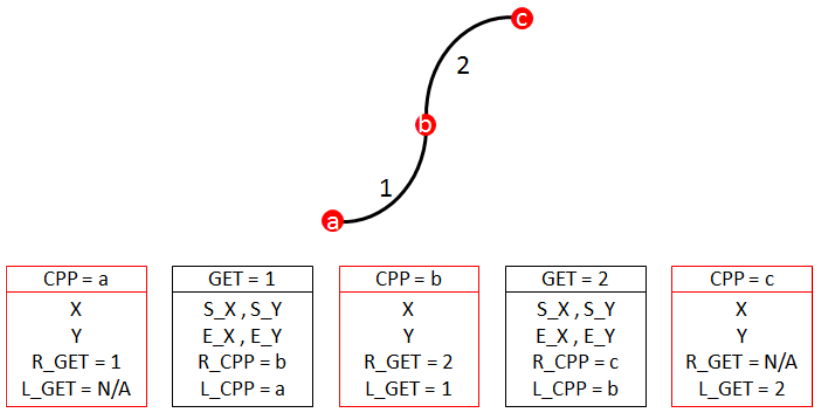
By applying these techniques the CPP points of the image are obtained and categorized into eight groups of Figure 2.9. The first six categories are connections of two curve edges, while one of them is the connection of a curve and a line, and the other one is the connection of two lines.

When the CPPs are found in the image, the curves between CPPs form the set of GETs of the image. These GETs are categorized into eight groups which are represented in Table 2.1. They are discernible based on their curvature and rotation values. There are four groups of curve GETs and four groups of line GETs.

Table 2.1: Definition of 8 groups of GETs based on the properties of Tangent function set ([56]).

GET	Name	$f(x)$	$\varphi(x)$	$\acute{f}(x)$	$\acute{\varphi}(x)$
◞	CS1	$M+$	$M+$	$M+$	$M-$
◟	CS2	$M-$	$M-$	$M+$	$M-$
◜	CS3	$M+$	$M+$	$M-$	$M+$
◝	CS4	$M-$	$M-$	$M-$	$M+$
╲	LS1	$M-$	$M-$	C	c
╱	LS2	$M+$	$M+$	C	c
—	LS3	C	N/A	∞	0
|	LS4	N/A	C	0	∞

The GETs are classified by considering monotonic characteristics of the Tangent Function set of $S = \left\{f(x), \varphi(x), \acute{f}(x), \acute{\varphi}(x)\right\}$. In this set, if $y = f(x)$ is the corresponding function for a curve, the $x = \varphi(y)$ is the inverse of the curvature function and $\acute{f}(x)$ and $\acute{\varphi}(x)$ are the first derivatives of $f(x)$ and $\varphi(x)$ respectively. As an example, $LS1$ is a line with negative slope, $f(x) = (M-)x$, where its inverse function also has a negative slope, $\varphi(x) = (1/(M-))x$, and since they are line their derivations are constant values.



CPP = a	GET = 1	CPP = b	GET = 2	CPP = c
X	S_X , S_Y	X	S_X , S_Y	X
Y	E_X , E_Y	Y	E_X , E_Y	Y
R_GET = 1	R_CPP = b	R_GET = 2	R_CPP = c	R_GET = N/A
L_GET = N/A	L_CPP = a	L_GET = 1	L_CPP = b	L_GET = 2

Figure 2.10: A sample of PCPG data structures for a trace with $length = 2$.

By using the PCPG package, we end up to have data structures for CPPs and GETs. We have an array of CPPs where each segment has the xy-coordination of a CPP point in the image coordination system. We also have the information for the left and right GETs which are connected through that CPP. We also have an array of GETs where each segment has general information about a GET such as curvature, coordination of the start and end points, type, and so on, as well as information about two CPPs in both sides of the GET. A sample of these data structures for a trace with $length = 2$ is represented in Figure 2.10.

As an example, consider the trace in Figure 2.10. Starting from the CPPs without any GET in their left (CPP_a), we trace the edge. In the right side of this CPP, we have GET_1 whose type is $CS1$. The ending point for this GET, is CPP_b. Since we are looking for Bigrams (traces with two GETs), we have to continue tracing the edge (if the length of trace is greater than or equal to two). The GET_2 is located in the right side of the CPP_b whose type is $CS3$ and ends in the CPP_c. For this trace, we have information about the types of its constituent GETs with their lengths, and the coordination of the its ending points.

2.3.3 Canny Edge Detection

Canny is amongst the most popular edge detection methods in computer vision which finds clean edges that are connected to each other [22]. This method starts with a preprocessing step, then it calculates the gradients and forms the edges.

Since extracting edges is prone to noises in capturaing the image, as a preprocessing step, the image must be smoothed by applying a filter such as Gaussian filter. This step is usually not part of the Canny algorithm and is done separately before extracting the edges.

Canny algorithm processes the image in two different phases to find all the edge pixels in the image. In the first step, it utilizes a upper threshold value to extract strong edge pixels, while in the second phase, it utilizes a lower threshold value to find edge pixels that have weaker indicators.

In the first phase, magnitude and direction of gradient for all the pixels in the image is calculated according to Eq. (2.3) using the sobel edge detector. The pixel magnitude will be compared to the upper threshold to determine if a pixel has laid on an edge or not. The direction is perpendicular to the edge direction and determines the edge's orientation. In these equations, G_x and G_y are the first derivatives in the current location in x and y directions.

$$m \quad = \quad \sqrt{G_x^2 + G_y^2} \qquad \theta = \arctan\left(\frac{G_y}{G_x}\right) \tag{2.3}$$

After calculating gradient's magnitude and direction, Canny investigates all pixels and performs non-maximum suppression to find the edge pixels. Two neighbors of each pixel are selected based on the pixel's gradient direction. The pixel will be chosen as an edge pixel, if its magnitude is greater than these neighbor pixels' magnitudes and the upper threshold.

In the last step, the pixels with less confidence on being edge pixels must be detected and added to the edge map. In this phase, for each edge pixel, its neighbors are selected according to its gradient's direction. If either of the neighbors' gradient direction is similar to the edge pixel, and its magnitude is greater than the lower threshold while it is the maximum among its neighbors (non-maximum suppression on the neighbor pixels), it will be marked as edge pixel.

2.3.4 Hough Transform

The Hough transform extract instances of a certain shape from the image [40]. In this technique, all instances of a specific shape which pass each line are extracted and among which the one with maximum vote is selected as the shape. The original Hough transform have considered only line shape, while in later researches circular and elliptical shapes are also examined. Since in this thesis we are extracting edge segments with line shape, we will cover the way Hough transform is applied to extract lines from the edge map.

Line's definition in parameter space ($y = mx + b$) produces very large values for m in the cases of vertical lines which causes calculation problems. To avoid this scenario, Duda et al have defined lines in its Hesse normal form (Eq. (2.4)) in which r is the radius from the origin and θ is the angle.

$$r = x\cos\theta + y\sin\theta \tag{2.4}$$

Several lines in various angles will be passed through each edge pixel and their radiuses (r) and angles (θ) are calculated. An accumulator bin will be generated for each pair of (r, θ) whose values will be incremented whenever a new line with those parameters are found. At the end, the pairs with maximum occurances are selected as the existing lines in the image's edge map.

2.3.5 Datasets

We have selected the following four datasets to evaluate our methods. The first two datasets contain multi-labeled images while the two latter are single-labeled datasets.

Pascal VOC 2007 was published for Pascal Visual Object Classes challenge in 2007 with the goal of recognizing objects from realistic images. It contains twenty classes of images with their labels, including person, animals, vehicles, and indoor objects. The main task in classification challenge is predicting the presence or absence of an object in the test image. This dataset includes 2501 images for training, and 4952 images for the test [47].

Pascal VOC 2012 was published for the Pascal challenge in 2012 to recognize objects from a set of real images. It contains the same classes as Pascal VOC 2007 while the images and their quantities have changed. There are 5717 training images and 5823 validation images available for this dataset. Similar to Pascal VOC 2007, for each single image, multiple labels have been provided. In all experiments regarding this dataset, we have chosen its validation set as the test set while the learning process does not have any information about the validation set [48].

Caltech 101 contains 102 different object categories, each of which has between 40 and 800 images approximately comprising 8677 images for all 102 categories. Images in this dataset come with the approximate size of 300×200 pixels. Two of the common evaluation settings for this dataset is choosing 15 or 30 images as training samples of each class and at most 50 images per class for test [51].

Caltech 256 includes 256 classes of objects, with a class of clutter, each of which includes at least 80 images with a total of 30607 images across all classes. The image's dimensions and nature are similar to Caltech 101 with removing the rotation artifacts. We have chosen 15 or 30 images per class for training purposes and up to 50 images for the test [62].

2.4 Summary

Having studied the summarized researches on the object recognition problem, each of which targeted the problem partially or fully, we came up with the conclusion that the object recognition is still a challenging topic, especially in the tasks related to image representation, and localization. Despite similarly-to-human performance of the image representation techniques using deep neural networks, this is still an open problem to find methods that are working as good on small datasets and in applications with limited computing power. On the other hand, using the deep-learning-based image representation techniques, the object recognition still does not perform similarly to human and one of its main reasons is lack of accurate object localization [59]. This understanding has motivated us for focusing on these two areas and find a way to solve them. First, we have targetted the image representation problem and proposed a local image representation technique to solve that issue. Second, we focused on object localization after the objects in the image are detected. This will improve the performance of the entire object recognition pipeline. The details about each of these methods are presented in the following chapters of this report.

Chapter 3

Image Classification by N-grams of Shape Words and Spatial Pyramids[1]

3.1 Introduction

As discussed in Literature Review (Chapter 2), object detection methods follow four different ways for finding an object in the image which are: Template Matching based methods, Knowledge based methods, Object based Image Analysis (OBIA) based methods and Machine Learning based methods [31]. In this chapter the focus is on Machine Learning based methods which contain three main steps of image representation, representation fusion, and image classification.

In this research, we introduced a local representation for the image based on its perceptual features. In the proposed methods, we extract the Generic Edge Tokens (GETs) of the image and describe traces in the image by applying the N-gram notation on GETs' combinations. In this step, we have a dictionary of visual words i.e. the N-grams of the image. These visual words are used for describing the image using the Bag of Words technique. By introducing a Shape Pyramid structure from the dictionaries of the N-grams, a hierarchical structure for our bag of visual words is generated.

The image representation obtained from the Shape Pyramid structure shows more accurate results in image classification task in comparison with local image representation methods. To achieve further improvements in our image representation, we add a hierarchical structure on selection of the local patches of the image by using the Spatial Pyramid structure applied on top of the Shape Pyramid structure. By combining these pyramids, we came up with a Spatio-Shape Pyramid structure and used it for describing the image. The experimental results show performance improvement resulted from using this structure in the proposed method.

[1] The contents of this chapter is partially published in [46, 45].

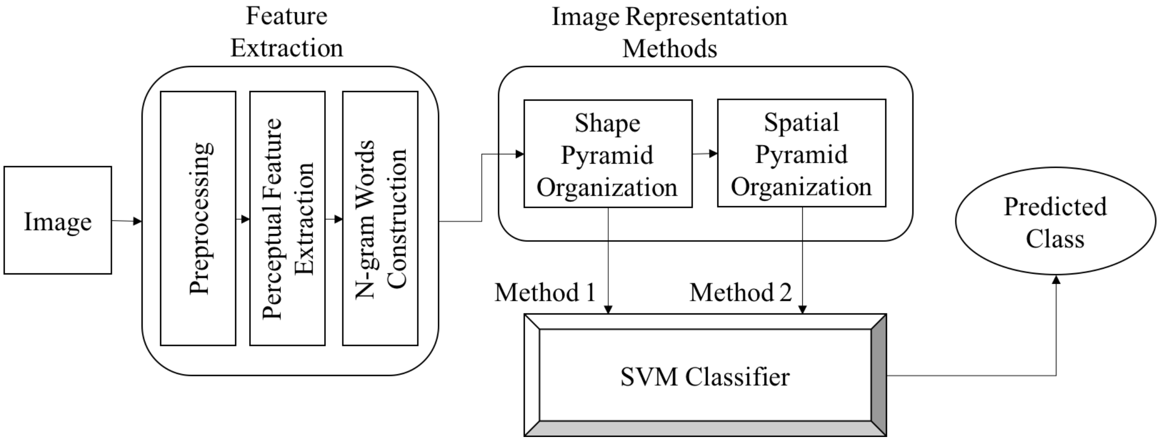

Feature Extraction
Image Representation Methods
Image
Preprocessing
Perceptual Feature Extraction
N-gram Words Construction
Shape Pyramid Organization
Spatial Pyramid Organization
Method 1
Method 2
SVM Classifier
Predicted Class

Figure 3.1: The main diagram of the image representation methods. By changing the dictionary and the descriptive blocks from flat to hierarchical, we have improved our proposed method.

We used image classification as evaluation platform for our proposed methods as it is the last step in Machine Learning based Object Detection. Our experimental results fall into two parts of parameter evaluation, and comparison. In the first part, we chose the small dataset of Wang [143] for evaluating the parameter settings of the proposed methods. In the second part, we compared our proposed methods with the well-known methods on the benchmark datasets of Caltech 101 [51] and Caltech 256 [62]. We provided detailed discussion on the performance of the proposed methods on each dataset as well as presenting class-based comparisons.

Following in this chapter, we introduced and discussed the proposed image representation methods by illustrating the N-gram representation (Section 3.2.1), Shape Pyramid (Section 3.2.2), and Spatial Pyramid structures (Section 3.2.4) and their representations (Sections 3.2.3 and 3.2.5). We also evaluated the proposed methods on benchmark datasets and compared our proposed methods to the well-known methods of the image representation in the experimental results (Section 3.3). Finally, a discussion on the advantages and disadvantages of the proposed methods as well as possible future work is provided.

3.2 GET-CPP based Representation Methods

In this research, we propose a local image representation method based on the perceptual features of the image. In this method we use the well-known N-gram model of natural language processing for describing the image. In this analogy, the image, its made-up edge traces and Generic Edge Tokens (GETs), are equivalent to the document, its sentences, and its words respectively. In this context, we are considering word N-grams of the image in which GETs are perceptual form segments of the image and used to construct the N-gram visual words. These visual words are encoded using the Shape Pyramid and the Spatio-Shape Pyramid structure for representing the image. The entire routine for image representation is summarized in Figure 3.1.

3.2.1 N-gram Representation

The BoVW technique showed improvement in image representation methods. This technique encodes the image using a set of visual words, a dictionary, which is called the Bag of Visual Words. There are many ways to find this dictionary such as clustering the local patches of the images in the training set to several clusters where each word is the representative of one cluster. When the dictionary is built, an encoding method will be applied to the images to represent them using this predetermined dictionary.

In this method, we applied an innovative way to create this dictionary. In our method, we extract N-grams of the image and treat them as the visual words. The idea of using N-grams as visual words for representing the image came from the N-gram models of Natural Language Processing (NLP). Word N-grams in NLP are sets of words that are happening together in a sliding window of size N. This window slides over the entire corpus to find the frequent N-grams. We use the same terminology for sets of N GETs that are connected to each other on the edge traces of the image.

After using the PCPG package that extracts the perceptual features of the image and organizes them into its GETs and CPPs, we are able to describe the edge traces in the image which are sets of GETs connected through several CPPs. Each trace consists of some GETs which are connected through several CPPs. If we treat an image as a document, each trace in the image corresponds to a sentence in the document. By this assumption, since each sentence can be described using the N-grams language model [11], each trace can be described using the visual N-grams as well.

To extract the visual N-grams of the image we have utilized Mukanova's definition in which each N-gram is a set of N connected GETs [107]. The very basic N-grams are

Algorithm 3.1. The algorithm to parse a single trace in an image

```
procedure ParseTrace(item $Trace$, item $N$)
    ▷ Input: The trace that needs to be parsed to its $N$-grams
    ▷ Input: Length of required N-grams
    ▷ Output: The set of $N$-grams exists in the trace
    $N - Grams\_set = \phi$                                  ▷ N-grams found in the trace
    while $Trace.Length > N - 1$ do
        $CPPs\_set = \phi$                                   ▷ CPPs in the current N-gram
        $GETs\_set = \phi$                                   ▷ GETs in the current N-gram
        $CPPs\_set = CPPs\_set \cup CPP\_0$                  ▷ Select the current CPP
        $i = 0$
        while $i < N$ do                                     ▷ Find $N$ consecutive GETs
            $GETs\_set = GETs\_set \cup GET\_i$
            $CPPs\_set = CPPs\_set \cup CPP\_i + 1$
            $i = i + 1$
        $Trace = Trace - CPP_0, GET_0$                       ▷ Slide the window for one GET
        $N - gram = CPPs\_set \cup GETs\_set$
        $N - Grams\_set = N - Grams\_set \cup N - gram$
```

defined as Unigrams which are N-grams with $length = 1$, i.e. sets with a single GET. To extract N-grams of the image, we follow the method described in Algorithm 3.1. In this method, a window with the size of N GETs slides over each edge trace in the image with the stride of one GET and all the GETs inside this window create a set that represents a single n-gram. This routine continues until the remaining length of the trace is less than N.

Each extracted Bigram (an N-gram with $N = 2$) from the image is categorized into one of the classes introduced in Table 3.1, like the clustering methods of the BoVW technique. These classes of Bigrams are defined based on the characteristics of the GET segments of each Bigram. The first characteristic which is considered is the type of the constituent GETs. If the Bigram consists of two curves, its class name comes from the similar objects in the real environment which can be observed from the first set of Bigrams in Table 3.1 and defined based on types of the GETs.

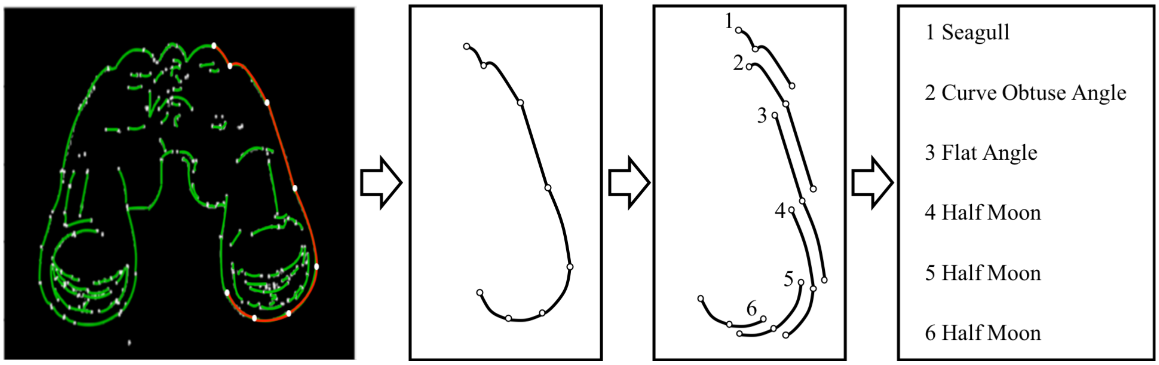


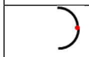
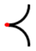
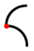
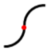
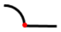
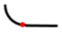
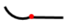
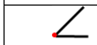
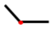
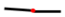
Figure 3.2: An N-gram with $length = 7$ is named to: $Seagull + CurveObtuseAngle + FlatAngle + HalfMoon + HalfMoon + HalfMoon$

Whenever at least one of the GETs from the Bigram is not a curve, the naming system uses the angle between the GETs to name the Bigram which can be seen from the rows two and three of Table 3.1. In these cases, if we have a curve GET in our Bigram, we use the word *curve* in the name of the category.

These categories of Bigrams are used for naming the longer N-grams. Each N-gram is divided into its constituent Bigrams. The name of this N-gram comes from the concatenation of the names of categories of its constituent Bigrams. An example of this naming method is represented in Figure 3.2. In the very first step, all the existing Bigrams in the N-gram are extracted and their categories are determined. The name of the N-gram is a concatenation of the category names of its constituent Bigrams. Algorithm 3.1 presents the pseudo code of this routine. Although we have not utilized this naming system in our thesis, it is an important step to bridge the gap between objects description from human and their visual display.

As the length of the N-gram increases, the likelihood of its occurrence decreases which causes sparsity in the image representation that considers these N-grams. To tackle this issue, in our experiments we have selected the N-grams whose lengths are less than or equal to two GETs, i.e. Unigrams and Bigrams, to have a more comprehensive bag of words while keeping the system robust to noises in the image and sparsity of the feature vectors. A set consisting of Unigrams and Bigrams is treated as the dictionary for encoding the image content. This set describes the shape of the objects by encoding them using the visual words which are the perceptual segments of their edges. An image representation which considers this set as its dictionary

Table 3.1: A set of possible Bigrams for describing the shape of an image.

Bigram	GET_1	GET_2	θ	Name
	CS	CS	$\simeq 90$	Half moon
	CS	CS	< 90	Leaf
	CS	CS	$\simeq 0$	Seagull
	CS	CS	< 90	Shark fin
	CS	CS	$\simeq 180$	S-Shape
	CS	LS	< 90	Curve Acute Angle
	CS	LS	$\simeq 90$	Curve Right Angle
	CS	LS	> 90	Curve Obtuse Angle
	CS	LS	$\simeq 180$	Curve Flat Angle
	LS	LS	< 90	Acute Angle
	LS	LS	$\simeq 90$	Right Angle
	LS	LS	> 90	Obtuse Angle
	LS	LS	$\simeq 180$	Flat Angle

creates a baseline perceptual representation for the image, but these flat dictionaries result in representations which are sensitive to noise, scaling, and occlusion. E.g. a curve visual word in a specific scale will be a line visual word in another scale. To solve this issue, we have considered a hierarchical dictionary which is introduced as Shape Pyramid in the next section.

3.2.2 Shape Pyramid

Although by defining the BoVW, containing Unigrams and Bigrams of the image, we can encode the shape of the objects, considering a hierarchy of visual words improves the performance of the proposed method. In this section, the Shape Pyramid structure is defined and investigated.

The reason behind considering a hierarchy of visual words in our representation is enhancing the robustness of the proposed method against changes in the scale of the image. Due to scaling, some of the curve visual words with a small amount of curvature may turn into lines. On the other hand, noise in the environment or noise in capturing the image and extracting the GETs have the same effect on the type of

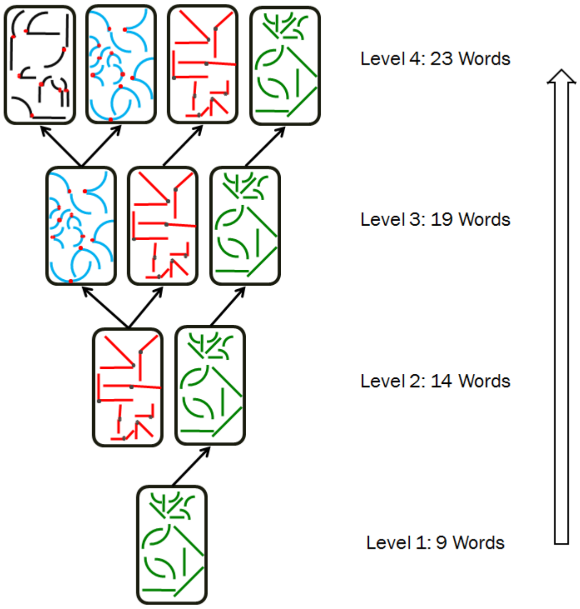


Figure 3.3: The Shape Pyramid structure.

extracted GETs. The Shape Pyramid structure (represented in Figure 3.3) improves the base model to some extent.

At the first level, we assume that it is possible to represent all the edges of the object by using only Unigram visual words. This is a valid assumption since Unigrams are basic segments of the edges (GETs) and they can represent the edges together. The detail definition of GETs are reviewed in Section 2.3.2 where the GETs are classified to eight groups of lines and curves according to their directions and curvatures. At this level, we have also defined Uninoise which is a set of Unigrams whose lengths are less than a predefined threshold. These noise Unigrams are treated in a similar way and as a unique visual word. This visual word helps us to represent areas of the image with simple texture such as the texture of the grass region of the image.

The dictionary of the second level of the pyramid not only has the Unigrams, but also has very simple Bigrams of the image whose constituent GETs are just lines. In this situation, we encode the relationship of line visual words together, while we encode the curve edges just by using the Unigrams. This level adds a higher level of representation by considering more visual words. In this level, we add a visual word which is called Binoise. This visual word is the representative for all Bigrams shorter than a predefined threshold. This visual word provides a tool for representing areas with more complex texture such as leaves.

The third level of the Shape Pyramid has more Bigrams to represent the relationship between two curves as well as the relationship between two lines. Using this set of visual words, we can represent objects which are made of curves such as balls and the objects which are made of lines such as pens with Bigrams, while they previously were encoded by Unigrams only. In this level, the Binoise visual word is representative of nine types of small Bigrams, where this visual word represented just four small Bigrams in the previous level.

By adding the Bigrams that consist of a line and a curve, the set of visual words becomes more mature. This means that we can represent almost everything with this dictionary. From level two to level four of the pyramid, the Binoise visual word is the representative for different sets of the small Bigrams depending on the Bigrams which are considered in each level. For instance, in the fourth level of the pyramid, the Binoise visual word is representative of 13 small Bigrams.

Each level of this pyramid, by itself, can represent the shape of objects in the scene. Although, by having the hierarchy structure in our representation, we may have four different representations for an object in the image to consider different scales of the objects in the scene and resist against noise. On the other hand, different levels of this pyramid have 9, 14, 19, and 23 visual words which means that each level has a different capability for describing the scene from which the higher level is the more descriptive one, and the lower level provides the very basic description for the objects in the scene.

3.2.3 Shape Pyramid based Representation

In this section, the proposed image representation method using the Shape Pyramid structure is illustrated. At first, the basic idea of representing an image using a flat dictionary is discussed, then the proposed Shape Pyramid based representation method is introduced.

The proposed image representation method is a local image descriptor. This means that it describes the image using the descriptors of its local patches. These local patches are obtained by dividing the image into $N \times N$ non-overlapping blocks, where each block is a local patch. To describe each block, we use a dictionary with M visual words. The pseudo code of describing an image using a predefined dictionary

Algorithm 3.2. Basic Image Representation

```
procedure BasicRepresentation(item $GETMap$,item $Dictionary$)
    ▷ Input: Divided GET Map of the image
    ▷ Input: Dictionary of Visual Words
    ▷ Output: Image Representation Histogram
    $N = Number \quad of \quad Local \quad Patches$
    $M = Dictionary \quad Size$
    for $j = 1 \to N \times N$ do
        for $i = 1 \to M$ do
            $SumFrequency = SumFrequency + f_{i,j}$
            ▷ Summation of frequencies of each word in the image.
            $SumLength = SumLength + l_{i,j}$
            ▷ Summation of lengths of each word in the image.
            $Frequency = [Frequency, f_{i,j}]$
            $Length = [Length, l_{i,j}]$
        $Frequency = Frequency/SumFrequency$
        ▷ Normalizing the frequency of each word in the patch.
        $Length = Length/SumLength$
        ▷ Normalizing the length of each word in the patch.
        $Histogram = [Histogram, Frequency, Length]$
        ▷ Concatenating the normalized frequency and length of the current patch with others.
```

is presented in Algorithm 3.2.

For each of these local patches, we calculate two metrics of frequency and length. The frequency represents the number of occurrences for each of the visual words in that local patch, while the length shows the total length of each visual word in terms of the pixels. Since, on one hand, the range of length values is different from the range of the frequency values, and on the other hand, because the images' dimensions are different, the length values may be very different and we need to apply normalization on both metrics.

$NormalizedFrequency$ of visual word i in the local patch j is calculated using the Eq. (3.1) where M is the number of visual words in the dictionary. In this equation,

Algorithm 3.3. Shape Pyramid based Image Representation

```
procedure ShapeRepresentation(item $GETMap$,item $ShapePyramid$)
    ▷ Input: Divided GET Map of the image
    ▷ Input: Shape Pyramid structure
    ▷ Output: Image Representation Histogram
    $L = Shape \quad Hierarchy \quad Levels$
    for $l = 1 \to L$ do
        $FindDictionary_l$
        $RepLevel_l = BasicRepresentation(GETMap, Dictionary_l)$
        $Histogram = [Histogram, RepLevel_l]$
```

the number of occurrences of the selected visual word in the current local patch, $f_{i,j}$, is divided into the total number of occurrences of all the visual words in that local patch.

$$NormalizedFrequency_{i,j} = \frac{f_{i,j}}{\sum_{k=1}^{M} f_{k,j}} \tag{3.1}$$

The $NormalizedLength$ of visual word i in local patch j, when the size of the dictionary is M, is calculated by dividing the length of the selected visual word in the current local patch, $l_{i,j}$, into the total number of edge pixels in that local patch. The formula for this calculation is presented in Eq. (3.2).

$$NormalizedLength_{i,j} = \frac{l_{i,j}}{\sum_{k=1}^{M} l_{k,j}} \tag{3.2}$$

To represent an image using the Shape Pyramid, we create four basic representations, one for each level of the Shape Pyramid. This means that we represent an image using four different dictionaries corresponding to different sets of visual words for different levels. The final representation for the image is obtained by concatenating these basic representations together. The general algorithm of this method is represented in Algorithm 3.3.

The Shape Pyramid based representation module suffers from the lack of location information. Each local patch is described independently and their relationship to one another is ignored. To mitigate this issue, we have considered integrating the spatial pyramid coding method into our image representation method which is described in the next section.

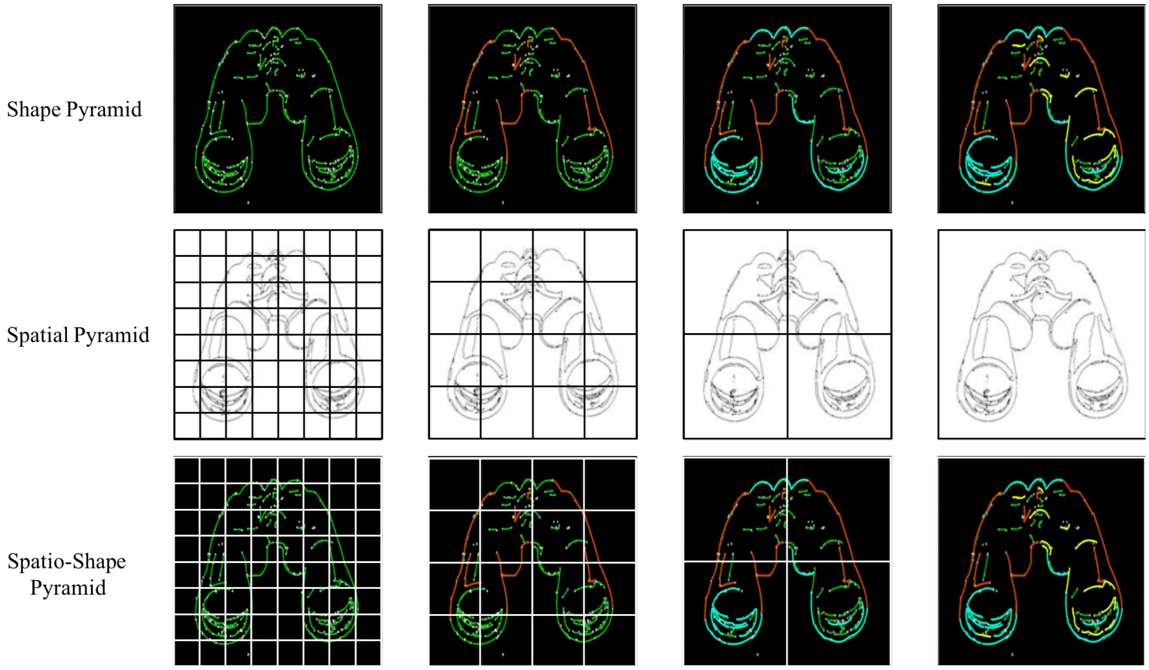


Figure 3.4: The Spatio-Shape Pyramid structure.

3.2.4 Spatio-Shape Pyramid

Although the Shape Pyramid structure helps us to improve the accuracy of image classification, we may have more improvement by applying the Spatial Pyramid structure to create the Spatio-Shape Pyramid which is represented in Figure 3.4. This pyramid brings the advantages of both Spatial and Shape pyramids in a single pyramid of their combination.

The Spatial Pyramid structure helps us to provide a coarse to fine representation for the image. In the first level of this pyramid, the image is considered as a local patch for description to provide a global understanding of the image. On the other hand, in the fourth level of this pyramid, local patches are the smallest local patches in this structure and provide local information about the image content. In the middle levels of the Spatial Pyramid, the local patches with the larger size are considered for description. Using this structure provides both global and local information about the image. The Spatial Pyramid which is used in this research consists of four levels where different levels are divided into $N \times N$ blocks where $N = 1, 2, 4, 8$ correspond to levels 1 to 4 of the pyramid respectively.

In the definition of the Spatio-Shape Pyramid structure, we assigned a spatial layout to each level of the Shape Pyramid to represent the image. This assignment is performed by considering the size of the local patch and the approximate normalized size of the visual words in the dictionary. For instance, when the spatial layout with 8×8 division is considered, the dictionary with only Unigrams and Uninoise is used for image description. The reason for this choice comes from the idea that using larger dictionaries will cause a sparse image representation since the probability of occurrence for longer n-grams in those small patches is trivial.

On the other hand, for describing the image where the local patch equals the entire image, we considered the largest dictionary which consists of all Unigrams, Bigrams, Uninoise, and Binoise. In this case, we have more power for describing the image since we have more visual words.

3.2.5 Spatio-Shape Pyramid based Representation

To represent an image using the Spatio-Shape pyramid which is defined in Section 2.1.1, for each level of the pyramid, we have two choices to make: the spatial layout, and the dictionary. For each level of the Spatio-Shape Pyramid, a basic image representation is obtained where the spatial layout and the dictionary for this description are selected based on the definition of the Spatio-Shape Pyramid. After calculating four basic representations, the final representation for the image is a concatenation of those basic representations together. The pseudo code for the Spatio-Shape representation is illustrated in Algorithm 3.4. Although considering the spatio-shape pyramid improves the performance of the image representation module, it increases the computation and memory complexity of the image representation. The choice for considering which of these methods depends on the nature of the application and its computing power.

3.3 Experimental Results

In this section, we are going to evaluate our proposed methods on the benchmark datasets. As an evaluation platform, we chose the image classification domain and performed a set of preprocessing steps to make the testing datasets uniform, in terms of their dimension and their color scale to be ready for this purpose. We performed a set of experiments on a small size dataset to figure out which parameters are more

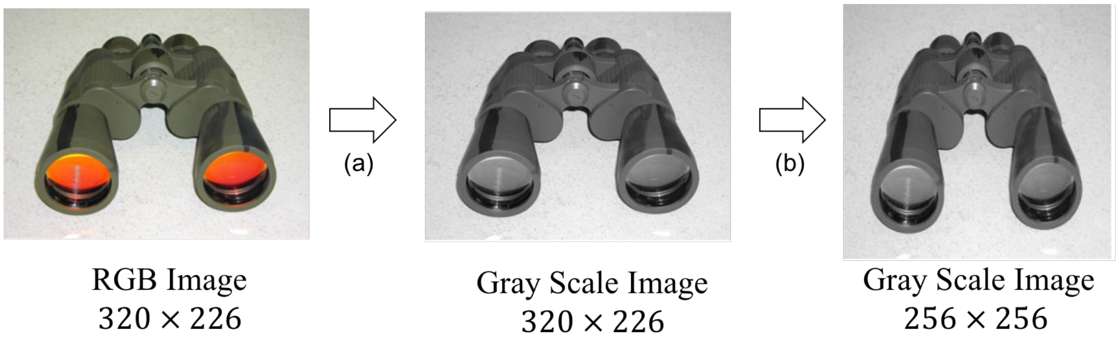
Algorithm 3.4. Spatio_Shape Pyramid based Image Representation

```
procedure SpatioShapeRepresentation(item $GETMap$,item $ShapePyramid$,item $SpatialPyramid$)
    ▷ Input: Divided GET Map of the image
    ▷ Input: Shape Pyramid structure
    ▷ Input: Spatial Pyramid structure
    ▷ Output: Image Representation Histogram
    ▷ S=Spatio_Shape Hierarchy Levels
    $S = [S_s, S_l]$
    for $l = [S_s, 1] \rightarrow [1, S_l]$ do
        $Divide \quad GETMap \quad Into \quad S_s \quad Local \quad Patches$
        $LevelRep_l = BasicRepresentation(GETMap, DictionaryS_l)$
        $LevelRep = [LevelRep, LevelRep_l]$
        $Histogram = [Histogram, LevelRep]$
```

suitable for the experiments. We also compared different settings of our proposed methods with other well-known methods for image representation on larger datasets. The global and class-based comparisons are provided. In our experimental results, we used ν_SVC classifier from the LibSVM package [27] and chose the Radial Basis Function as the kernel for the SVM classifier. We selected ν to be 0.5 in our experiments and we chose the *Gamma* equal to 0.05.

Preprocessing

To perform our evaluation of the proposed methods, we tested our methods on different benchmark datasets which are introduced following in this section. Since the images in each of these datasets have various dimensions and color scales, and since the well-known methods that we compare with have some limitations in terms of the color scale of the images, we chose to perform some preprocessing tasks on the images to unify their dimension and color. The first task is changing the color scale of the image. Some images in datasets are in RGB color scale, while most of the compared image representation methods use the images in their grayscale. To be able to compare our method with other existing methods, and to unify our evaluation, we have changed all the images in the dataset to grayscale images. After unifying the



(a)

Figure 3.5: Preprocessing of an image. (a) The left image is in RGB color scale and changed to Gray scale image of the right. (b) The image with rectangular dimension (in the left) is transformed to a square image in the right side.

color scale, we make the image dimensions equal. Images in the dataset has different dimension values and this makes it hard to process them in a constant way. In this case, we changed the dimension of the images into a standard dimension of 256×256. In the definition of the images, their dimensions represent the number of pixels in each row and each column of the image. Figure 3.5 shows an image before and after the re-sizing process.

3.3.1 Parameter Evaluation

There are some parameters in the proposed image representation methods whose values may affect its performance. To find the best setting of the proposed method, we conducted a set of experiments on the small dataset of Wang [143].

The Wang dataset consists of ten classes with a hundred images in each class. This dataset has classes of urban areas as well as some classes of nature and animals. It has classes with very well-shaped contents such as Dinosaurs, as well as classes with clutter areas such as beach and foods.

In the parameter evaluation phase, we used the 10-fold cross validation method [88] to verify the reliability of the chosen parameters. The evaluation metric which is utilized in this research, to investigate the parameters' effects on the performance of the proposed method, is the *accuracy* metric which is defined using the Eq. (3.3). The values of TP, TN, FP, FN are calculated as True Positive, True Negative, False

Positive, and False Negative samples in each experiment respectively.

$$
\begin{aligned}
Accuracy &= \frac{TP + TN}{P + N} & Precision &= \frac{TP}{TP + FP} \\
Recall &= \frac{TP}{TP + FN} & F_1 = 2 \times & \frac{Precision \times Recall}{Precision + Recall}
\end{aligned}
\tag{3.3}
$$

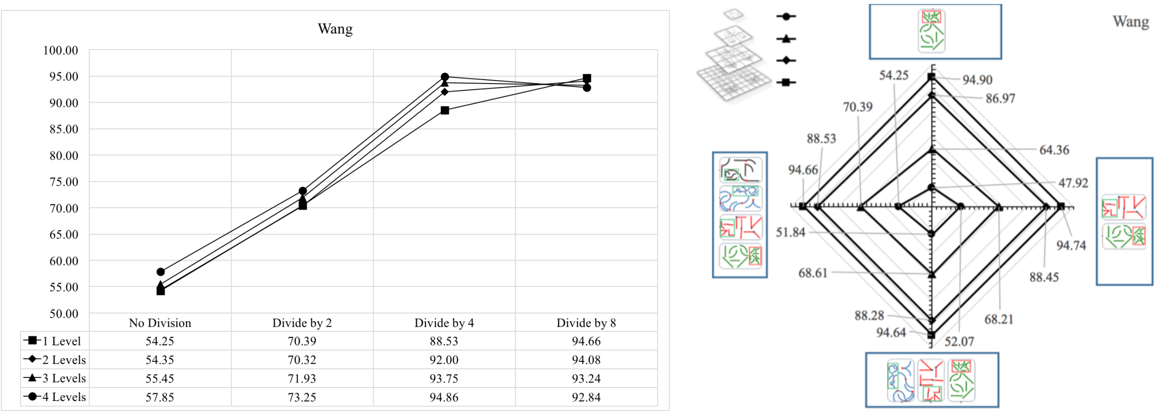
In our experimental settings, we chose the threshold on the length of the visual words to be considered as noise equal to 10 pixels. Considering the smallest local patch in our experiments, this means that the size of the noise is at most 1/100 of the size of the patch, where the patch size is 32 × 32 pixels, which is reliable.

1. We performed a set of experiments to see which sizes of the local patches, and which sizes of the dictionary, produce more accurate results and represented their corresponding results in Figure 3.6. This shows that by reducing the size of the local patches and dividing the image into more blocks, the accuracy results of the proposed methods are improved. We also see that considering more visual words and increasing the size of the dictionary, slightly improves the accuracy results of the proposed method.

2. We evaluated the number of hierarchy levels for each of the pyramids. We considered 1,2,3, and 4 levels of hierarchy for the Spatial Pyramid and the accuracy results are represented in Figure 3.6(a). This shows that increasing the number of the hierarchy levels improves the accuracy performance of the proposed method. This behavior was predictable since by adding more levels of the hierarchy, we add more local information into our representation and capture the distribution of the visual words more precisely.

   To evaluate the number of hierarchy levels of the Shape Pyramid, we selected 1, 2, 3, and 4 levels of the pyramid where one level just consists of 9 visual words while the pyramid with 4 levels consists of 9, 14, 19, and 23 visual words. The evaluation results which are represented in Figure 3.6(b) show that the pyramid with 4 levels of hierarchy produces more accurate representation of images.

   Interestingly when we increase the number of blocks (local patches) of the image from 4 × 4 to 8 × 8, the accuracy result of the larger dictionary sizes reduces slightly. This happens due to the fact that many of the words will not happen in



	No Division	Divide by 2	Divide by 4	Divide by 8
■ 1 Level	54.25	70.39	88.53	94.66
◆ 2 Levels	54.35	70.32	92.00	94.08
▲ 3 Levels	55.45	71.93	93.75	93.24
● 4 Levels	57.85	73.25	94.86	92.84

(a)

(b)

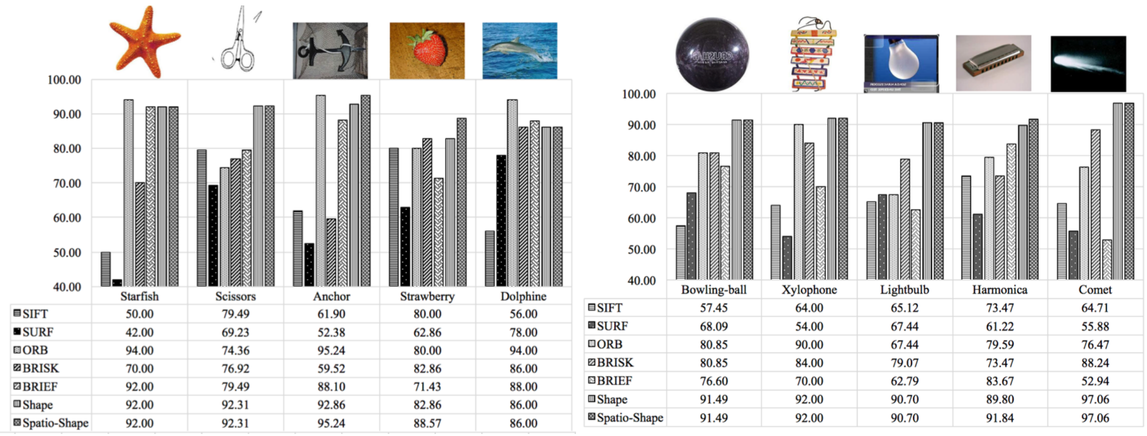
Figure 3.6: (a) The effect of changing the local patch size on the performance of the proposed methods. (b) The effect of considering different hierarchy structures for the utilized Shape and Spatial pyramid.

small regions which makes the representation sparse. Due to this reduction, we did not keep on dividing the image into smaller local patches, such as dividing it into 16 × 16, and so on.

3.3.2 Comparison

To demonstrate our contribution in this research, we compared our proposed methods with the well-known local image representation methods of *SIFT*, *SURF*, *ORB*, *BRISK*, and *BRIEF* on the benchmark datasets of Caltech 101 [51], and Caltech 256 [62]. For representing image using these feature vectors, we either used their own keypoint detection methods; for *SIFT*, *SURF*, *ORB*, and *BRISK*; or we used the *FAST* keypoint detection method to find the local patches. We created a Bag of Words with 500 visual words to represent the images using these local descriptors. We also compared the proposed methods with the deep learning features obtained by training the AlexNet model [90] using the Caffe framework [80] on these datasets.

Caltech 101 is a benchmark dataset for comparing the performance of different image processing tasks such as image classification. This dataset consists of 102 classes of images with a different number of images in each class. In our experiments, since we wanted to have a uniform distribution of samples per class, we chose at most 50 images from each class as a test set, and the other as the training set, to report



	Starfish	Scissors	Anchor	Strawberry	Dolphine
▤ SIFT	50.00	79.49	61.90	80.00	56.00
■ SURF	42.00	69.23	52.38	62.86	78.00
▩ ORB	94.00	74.36	95.24	80.00	94.00
▨ BRISK	70.00	76.92	59.52	82.86	86.00
▧ BRIEF	92.00	79.49	88.10	71.43	88.00
▥ Shape	92.00	92.31	92.86	82.86	86.00
▦ Spatio-Shape	92.00	92.31	95.24	88.57	86.00

(a) Caltech 101

	Bowling-ball	Xylophone	Lightbulb	Harmonica	Comet
▤ SIFT	57.45	64.00	65.12	73.47	64.71
■ SURF	68.09	54.00	67.44	61.22	55.88
▩ ORB	80.85	90.00	67.44	79.59	76.47
▨ BRISK	80.85	84.00	79.07	73.47	88.24
▧ BRIEF	76.60	70.00	62.79	83.67	52.94
▥ Shape	91.49	92.00	90.70	89.80	97.06
▦ Spatio-Shape	91.49	92.00	90.70	91.84	97.06

(b) Caltech 256

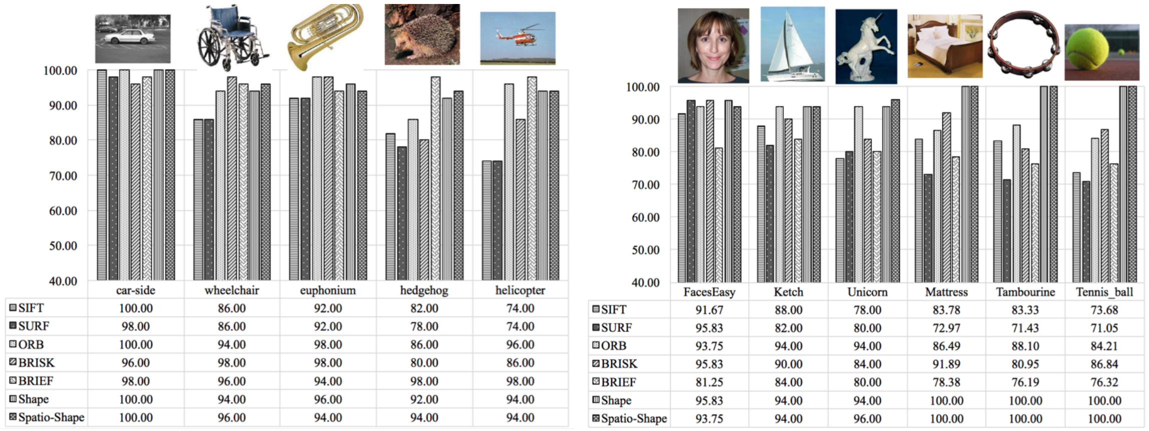
Figure 3.7: Comparison of image representation methods on hard classes for classification.

the accuracy results.

Caltech 256 is selected as another benchmark dataset for evaluating the image representation. This dataset consists of 257 classes of images with a different number of images per class. Again, for this dataset, we chose the number of test images per class equal to 50 at most, and the other images as the training set.

We compared the accuracy results of the proposed and benchmark methods on classes with the weakest and strongest predictions for at least one image representation methods. The ideas behind choosing these classes are showing the superiority of the proposed methods on classes that are hard for other methods to represent, and to show the similar performance on classes in which other representation methods perform accurately.

The results of the weakest predictions, hardest categories to classify, are presented in Figure 3.7 and show that for classes of Scissors and Strawberries (from Caltech 101 dataset) the proposed methods outperform all the others, while for Starfish and Anchor they work like ORB while they are superior to the others. The reason for this behavior is the well-shaped objects of these classes which help our representation to encode images content more accurately. In the case of Dolphins, the proposed method works weaker than ORB and like BRIEF and BRISK, while they are still more accurate than SIFT and SURF. The shapes of both objects are rotation sensitive and have unique shape features (ie. star, and arch shape). The possible reason for



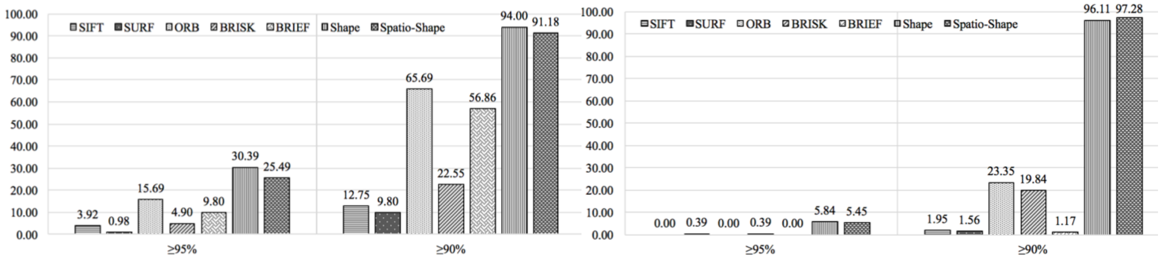
	car-side	wheelchair	euphonium	hedgehog	helicopter
▤ SIFT	100.00	86.00	92.00	82.00	74.00
■ SURF	98.00	86.00	92.00	78.00	74.00
▦ ORB	100.00	94.00	98.00	86.00	96.00
▨ BRISK	96.00	98.00	98.00	80.00	86.00
▩ BRIEF	98.00	96.00	94.00	98.00	98.00
▥ Shape	100.00	94.00	96.00	92.00	94.00
▩ Spatio-Shape	100.00	96.00	94.00	94.00	94.00

(a) Caltech 101

	FacesEasy	Ketch	Unicorn	Mattress	Tambourine	Tennis_ball
▤ SIFT	91.67	88.00	78.00	83.78	83.33	73.68
■ SURF	95.83	82.00	80.00	72.97	71.43	71.05
▦ ORB	93.75	94.00	94.00	86.49	88.10	84.21
▨ BRISK	95.83	90.00	84.00	91.89	80.95	86.84
▩ BRIEF	81.25	84.00	80.00	78.38	76.19	76.32
▥ Shape	95.83	94.00	94.00	100.00	100.00	100.00
▩ Spatio-Shape	93.75	94.00	96.00	100.00	100.00	100.00

(b) Caltech 256

Figure 3.8: Comparison of the image representation methods on classes with higher accuracy of prediction.



(a) Caltech 101

(b) Caltech 256

Figure 3.9: Comparison of the image representation methods on the number of classes with more accurate prediction.

this behavior is the fact that ORB employs intensity centroid for strong rotation invariance property, thus, the accuracy of ORB is higher since it can tolerate shape variance. Though our GET/CPP has quasi-rotation invariance property, compared to ORB, it is worthy of improving it in the future. Considering the Caltech 256 dataset, these results show that the proposed methods provide more than 90% accuracy in almost all the classes since most of them have well-defined shapes.

In comparison of the methods on the easiest classes (from Caltech 101 dataset) to predict, presented in Figure 3.8, the proposed methods provide 100% accuracy for Car_side class while their accuracy on other classes is more than 94% which is better than SIFT and SURF in general and like the others. A similar behavior is observable from the results of the representation methods in Figure 3.8, where the proposed methods obtained 100% accuracy in four classes of Caltech 256. By having a closer look at images from these classes, objects with precise boundaries can be found.

An interesting fact in a comparison of the proposed method with the other selected methods is the percentage of classes where the proposed method predicts accurately. These results, which are represented in Figure 3.9, show that while the best baseline method predicts about 16% of classes from Caltech 101 with accuracy higher than 95%, our proposed methods accuracy for 30% of classes is higher than 95% while this rate goes to higher than 90% for classes with 90% accuracy in prediction. This trend exists in the experiments on Caltech 256 where the proposed methods predicted more than 96% of classes with accuracy higher than 90%.

Comparing our proposed methods together shows that adding more features for representing the image by using the Spatial Pyramid structure does not have an obvious positive effect on average on these datasets; although, in some of the classes the SpatioShape structure provides better results, see group precision for both Caltech 101 and Caltech 256 datasets in Table 3.2 and Table 3.3 respectively. The choice of considering the longer feature vector or the shorter one depends on the application and the platform.

The global results of comparing the proposed methods with the well-known methods on both Caltech 101 and Caltech 256 are presented in Figure 3.10. The proposed methods by obtaining more than 93% accuracy on average, on Caltech 101, and more than 92% accuracy on Caltech 256; improved the accuracy results of the local image classification methods while its accuracy is still less than the deep learning representation. On the other hand, by comparing the precision of the proposed methods with all the benchmark methods, this figure shows the superiority of the proposed methods in both datasets.

Not only were the accuracy results of the proposed methods compared with the well-known methods, but also the precision, recall and F-measure values are considered as performance metrics for our evaluation. The evaluation results based on these metrics, defined in Eq. (3.3), show that the proposed methods in both mean and standard deviation values outperform the other local representation in all the metrics, while its precision, recall, and F-measure outperform the deep learning representation. These results are presented in Figure 3.10 for Caltech 101 and Caltech

Table 3.2: Group precisions for different classes of Caltech 101 dataset.

Classes	Size	SIFT	SURF	ORB	BRISK	BRIEF	CNN	Shape	Spatio Shape
Animal	21	75.28	91.71	70.50	91.12	82.65	91.34	77.00	**93.72**
Toy	3	80.07	66.83	87.77	72.80	88.23	76.67	92.23	**100.00**
Transport	8	83.89	81.41	93.78	87.69	85.20	74.87	**95.15**	94.25
Fashion	2	87.90	84.65	92.25	95.85	92.35	85.96	96.15	**97.95**
Plants	5	84.00	81.30	93.96	88.50	92.62	78.56	92.50	**96.00**
Science	5	82.00	76.64	90.26	74.36	87.48	87.91	**96.46**	95.12
Desktop	5	76.70	68.76	89.98	87.06	91.28	83.67	95.24	**95.98**
Sport	2	89.20	89.70	89.50	83.40	95.95	91.67	**96.25**	89.70
Music	7	79.66	79.13	87.61	85.53	87.61	84.97	96.21	**98.29**
Tools	5	84.24	77.48	90.48	81.06	91.58	71.00	97.24	**98.58**
Military	4	90.88	72.45	87.55	93.93	89.63	98.33	95.43	**95.83**
Civilization	4	74.45	67.85	88.43	77.85	87.80	95.99	92.08	**97.43**
Restaurant	4	73.45	77.65	88.28	83.20	85.05	83.48	96.45	**98.45**
Insect	6	91.65	89.32	93.88	88.00	94.08	67.44	**94.48**	93.43
Design	7	80.51	75.21	93.29	86.86	92.19	81.18	92.39	**96.59**
Bird	5	88.40	73.08	92.16	90.58	94.62	85.71	92.70	**96.36**
Marine	9	75.94	81.57	95.26	87.81	92.81	59.37	92.56	**93.18**

256 respectively.

3.4 Summary

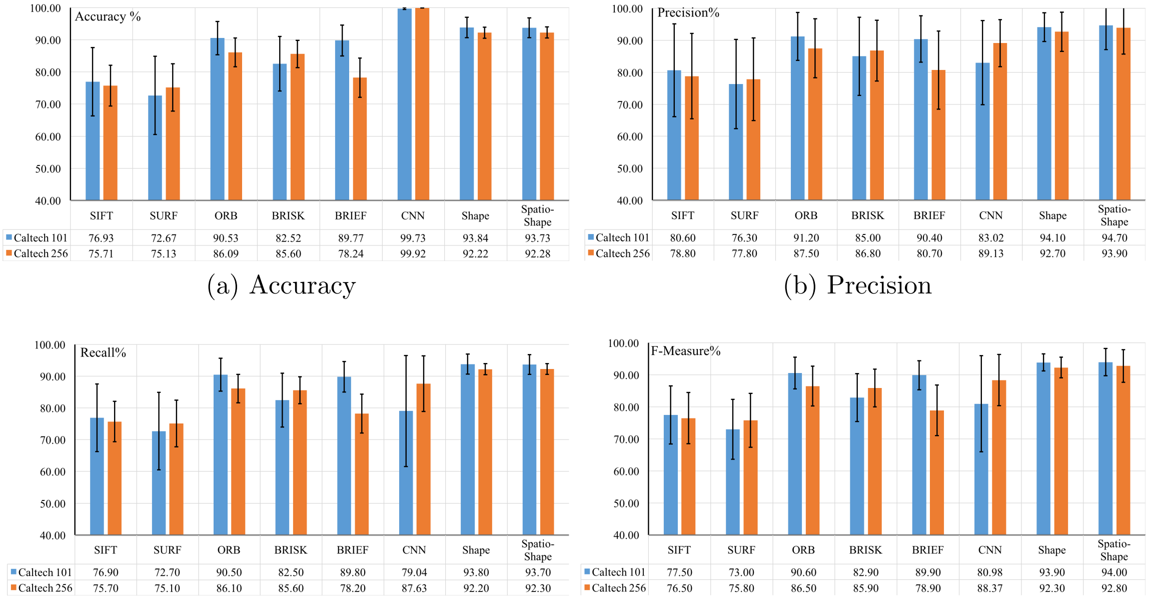
In this chapter, we proposed an image representation method based on the perceptual segments of image's objects' shapes. We extracted Generic Edge Tokens (GETs) of the image which represent the perceptual segments of the image using PCPG package. These tokens are utilized to categorize N-grams of the image. These N-grams are used as a dictionary of visual words for representing images. We applied a hierarchical structure to the utilized dictionary for image representation by introducing the Shape Pyramid structure. We augment our representation with the spatial structure of the visual words in the scene by imposing the Spatial Pyramid structure on top of the Shape Pyramid to introduce the Spatio-Shape Pyramid structure. We represent images using the lengths and frequencies of these visual words in the image.

Parameters of the image representation methods are determined by performing a set of evaluation experiments. Another experiment was also conducted to compare the performance of the proposed methods with the well-known image representation

Table 3.3: Group precisions for different classes of Caltech 256 dataset.

Classes	Size	SIFT	SURF	ORB	BRISK	BRIEF	CNN	Shape	Spatio Shape
Animal	28	73.94	83.57	90.10	88.34	82.21	69.32	92.00	**94.24**
Toy	10	79.65	81.15	89.59	87.36	85.87	63.27	93.96	**94.40**
Transport	22	80.28	76.81	83.49	85.81	74.91	67.21	**94.09**	93.29
Fashion	11	83.53	76.00	90.17	87.48	84.86	73.87	**94.60**	93.87
Plants	10	66.63	80.23	88.86	89.76	84.37	76.52	93.65	**95.36**
Science	13	80.13	81.79	84.26	85.42	75.05	74.89	91.95	**93.07**
Desktop	17	82.01	78.16	89.08	85.46	80.19	65.95	**94.36**	91.28
Sport	21	81.67	75.50	89.89	86.04	80.60	71.26	92.72	**94.39**
Music	14	81.25	69.64	86.14	84.21	78.00	76.47	93.37	**95.31**
Tools	19	84.55	77.18	86.66	83.78	80.79	69.91	92.88	**95.84**
Military	6	83.10	81.48	91.60	90.95	80.08	63.22	91.62	**95.62**
Civilization	13	85.18	76.51	86.08	87.92	80.67	78.23	93.82	**97.66**
Nature	7	60.06	75.49	87.94	91.79	83.76	75.68	93.03	**96.91**
Restaurant	24	77.44	76.10	84.33	88.11	79.73	62.49	89.76	**89.77**
Insect	9	80.61	72.10	84.87	84.63	82.56	69.37	**93.40**	93.00
Design	7	80.21	72.34	86.21	84.27	83.71	71.37	90.07	**93.33**
Bird	9	83.72	80.10	88.64	89.30	82.17	74.00	90.02	**94.18**
Marine	7	69.93	79.69	87.79	88.24	80.06	69.67	93.39	**93.80**

methods. These results illustrate the superiority of the proposed method in comparison with the well-known methods for image representation. Besides performance gain, the proposed method incorporates perceptual shape features of the image. This may be beneficial in applications that search for images by providing sketches of the image. Usually sketches are simpler versions of the objects with only their shape and boundaries. This can help our system to match sketches of objects with their real version.



(a) Accuracy

(b) Precision

(c) Recall

(d) F_Measure

Figure 3.10: Global comparison of the image representation methods on Caltech 101 and Caltech 256 datasets.

Chapter 4

Hybrid Image Coding using Bag of Shape Tokens from Octaves of Edge Segments [1]

4.1 Introduction

Although the N-gram based image representation method considers the perceptual characteristics of the human vision, it introduces classes of N-grams that are defined using rule-based algorithms considering the curvature, direction, and angle between their constituent GETs. This definition limits our words to those that are seen by human and may ignore some of the frequent N-grams. As a result, we have decided to create our Bag of Words automatically by using the clustering algorithm which groups similar edge segments in a cluster.

Our proposed method extracts perceptual structure-based edge segments from the image edge map, describes the area around them, and clusters those segments to find edge tokens. Each image will be encoded using these edge tokens obtained from the training set. In our proposed method, we have considered octaves of images, where different smoothing filters are applied to each octave to extract edge maps. Smoothing an image using different parameters helps us to have a hierarchy of edge segments, since edges obtained from the most smoothed image are coarser and less noisy and representative of objects' boundaries. Going to less smoothed versions, we will find finer edges representing smaller objects and textures in the image.

We utilize the Canny Edge detection algorithm [22] along with the Hough transform [54] to find edge segments of the image in each level of smoothing, and each octave. The feature vectors for these segments are created by applying a local descriptor to the area around them. These feature vectors are clustered using K-means algorithm to find edge tokens from the training set. These tokens are utilized by the proposed method to find an encoding for each image.

[1] The contents of this chapter is partially published in [44].

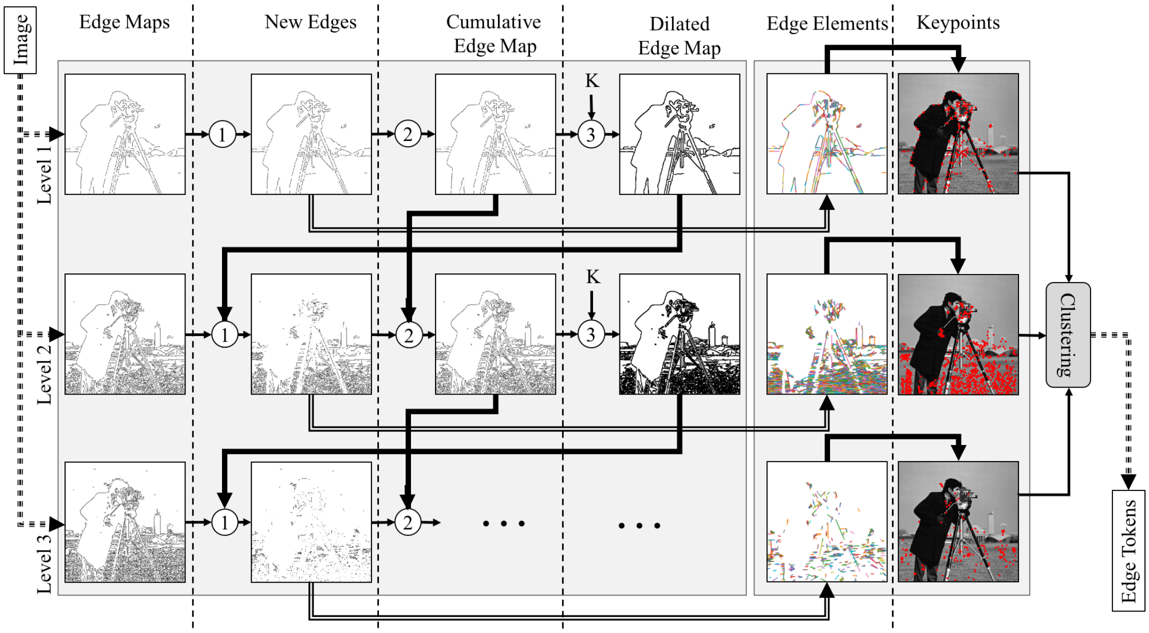


Figure 4.1: The main diagram of the proposed method for finding edge segments from different octaves of image in different smoothing levels. (1) Remove the previously seen edges. (2) Add new edges to the cumulative edge map. (3) Dilate the cumulative edge map with kernel K.

Our proposed method has been tested on the multi-class multi-label image classification problem and its performance comparison is elaborated in the experimental results section. In multi-class multi-label classification problems a single image has multiple labels each of which considers one of its objects and the entire dataset contains more than one object class. Our evaluation shows that the proposed method has improved the results in this challenging task by around 2%, while its time complexity is in the same range as the existing methods.

In Section 4.2 of this chapter, we introduce the details of our proposed method. Our experimental setting, and evaluation results, followed by discussion on the obtained performance, are presented in Section 4.3. Finally, this chapter is summarized in Section 4.4.

4.2 Proposed Method

The proposed image representation method focuses on the fact that human visual perception mostly relies on the shapes of objects. Since an object's shape is defined by its boundaries, just considering the corner points as descriptive areas is not sufficient to carry structural information. The edge map of the image is a representation of its shape. In our proposed method, we consider octaves of an image to have robustness against object scaling. Each octave has been created by re-sizing the input image to a specified scale. For each octave we extract perceptual structure-based shape descriptors (edge segments) from a hierarchy of edge maps using various smoothing parameters. We extract edge segments from each of these edge maps and describe their surrounding areas using a local descriptor. Descriptors in the training set are utilized to find a set of edge tokens using K-means clustering algorithm. Each image will be encoded according to these edge tokens.

The proposed keypoint detection method utilizes the edge characteristics of the image as a tool to find important locations in the image. Edges are representing boundaries of objects, and separate foreground objects from background. According to this fact, not only the corners are important points in the image, but also the edge segments. To extract these keypoints, we utilize the canny edge detection algorithm, followed by the Hough transform to find the edge tokens in the image. For each token, we have calculated the keypoint location, the descriptive area around it, and the rotation of edges in respect to the image coordinates. These features are utilized by existing local feature descriptors such as SIFT and SURF to produce more descriptive features. The main diagram of the proposed method is represented in Figure 4.1.

4.2.1 Edge Segments

In the proposed method, we have considered several octaves of images for finding edge segments each of which is a resized version of the input image to a specified scale of 2^k and smoothed for finding the coarse and fine edge maps. Image scaling results in extracting perceptual segments from various sizes of an object. As a result, more descriptive and general edge tokens are generated, and the image representation is more robust against object scaling.

The image from each octave is processed to create a hierarchy of edge maps smoothed by applying the Gaussian filter to the image. In this equation, a filter with the size of k is created whose elements are calculated according to Eq. (4.1) in which σ is the standard deviation. Each edge map obtained from a smoothed image with a different value for σ produces different edge segments. The higher the value for σ, the smoother the image, and the less noisy and coarser the edge map. This hierarchy creates edge segments with longer and less noisy structures, as well as shorter and noisier structures. This wide range of edge segments can describe objects with very well-defined boundaries and objects with noisy textures.

$$H_{ij} = \frac{1}{2\pi\sigma^2} e^{(\frac{-(i-(k+1))^2+(j-(k+1))^2}{2\sigma^2})}, \quad 1 \leq i, j \leq (2k+1) \tag{4.1}$$

The Canny Edge Detection method [22] is applied to each smoothed image in each octave to detect its edge map. Given the smoothed image, Canny calculates the intensity gradient of the image, and cancels false detected edges by applying a non-maximum suppression. By applying double thresholding, Canny finds potential edges and tracks those edges to reject the detected segments which are not connected to a strong edge.

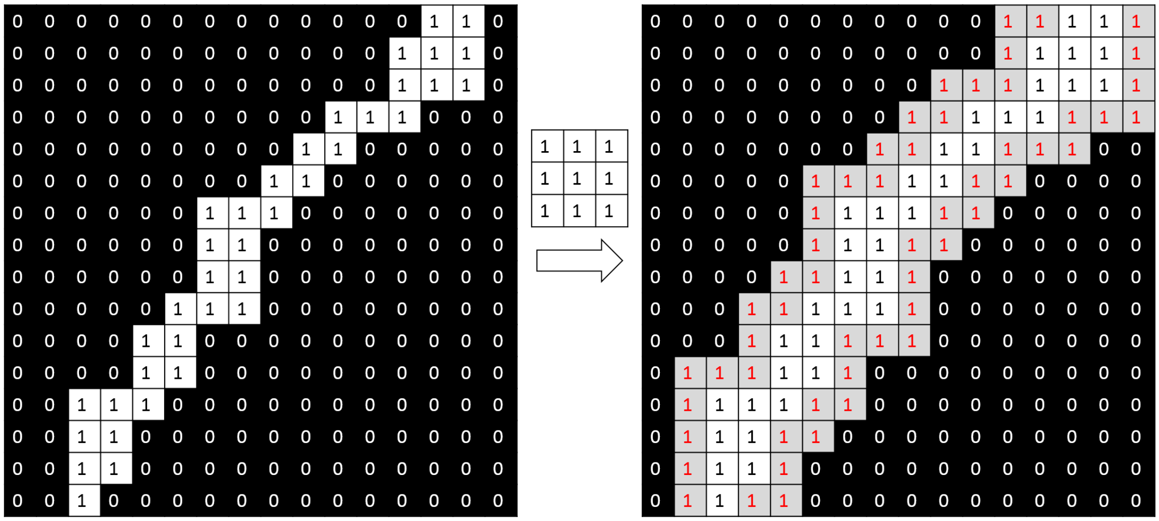
We have applied the edge detection algorithm with different values of σ to each octave of the proposed method. Applying Gaussian filter to the image will result in a blurred image, and edge detection on a blurred image using higher σ creates coarser edge segments, while on a blurred image using smaller σ tends to find noisy edge segments as well.

Since in our proposed method we are interested in finding longer and less noisy edges whenever possible, we first extract edge segments from the most smoothed image, the highest σ value, using the Canny edge detection method. In the next levels, we reduce the smoothing parameters to produce shorter and noisier edge maps.

As a result longer edges with more consistent lines are produced. If the image contains very small objects in the foreground, this parameter setting cannot find any edge in the image. To make the method more robust to objects with different sizes, we considered a set of σ values to extract edge segments.

This decision creates two difficulties to be handled. First, since a single edge in the image may be extracted as a long straight line in one smoothing level, and as a noisy curved in another, duplicate edges will be created from different smoothing levels. Second, sometimes a very small contrast between the foreground object and the background makes it impossible to extract edge maps from the original image.

To mitigate the first problem, we must keep the less noisy edge segment and ignore all of its duplications. Since we utilize various Gaussian Filter parameters, these



0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

1	1	1
1	1	1
1	1	1

0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0
0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0

Figure 4.2: The dilation process slides the filter over the image, and whenever the center of the filters hits a 1 all the pixels around it will turn to 1.

duplicated edges are not exactly in the same coordinate in the image for different edge maps, but they are very close to each other. To solve this concern, in each level we have dilated the edge map obtained from the previous level(s) with a $1_{3\times3}$ kernel to make the detected edges thicker (Figure 4.2). Dilation is a morphological operation on the image which slides a kernel K on the binary image P, i.e. the image's edge map, according to $P \oplus K = \bigcup_{p \in P} K_p$.

Using the cumulative edge map of all previous levels, P, and the detected edge map of the current level, D, the edge map obtained from the current level without their duplications are obtained using the logical operation of $N = \bar{P} \cdot D$.

At the end of each smoothing level, the cumulative edge map, P, is updated by adding the newly detected edges to the map using the logical operation of $P = P + N$. The general diagram of this procedure is represented in Figure 4.1.

To address the second problem, we sharpen the image at the level where we apply a very small σ, near to zero, for smoothing the image. The sharpening filter of Eq. (4.2) enhances the image's contrast and increases the likelihood of detecting edges around the foreground object in the image.

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \tag{4.2}$$

Table 4.1: Overall performance of the proposed method (EDGE) and the other well-known methods on two benchmark datasets using two different local descriptors.

Settings	Method	Accuracy	Precision	Recall	F-Measure
VOC 2007 SIFT	SIFT	88.91	28.24	28.49	28.37
	SURF	89.14	29.36	29.06	29.21
	ORB	88.26	23.88	23.91	23.89
	FAST	90.10	35.09	**33.41**	34.23
	EDGE	**91.33**	**42.01**	32.66	**36.75**
Settings	Method	Accuracy	Precision	Recall	F-Measure
VOC 2007 SURF	SIFT	88.33	25.26	26.24	25.74
	SURF	89.17	29.56	29.29	29.43
	ORB	87.94	23.71	25.44	24.54
	FAST	**91.31**	**41.35**	30.47	35.09
	EDGE	91.18	41.09	**33.22**	**36.74**
Settings	Method	Accuracy	Precision	Recall	F-Measure
VOC 2012 SIFT	SIFT	89.14	27.79	26.73	27.25
	SURF	89.19	27.95	26.70	27.31
	ORB	88.29	22.23	21.57	21.90
	FAST	89.88	33.02	32.14	32.58
	EDGE	**89.94**	**34.01**	**34.25**	**34.13**
Settings	Method	Accuracy	Precision	Recall	F-Measure
VOC 2012 SURF	SIFT	88.62	25.23	25.29	25.26
	SURF	89.23	28.73	28.11	28.42
	ORB	88.45	22.38	20.98	21.66
	FAST	91.49	42.14	**31.67**	36.16
	EDGE	**92.30**	**49.05**	29.62	**36.94**

After detecting edge maps in each octave and in each smoothing level, we utilize the Hough transform [54] to extract the edge segments from the obtained edge map. The Hough transform finds all instances of a specified shape in the image. Since in our proposed method the local descriptor provides the appropriate description for the area around each edge segment, considering complex shapes only increases the complexity of the proposed method without any additional value. As a result, we have utilized the Hough transform for finding the existing lines in the edge map.

4.2.2 Edge Tokens and Image Encoding

Edge tokens are defined as classes of common edge segments, or perceptual structure-based shapes obtained from Hough transform, that can describe each image. In the

Table 4.2: Time (miliseconds) comparison of the benchmarked methods on the entire training set of VOC 2007 and 2012 using SIFT descriptor.

Method	VOC 2007 SIFT	VOC 2012 SIFT
SIFT	352.13	1081.31
SURF	271.38	973.25
ORB	214.48	917.52
FAST	186.42	804.55
EDGE	146.4	631.23

proposed method, we have utilized a local image descriptor such as SIFT [101] to describe the area around each edge segment obtained from the previous steps. SIFT descriptor calculates gradients of the image pixels in a local patch around the keypoint and creates a numeric vector to describe that local region. We have calculated this description by assuming a point in the centre of each edge segment,(L_x, L_y), and describing the area whose length, A, is equal to the length of the edge segment, around that point which is rotated according to the edge segment's angle, θ, with respect to the image coordinate system. To localize each keypoint corresponding to the detected edge token, we need to find its coordinates, size, and orientation with respect to the image coordinate system whose notations are (L_x, L_y), A, and θ respectively. These characteristics are calculated according to Eq. (4.3) for an edge segment whose starting and ending points are (S_x, S_y), and (E_x, E_y).

$$\begin{aligned} &\forall w \in \{x, y\}, \quad L_w = S_w + \frac{E_w - S_w}{2} \\ &A = \sqrt{(E_x - S_x)^2 + (E_y - S_y)^2} \\ &\theta = \cos^{-1}\left(\frac{E_y - S_y}{(E_x - S_x) \times A}\right) \end{aligned} \tag{4.3}$$

All descriptors from the training set are collected and fed into the K-means clustering algorithm for finding edge tokens describing the current dataset. Each image in the dataset can be encoded using these edge tokens. To encode each image, all edge segments in that image are extracted and mapped to their corresponding edge tokens which are the cluster centers with the shortest $L2 - norm$ distance from the edge segment according to Eq. (4.4) in which X is the described edge token and C is the cluster center.

$$\|X - C\| = \sqrt{(X - C).(X - C)} \tag{4.4}$$

4.3 Experimental Results

We have tested our proposed method on the multi-label and multi-class image classification datasets of Pascal VOC 2007 test set (4952 images) [47], and VOC 2012 validation set (5823 images)[48]. In our experiments we utilized SIFT [101], and SURF [10] to describe the areas around the edge segments and created 500 edge tokens. We compared our proposed image representation method with other existing methods, such as SIFT [101], SURF [10], ORB [120], and FAST [119], using SIFT [101] and SURF [10] as their descriptors.

We have trained a Multi-Layer Perceptron network with two hidden layers of size 200, on 2501 and 5717 training images of VOC 2007 and 2012, with *ADAM* solver until it reached a maximum of 5000 iterations, or until its calculated loss in two consecutive epochs did not improve by 0.001. Our octaves resize factor was selected from $K \in \{1, 0, -1, -2\}$ and the standard deviations for Gaussian filters were $\sigma = 3, 1, 0.01$. The positive K values correspond to scales larger than the original image while the negative values resize the image to smaller sizes.

The performance metrics that are adapted in this work are categorized into two classes of overall performance, and per-class performance. For each of these categories the Accuracy, Precision, Recall, and F-Measure are presented. The equations for calculating the precision metric for these performance classes (O_p and P_p for overall and per-class precision) are shown in Eq. (4.5) and the same terminology is applied to the other metrics. In these equations, K is the number of classes in the dataset, N_k^c is the number of correctly predicted instances, and N_k^p is the total number of predicted samples per class k.

$$O_p = \frac{\sum_{k=1}^{K} N_k^c}{\sum_{k=1}^{K} N_k^p} \qquad\qquad P_p = \frac{1}{K} \sum_{k=1}^{K} \frac{N_k^c}{N_k^p} \tag{4.5}$$

The overall performance results are represented in Table 4.1 where in each column the method with the highest performance is bolded. Comparing these results shows the proposed method provides higher F-Measure for all different cases, where in some cases the accuracy, precision, or recall of the FAST method is slightly higher. These results prove that our proposed method provides a better trade-off between precision and recall compared to the other methods.

Besides performance comparison, we have performed time comparison among our proposed method and the other methods which are shown in Table 4.2. These results demonstrate that our proposed method is in the same scale as the others in terms of time complexity.

The per-class and each individual class performance are presented in Table 4.3 and Table 4.4. These results demonstrate the superiority of our proposed method against baseline methods in almost all metrics for two different datasets. Only for the recall on Pascal VOC 2007 the FAST keypoint detection method shows slightly higher performance, around 0.2% compared with our proposed method.

The interesting fact that is noticeable among the results in Table 4.3 and Table 4.4 is the superiority of the proposed method on classes with well-defined shapes, such as human-made objects like *aeroplane*, *boat*, *bus*, and *car*.

The only classes where some other methods show higher performances are animals, or plants whose shapes have many varieties and are not well-defined. Among these classes, our proposed method performance is very close to the best method except for the classes of *sheep* and *plants*.

The keypoints detected by different existing methods, and a single level of the proposed method are presented in Figure 4.3. The first row of this figure represents the weakness of methods such as SIFT and ORB in detecting keypoints where the lighting condition is not appropriate. These methods were not able to extract any keypoint in this image. SURF and FAST suggested keypoints that did not belong to the foreground object. On the other hand, keypoints associated to the edge segments of the proposed method are located on the object's boundary.

In the sample image with an small scale airplane as foreground, we have noticed that SIFT failed to detect keypoints on the foreground object which resulted in not being able to describe the image appropriately (Figure 4.3). On the other hand SURF was confused and detected the column in the image as foreground. ORB, FAST, and EDGE detected keypoints on the airplane, while FAST produced more keypoints in the background.

The frequency of the extracted keypoints from an image with a noisy background and noisy foregrounds shows how susceptible to noise the FAST algorithm is (Figure 4.3) . It is noticeable that FAST failed in detecting representative keypoints for foregrounds. SURF and SIFT suffered from the same problem with slightly more keypoints in the foreground, and less keypoints in the background. ORB produced the closest output to the proposed method, although it has also missed one of the foregrounds while detecting keypoints.

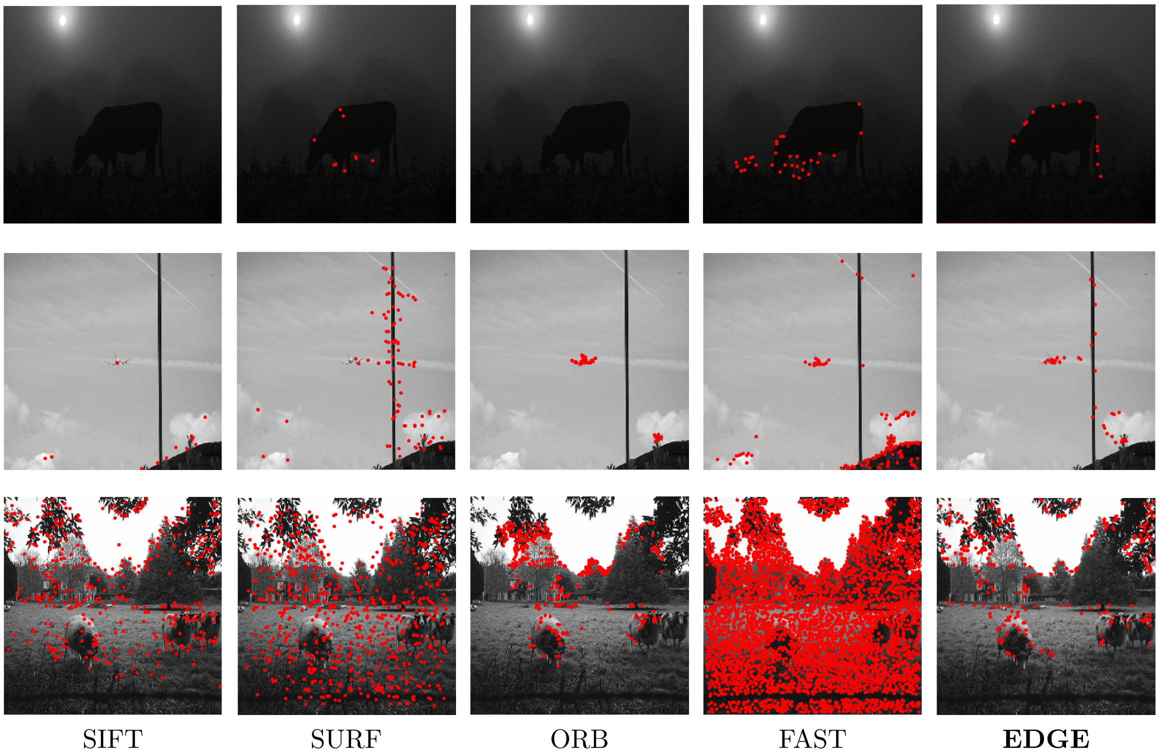


Figure 4.3: Qualitative comparison of the existing keypoint detection methods with the points associated to edge segments of the proposed method (EDGE) on images with low lighting condition, small object, and noisy background with noisy objects.

4.4 Summary

In this proposed method, we utilize the Canny edge detection algorithm to create octaves of edge maps for each image. These edge maps are divided into their constituent edge segments using the Hough transform. Center point of each edge segment is utilized as descriptive point and described using SIFT or SURF local descriptors. These local descriptions are clustered to create the Bag of Words which is then utilized for image representation. This algorithm produces higher performance compared to other man-made local representations in multi-class and multi-label datasets. Compared to N-gram based representation, this method detects perceptual elements and creates the bag of word automatically without any human supervision. This makes the method more scalable and robust to shape variation.

Table 4.3: Per-class performance and the performance for each individual classes of Pascal VOC 2007 test set. SIFT descriptor has been utilized for all methods.

VOC 2007	**Method**	**Aero**	**Bike**	**Bird**	**Boat**	**Bottle**	**Bus**	**Car**	**Cat**	**Chair**	**Cow**	**Table**	**Dog**	**Horse**	**Mbike**	**Person**	**Plant**	**Sheep**	**Sofa**	**Train**	**TV**	**MAP**
Accuracy	SIFT	94.5	92.0	90.7	94.5	89.4	93.0	76.6	89.4	82.7	95.2	91.3	85.0	91.7	92.2	60.7	89.4	96.9	87.1	90.5	91.0	88.7
	SURF	95.3	92.4	89.7	94.6	90.0	94.4	78.8	89.3	83.6	94.9	92.1	85.8	91.3	92.9	58.8	90.0	96.7	88.5	92.3	91.5	89.1
	ORB	94.3	91.1	89.4	94.4	89.4	93.7	76.6	88.0	81.2	95.3	91.1	84.7	90.4	92.7	56.2	90.1	**97.4**	87.5	91.7	90.3	88.3
	FAST	94.6	93.0	90.1	95.5	90.9	94.7	81.4	90.0	84.6	95.9	92.4	86.7	93.0	93.4	64.8	90.6	97.1	88.7	93.0	92.0	90.1
	EDGE	**96.1**	**94.4**	**91.5**	**96.3**	**93.1**	**95.4**	**81.8**	**91.4**	**86.4**	**96.5**	**93.6**	**88.5**	**93.8**	**94.7**	**65.7**	**93.2**	**97.4**	**90.1**	**94.0**	**93.0**	**91.3**
Precision	SIFT	28.9	17.8	19.1	15.5	7.7	10.2	26.5	21.1	23.4	7.8	14.9	16.0	24.4	16.5	53.3	**12.5**	9.1	16.0	14.0	17.3	18.6
	SURF	40.3	20.1	17.8	21.8	8.5	18.4	32.4	14.7	23.7	8.6	15.6	15.0	22.9	20.4	51.3	7.2	4.2	19.5	21.6	19.6	20.2
	ORB	29.7	7.1	13.5	11.8	6.4	12.6	24.8	11.5	16.7	4.3	11.5	11.1	16.5	10.3	48.3	5.0	9.8	15.3	16.8	13.6	14.8
	FAST	32.2	24.1	21.4	30.4	9.2	24.0	40.3	22.0	30.5	14.0	20.7	**21.7**	38.7	23.8	58.6	11.6	10.2	19.8	32.2	19.7	25.2
	EDGE	**55.4**	**37.8**	**25.6**	**45.5**	**13.0**	**30.3**	**42.7**	**29.3**	**36.6**	**18.4**	**27.1**	18.8	**43.3**	**37.9**	**60.8**	9.8	**20.0**	**28.7**	**40.7**	**27.1**	**32.4**
Recall	SIFT	22.0	16.4	18.3	12.5	**10.8**	11.5	27.7	**21.4**	25.3	7.9	15.8	16.9	22.6	16.3	**57.8**	**17.7**	6.1	18.9	15.8	19.6	19.1
	SURF	29.3	16.8	21.1	20.5	**10.8**	14.8	32.5	12.4	22.0	10.2	13.4	13.4	23.3	17.6	54.9	7.9	3.1	19.4	18.2	20.8	19.1
	ORB	28.3	6.4	15.2	9.1	8.8	12.0	24.4	11.8	17.8	3.9	11.7	10.6	17.2	7.3	48.7	5.1	4.1	16.3	15.1	16.5	14.5
	FAST	26.8	**18.0**	**26.3**	21.6	10.0	**20.2**	39.5	19.3	31.4	**11.8**	**18.6**	**19.9**	**41.6**	18.5	57.2	12.6	6.1	19.2	30.1	17.7	**23.3**
	EDGE	**35.1**	16.8	24.2	**22.7**	7.5	18.0	**47.7**	20.5	**32.8**	11.0	17.0	9.5	31.2	**21.5**	53.5	3.9	**11.2**	**25.4**	**30.5**	**21.6**	23.1
F-Measure	SIFT	24.9	17.1	18.7	13.8	9.0	10.8	27.1	21.3	24.3	7.8	15.3	16.4	23.5	16.4	55.5	**14.7**	7.3	17.3	14.9	18.4	18.7
	SURF	33.9	18.3	19.3	21.1	**9.5**	16.4	32.5	13.4	22.8	9.4	14.4	14.2	23.1	18.9	53.0	7.5	3.6	19.5	19.7	20.2	19.5
	ORB	29.0	6.7	14.3	10.3	7.4	12.3	24.6	11.6	17.2	4.1	11.6	10.9	16.8	8.5	48.5	5.1	5.8	15.8	15.9	14.9	14.6
	FAST	29.3	20.6	23.6	25.3	9.6	22.0	39.9	20.6	31.0	12.8	19.6	**20.7**	**40.1**	20.8	**57.9**	12.1	7.6	19.5	31.1	18.6	24.1
	EDGE	**43.0**	**23.3**	**24.9**	**30.3**	**9.5**	**22.6**	**45.1**	**24.1**	**34.6**	**13.8**	**20.9**	12.6	36.3	**27.4**	56.9	5.6	**14.4**	**26.9**	**34.9**	**24.0**	**26.6**

Table 4.4: Per-class performance and the performance for each individual classes of Pascal VOC 2012 validation set. SIFT descriptor has been utilized for all methods.

VOC 2012	**Method**	**Aero**	**Bike**	**Bird**	**Boat**	**Bottle**	**Bus**	**Car**	**Cat**	**Chair**	**Cow**	**Table**	**Dog**	**Horse**	**Mbike**	**Person**	**Plant**	**Sheep**	**Sofa**	**Train**	**TV**	**MAP**
Accuracy	SIFT	92.6	91.5	89.9	93.2	88.4	95.0	84.5	86.2	83.5	95.9	91.2	83.1	93.0	92.4	61.8	91.2	95.5	90.0	92.7	91.6	89.1
	SURF	93.4	92.9	89.4	92.7	88.6	95.3	83.1	85.6	84.0	95.3	91.4	83.9	92.5	92.8	60.7	91.3	95.2	90.3	92.6	92.9	89.2
	ORB	92.9	90.5	88.8	92.4	88.8	94.0	82.0	84.6	81.9	**96.0**	91.0	82.4	93.0	92.1	57.0	90.5	95.2	89.9	91.9	91.2	88.3
	FAST	93.8	92.3	**90.1**	93.9	88.2	95.4	85.0	**86.5**	84.3	95.6	91.7	83.9	**93.8**	92.7	64.9	**91.7**	**95.9**	90.9	94.0	93.0	89.9
	EDGE	**94.8**	**93.0**	**90.1**	**94.3**	**89.2**	**96.0**	**85.8**	86.4	**85.2**	95.6	**92.1**	**84.9**	93.5	**93.2**	**66.0**	91.6	95.5	**91.5**	**94.2**	**93.5**	**90.3**
Precision	SIFT	37.4	17.6	21.0	20.5	8.9	30.2	22.9	25.3	22.8	9.4	17.1	22.7	11.4	13.8	50.1	6.6	11.0	15.4	21.6	18.2	20.2
	SURF	44.1	24.3	18.8	15.0	11.2	35.1	21.0	22.4	23.9	4.6	15.4	23.3	13.2	23.3	48.6	6.6	9.7	14.1	23.8	29.2	21.4
	ORB	39.3	8.8	11.9	12.9	9.7	20.1	14.0	16.8	17.6	8.5	14.6	18.6	8.3	14.5	43.8	6.0	4.4	11.3	12.6	14.0	15.4
	FAST	48.4	23.0	22.7	27.3	13.5	37.0	27.9	**28.4**	27.9	11.8	24.4	25.5	**24.9**	23.2	54.3	**11.4**	**22.7**	16.8	36.1	30.6	26.9
	EDGE	**57.7**	**27.0**	**25.3**	**32.0**	**15.3**	**45.2**	**33.0**	27.5	**32.4**	**11.9**	**26.3**	**30.5**	24.2	**26.0**	**55.9**	10.3	16.6	**24.8**	**38.2**	**34.6**	**29.7**
Recall	SIFT	35.1	19.3	20.9	19.8	8.9	29.4	20.6	24.5	20.9	6.5	15.2	20.3	9.8	13.4	50.3	6.5	9.7	16.4	21.1	18.6	19.3
	SURF	39.9	20.3	19.5	14.7	11.7	34.6	22.5	21.7	20.6	3.9	12.4	18.5	13.9	26.3	47.1	6.1	9.7	13.4	25.5	28.4	20.5
	ORB	33.3	9.7	11.5	13.1	9.2	21.8	14.1	16.4	17.5	5.2	13.0	16.3	6.5	15.7	43.6	6.8	3.9	11.0	12.0	14.2	14.7
	FAST	46.3	**23.5**	22.7	25.0	**16.0**	37.0	27.5	**29.0**	26.6	**10.4**	23.2	21.9	23.3	27.5	52.8	**10.8**	**22.6**	14.6	33.8	29.1	26.2
	EDGE	**50.9**	**23.5**	**27.5**	**29.0**	15.5	**44.6**	**35.0**	27.9	**31.6**	**10.4**	**23.5**	**26.0**	**25.3**	**27.9**	**53.7**	9.7	17.4	**23.8**	**36.0**	**32.1**	**28.6**
F-Measure	SIFT	36.2	18.4	20.9	20.2	8.9	29.8	21.7	24.9	21.8	7.7	16.1	21.4	10.6	13.6	50.2	6.5	10.3	15.9	21.3	18.4	19.7
	SURF	41.9	22.1	19.2	14.9	11.4	34.8	21.8	22.0	22.1	4.2	13.7	20.6	13.5	24.7	47.9	6.3	9.7	13.7	24.6	28.8	20.9
	ORB	36.1	9.2	11.7	13.0	9.4	20.9	14.1	16.6	17.5	6.5	13.8	17.4	7.3	15.1	43.7	6.4	4.1	11.1	12.3	14.1	15.0
	FAST	47.3	23.2	22.7	26.1	14.6	37.0	27.7	**28.7**	27.3	11.0	23.8	23.6	24.1	25.2	53.5	**11.1**	**22.7**	15.6	34.9	29.8	26.5
	EDGE	**54.1**	**25.1**	**26.4**	**30.4**	**15.4**	**44.9**	**34.0**	27.7	**32.0**	**11.1**	**24.8**	**28.1**	**24.8**	**26.9**	**54.7**	10.0	17.0	**24.3**	**37.1**	**33.3**	**29.1**

Chapter 5

Image Representation Using Bag of Perceptual Curve Features [1]

5.1 Introduction

Although the proposed hybrid method (Chapter 4) has shown great performance on the benchmark datasets, it does not consider the perceptual characteristics of the human vision system. On the other hand, the N-gram based method has rule-based dictionary which reduces from its generalization to all possible classes of edge segments combinations. In this research, we are combining both of these features and proposed our Bag of CPP method.

Our proposed method improves the PCPG model [72] by generalizing the joint detection module and applying the laws of Gestalt to group perceptual structure-based edge segments. In this method, we have considered direction changes as a result of sign or magnitude change in the edge gradient which identifies the Generic Edge Tokens (GETs). We have grouped these GETs based on their proximities, and their slope and curvature similarities, while preserving the continuity of the edge traces.

In the proposed method, the joint Curve Partitioning Points (CPPs) connect groups of GETs and are utilized as descriptive points for the image. These CPPs are described and clustered to create a Bag of CPPs (BoC) which contains the representatives for different groups of similar CPPs in our training set. Each image is encoded according to this BoC by calculating its Normalized Curve Histogram in all levels of the Spatial Pyramid Matching [92].

We have evaluated our proposed method in single-label, and multi-label classification scenarios on four datasets of Caltech 101 [51], Caltech 256 [62], Pascal VOC 2007 [47], and Pascal VOC 2012 [48]. The obtained results from the proposed method

[1] The contents of this chapter is partially published in [42].

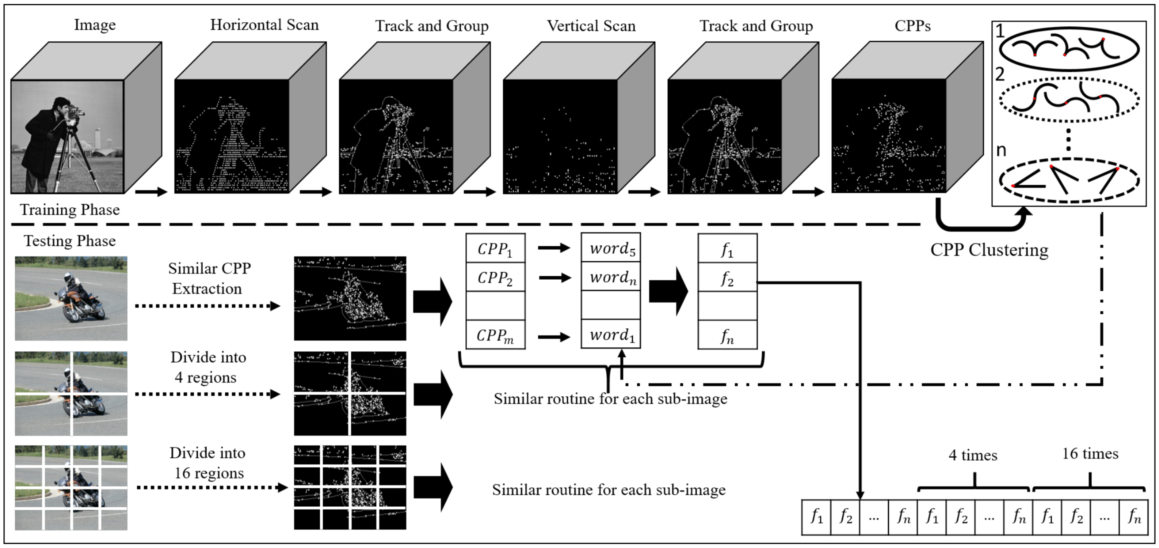


Figure 5.1: The main diagram of the proposed method. The top part illustrates the routine for generating Bag of CPPs from the training set. The bottom part is image encoding method using the Bag of CPPs and Spatial Pyramid Matching.

outperform the benchmarked local image representation methods.

Following in this Chapter, we have elaborated our proposed method in Section 5.2 and our evaluation settings and experimental results on some of the challenging datasets are discussed in Section 5.3. Finally we summarize this chapter in Section 5.4.

5.2 Proposed Method

We have proposed a hybrid image representation method by creating semantically rich keypoints, Curve Partitioning Points (CPPs), and describing them using an existing local descriptor. These CPPs are clustered to define a Bag of CPPs where each cluster center is the representative of a group of similar CPPs in the dataset. We have applied a hard coding technique for describing the image in various levels of a spatial pyramid. These codes are combined to create the final representation for the image. The overall diagram of the proposed method is depicted in Figure 5.1.

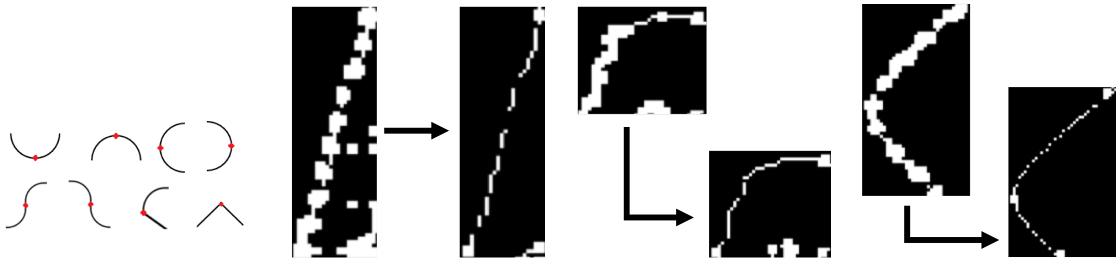

(a) (b) (c) (d)

Figure 5.2: CPPs (a) and application of proximity (b), similarity (c), and continuity (d) rules.

5.2.1 Perceptual Keypoints

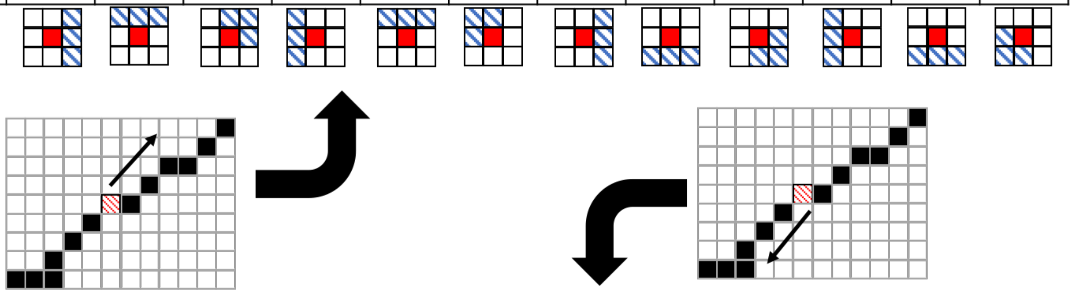
Gestalt Psychology suggests that human vision perception relies mostly on special relationships among its observations [28]. Based on this psychology, the human vision system groups its observations according to several characteristics, such as their proximity, similarity, and continuity. In this research we have applied these rules for grouping Generic Edge Tokens (GETs) of the image where the joint CPPs are utilized for image representation. GETs are the perceptual constituents of an image's edge map which carry shape-related information, such as the slope and curvature of the edge.

To create this edge map, we have horizontally and vertically scanned the image in a predefined interval to find the initial points for tracking the edge pixels. In the scanning process, each point's gradients are calculated according to Eq. 5.1 where (x, y) are the coordinates of the current pixel, and I is the image's intensity function. If any of these gradients of a certain point is greater than a predefined threshold, that point is selected as a candidate initial point.

$$dx = \sum_{i=-2}^{2} I_{(x+i,y)} + \sum_{i=-1}^{1} I_{(x+i,y)} - 2I_{(x,y)} \tag{5.1}$$
$$dy = \sum_{i=-2}^{2} I_{(x,y+i)} + \sum_{i=-1}^{1} I_{(x,y+i)} - 2I_{(x,y)}$$

Starting from each initial point, based on the signs and values of its dx and dy, three neighbor pixels of the current point are chosen as candidate points for continuing the edge tracking (see tables in Figure 5.3). For each of these candidates, the gradients dx and dy are calculated and the one with maximum gradient is selected as the next

$\boldsymbol{dx}$	+	+	+	+	+	+	−	−	−	−	−	−
$\boldsymbol{dy}$	+	+	+	−	−	−	+	+	+	−	−	−
$\boldsymbol{\frac{dy}{dx}}$	> 2	$< \frac{1}{2}$	$> \frac{1}{2}$ < 2	< -2	$> -\frac{1}{2}$	$< -\frac{1}{2}$ > -2	< -2	$> -\frac{1}{2}$	$< -\frac{1}{2}$ > -2	> 2	$< \frac{1}{2}$	$> \frac{1}{2}$ < 2



$\boldsymbol{dx}$	+	+	+	+	+	+	−	−	−	−	−	−
$\boldsymbol{dy}$	+	+	+	−	−	−	+	+	+	−	−	−
$\boldsymbol{\frac{dy}{dx}}$	> 2	$< \frac{1}{2}$	$> \frac{1}{2}$ < 2	< -2	$> -\frac{1}{2}$	$< -\frac{1}{2}$ > -2	< -2	$> -\frac{1}{2}$	$< -\frac{1}{2}$ > -2	> 2	$< \frac{1}{2}$	$> \frac{1}{2}$ < 2

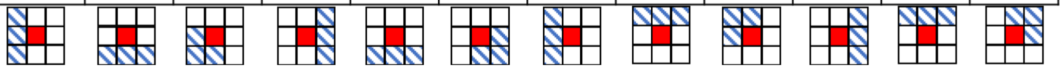


Figure 5.3: The edge tracking routine. Starting from the initial point (textured red) the algorithm follows two different directions. Based on the gradient values of the current point (solid red), three neighbors are selected as candidate tracking points (textured blue).

point, constrained to having a greater gradient than the predefined threshold. This edge tracking process continues until we reach one of the three untraceable conditions: reaching an already-visited edge point, the image's borders, or a candidate point that does not satisfy the gradient constraint.

Since each initial point may be located in the middle of a trace, the edge tracking routine must be done in two different directions from the initial point, where a trace is a connected set of edge pixels. This process is depicted in Figure 5.3 where for each tracking direction, based on the signs of gradients and their values' relationship a different set of candidate points are investigated.

For each edge point, we must examine its features to find whether it is a candidate CPP. For this purpose, we investigate two commonly observed scenarios for change in the edge's direction which are backed by our candidate point generation routine.

These are changes in either signs or values of gradients when moving from the current point to the new point along an edge trace. These conditions are formalized in Eq. (5.2) where dx_C and dy_C are gradients for the current point, and dx_N and dy_N are the gradients for the new point.

$$(dx_C \times dx_N) < 0 \quad or \quad (dy_C \times dy_N) < 0 \tag{5.2}$$

$$\frac{|dx_C - dx_N|}{|dy_C - dy_N|} > 0 \quad and \quad \frac{|dy_C - dy_N|}{|dy_C - dx_N|} > 0 \tag{5.3}$$

Edge pixels between two consecutive CPPs (Figure 5.2(a)) form a GET which is classified into one of the eight groups of Table 2.1 according to its slope and curvature. Each GET carries meaningful shape information in the image and perceptually describes the image's content. The slope, S, and curvature, C, of each GET is calculated using formulas in Eq. (5.4) where (x_i, y_i) are the coordinates of the point in the start, s, middle, m, or end, e, of the GET.

$$S = \frac{y_e - y_s}{x_e - x_s} \quad C = \frac{|x_m - x_s| - |x_e - x_m|}{|y_m - y_s| - |y_e - y_m|} \tag{5.4}$$

The candidate curve partitioning points contain many noise CPPs which must be removed for being compliant with the Gestalt Laws of grouping. Based on the proximity rule, the human vision perception groups items in a close distance. To apply this rule, if two CPPs are close enough, their distance is less than a predefined threshold, we have merged them and created a single CPP which joins their connected GETs (Figure 5.2(b)). The similarity rule implies that the human vision system groups similar objects when they are perceived. To apply this rule, for any two consecutive GETs, if their slope and curvatures are similar, we have merged them together and removed their connecting CPPs (Figure 5.2(c)). The continuity rule specifies that the human vision system groups items that are connected to each other. This rule's application on the proposed method removes any extra CPP which does not separate different classes of GETs (Figure 5.2(d)).

5.2.2 Encoding using Bag of Curve Partitioning Points

Although CPPs carry perceptual information, they suffer from lack of specificity. There are many similar CPPs in an image resulting in a redundant and weak representation. To solve this issue, we have clustered CPPs using Kmeans clustering

Table 5.1: Overall performance comparison on Pascal VOC 2007, Pascal VOC 2012, Caltech 101 and Caltech 256 with 15 and 30 training images.

Settings	Method	Accuracy	Precision	Recall	F-Measure
VOC 2007	SIFT	88.91	28.24	28.49	28.37
	ORB	88.26	23.88	23.91	23.89
	CPP	**89.77**	**33.25**	**32.43**	**32.84**
VOC 2012	SIFT	89.14	27.79	26.73	27.25
	ORB	88.29	22.23	21.57	21.90
	CPP	**89.71**	**31.72**	**30.63**	**31.17**
Caltech 101 15	SIFT	28.55	25.29	28.31	25.73
	ORB	30.08	28.33	29.90	28.18
	CPP	**41.49**	**40.12**	**40.61**	**39.60**
Caltech 101 30	SIFT	35.00	29.54	33.27	29.28
	ORB	37.20	32.93	36.34	32.40
	CPP	**49.43**	**43.30**	**46.68**	**43.27**
Caltech 256 15	SIFT	10.73	9.05	10.73	9.52
	ORB	9.52	8.44	9.53	8.70
	CPP	**15.18**	**13.97**	**15.18**	**14.18**
Caltech 256 30	SIFT	14.16	12.12	14.16	12.74
	ORB	11.80	10.48	11.80	10.88
	CPP	**18.06**	**16.81**	**18.06**	**17.11**

algorithm [104] and created a Bag of CPPs to be used for stronger and less noisy application-specific image representation. For this purpose, we have described the area around each CPP using a local patch descriptor such as SIFT [101]. These described CPPs are fed into the Kmeans algorithm and, a single CPP is determined as the representative for a group of CPPs belonging to any specific cluster.

To encode each image using this BoC, the vector quantization technique [61] has been utilized and the frequency of each CPP in the BoC is calculated for the image and a histogram of frequencies is created where each bin corresponds to a specific CPP in BoC. This histogram is normalized and utilized as image representation.

The obtained histogram suffers from a lack of location information for CPPs, yet this is an important source of information for image representation. To address this drawback, we have adapted the Spatial Pyramid Matching method to find final representation of the image. In this technique, the image has to be divided into gradually smaller sub-images, for each of which a frequency histogram for the entire BoC is generated. These histograms are concatenated to create the final representation of

Table 5.2: Time (miliseconds) comparison of the benchmarked methods on the entire training set of VOC 2007 and 2012 using SIFT descriptor.

Method	VOC 2007 SIFT	VOC 2012 SIFT
SIFT	352.13	1081.31
ORB	214.48	917.52
CPP	190.4	857.9

the image.

5.3 Evaluation

In this section we discuss the details of our evaluation method. Our implementation details, and evaluation settings followed by the evaluation metrics that were adapted are discussed in 5.3.1. Finally in 5.3.2 we have presented our results and provided discussion on the effectiveness of the proposed method over the other methods.

5.3.1 Experimental Settings

We have utilized Python 3.6 for implementing the proposed method, while we chose OpenCV 3.0 implementation for SIFT [101], and ORB [120]. We have selected the minimum gradient threshold for an edge pixel being 80 and a 10-pixels interval was chosen for horizontal and vertical scanning of the image to find initial tracking points. We have chosen the proximity and neighborhood threshold to be 5 pixels. To select these parameters, one must consider the nature of underlying application. If the algorithm is sensitive to very tiny edges, the gradient, interval, and neighborhood thresholds must be reduced and vice versa. These values are considered for a moderate sensitivity.

Multi-Label Classification: Pascal VOC 2007 [47] and 2012 [48] are selected as multi-label datasets for evaluation in which we have utilized the K-means [104] clustering algorithm for creating Bag of Visual Words and Bag of CPPs with the size of 500 words. The image encoding of Section 5.2.2 is performed by creating a normalized frequency histogram whose bins correspond to various words in the BoW. These representations are fed into a Multi-Layer Perceptron Neural Network [121] with two hidden layers with the size of 200. This classifier is trained using ADAM solver [87] with the constant learning rate of 0.001 on the training images.

The performance metrics that are adopted for this setting are categorized into two classes of overall, and per-class performances (Accuracy, Precision, Recall, and F-Measure). The precision metrics, O_p and P_p, for overall and per-class precision are formalized in Eq. (5.5). This terminology has been used for the other metrics. In these equations, K is the number of classes in the dataset, N_k^c is the number of correctly predicted instances, and N_k^p is the total number of predicted samples from class k.

$$O_p = \frac{\sum_{k=1}^{K} N_k^c}{\sum_{k=1}^{K} N_k^p} \qquad P_p = \frac{1}{K}\sum_{k=1}^{K}\frac{N_k^c}{N_k^p} \tag{5.5}$$

Single-Label Classification: The Mini Batch Kmeans clustering algorithm [125] is utilized for creating the Bag of Visual Words (Bag of CPPs) whose size is 1024 words. The images are encoded using Spatial Pyramid Matching with three layers of $\{1, 2, 4\}$. For each sub-image, a normalized frequency histogram is generated and all of them are concatenated to create the final image representation. The detail of image encoding is described in Section 5.2.2. The image representations are fed into a Linear SVM classifier [18] with $C = 10$ using the *One versus Rest* strategy for multi-class classification.

We have utilized Caltech 101 [51] and Caltech 256 [62] for evaluating this setting in which we have calculated the performance metrics such as precision, recall, accuracy, and F-measure for the entire dataset. Showing the per-class evaluation is not feasible because of the number of classes and is meaningless because of single label images. The precision metric is calculated according to Eq. (5.5) for the overall precision and the other performance metrics follow the same rule.

5.3.2 Experimental Results

We have compared the overall Accuracy, Precision, Recall, and F-Measure of the proposed method (CPP) with two other image representation methods based on SIFT, and ORB keypoints. The proposed method has gained a performance of approximately 5% higher than the closest competitor in terms of precision, recall, and F-measure, while its accuracy improvement is around 1% for multi-label classification (Table 5.1). On the other hand, it has outperformed the other methods by about 10% in the case of single-label classification.

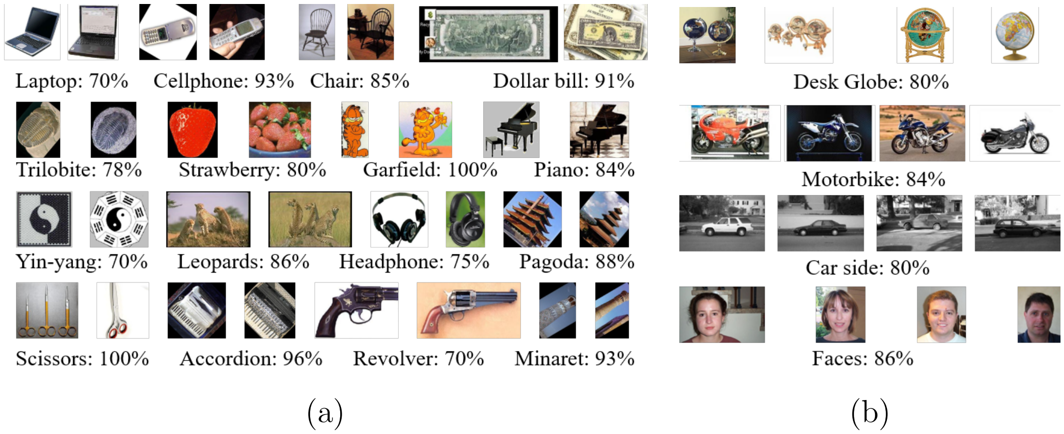


Figure 5.4: (a) Sample images from classes of Caltech 101 dataset with *accuracy* >= 70%. (b) Sample images from classes of Caltech 256 dataset with *accuracy* >= 80%.

Besides performance evaluation, we have compared our proposed method with the benchmarked methods and presented the results in Table 5.2. These results demonstrate that the proposed method takes approximately the same time as the existing compared methods.

We have also compared the per-class metrics for Pascal VOC 2007, and 2012 datasets. Almost in all classes of these datasets, the proposed method outperforms the rest and these results are presented in Tables 5.3 and 5.4. The only exceptions are image categories of nature, such as sheep and plant where the proposed method is marginally overcome by the SIFT or ORB methods. The probable reason for this behavior is the noisiness of the shape of objects in nature.

On the other hand, for classes of human-made objects such as *airplane*, *bottle*, and *bus* the proposed method shows more than 10% improvement. The sample images of some classes in Caltech 101, and Caltech 256 whose accuracies are higher than 70% or 80% respectively are shown in Figure 5.4. These results also illustrate the great performance of the proposed CPP method on images with hand-made objects such as *laptops*, *cellphones*, and *cars*.

5.4 Summary

In this chapter, we have proposed a perceptual image representation based on the shape of objects which is its core feature according to the human vision perception. Besides, we have relied on Gestalt psychology laws to group the generic edge tokens obtained from the image's perceptual edge map into less noisy and more general tokens. We have considered the proximity, similarity, and continuity rules for this grouping and extracted the points that connect every two groups of edge tokens. Those points are the more descriptive areas of the image and are utilized for creating a general Bag of CPPs used for image representation. This method has shown superior performance compared with the baseline methods. Compared to the hybrid representation method, the BOC method finds more robust keypoints because of applying the Gestalt laws of grouping. This reduces the sensitivity to noise and can be applied specifically for human made objects with well-structured shapes.

Table 5.3: Per-class performance and the performance for each individual classes of Pascal VOC 2007 test set.

VOC 2007	**Method**	**Aero**	**Bike**	**Bird**	**Boat**	**Bottle**	**Bus**	**Car**	**Cat**	**Chair**	**Cow**	**Table**	**Dog**	**Horse**	**Mbike**	**Person**	**Plant**	**Sheep**	**Sofa**	**Train**	**TV**	**MAP**
Accuracy	SIFT	94.5	92.0	**90.7**	94.5	89.4	93.0	76.6	89.4	82.7	95.2	91.3	85.0	91.7	92.2	60.7	89.4	96.9	87.1	90.5	91.0	88.7
	ORB	94.3	91.1	89.4	94.4	89.4	93.7	76.6	88.0	81.2	95.3	91.1	84.7	90.4	**92.7**	56.2	90.1	**97.4**	87.5	91.7	90.3	88.3
	CPP	**94.6**	**92.5**	90.1	**95.1**	**90.5**	**94.2**	**81.7**	**89.9**	**85.8**	**96.2**	**93.4**	**85.4**	**92.5**	**92.7**	**60.8**	**90.4**	97.2	**88.5**	**92.9**	**91.3**	**89.8**
Precision	SIFT	28.9	17.8	19.1	15.5	7.7	10.2	26.5	**21.1**	23.4	7.8	14.9	16.0	24.4	16.5	53.3	**12.5**	9.1	16.0	14.0	**17.3**	18.6
	ORB	29.7	7.1	13.5	11.8	6.4	12.6	24.8	11.5	16.7	4.3	11.5	11.1	16.5	10.3	48.3	5.0	**9.8**	15.3	16.8	13.6	14.8
	CPP	**31.3**	**19.6**	**19.6**	**22.5**	**8.5**	**18.0**	**40.9**	20.6	**34.4**	**13.4**	**28.3**	**16.1**	**31.6**	**20.6**	**53.5**	11.1	9.6	**21.2**	**30.4**	16.9	**23.4**
Recall	SIFT	22.0	**16.4**	18.3	12.5	**10.8**	11.5	27.7	**21.4**	25.3	7.9	15.8	**16.9**	22.6	16.3	57.8	**17.7**	**6.1**	18.9	15.8	**19.6**	19.1
	ORB	**28.3**	6.4	15.2	9.1	8.8	12.0	24.4	11.8	17.8	3.9	11.7	10.6	17.2	7.3	48.7	5.1	4.1	16.3	15.1	16.5	14.5
	CPP	25.4	16.0	**22.5**	**15.3**	10.0	**15.9**	**38.3**	17.8	**31.7**	**8.7**	**21.5**	15.9	**29.0**	**19.3**	**58.0**	12.6	5.1	**22.3**	**27.8**	17.7	**21.5**
F-Measure	SIFT	24.9	17.1	18.7	13.8	9.0	10.8	27.1	**21.3**	24.3	7.8	15.3	**16.4**	23.5	16.4	55.5	**14.7**	**7.3**	17.3	14.9	**18.4**	18.7
	ORB	**29.0**	6.7	14.3	10.3	7.4	12.3	24.6	11.6	17.2	4.1	11.6	10.9	16.8	8.5	48.5	5.1	5.8	15.8	15.9	14.9	14.6
	CPP	28.0	**17.6**	**21.0**	**18.2**	**9.2**	**16.9**	**39.6**	19.1	**33.0**	**10.5**	**24.4**	16.0	**30.3**	**20.1**	**55.7**	11.8	6.7	**21.7**	**29.0**	17.3	**22.3**

Table 5.4: Per-class performance and the performance for each individual classes of Pascal VOC 2012 validation set.

VOC 2012	**Method**	**Aero**	**Bike**	**Bird**	**Boat**	**Bottle**	**Bus**	**Car**	**Cat**	**Chair**	**Cow**	**Table**	**Dog**	**Horse**	**Mbike**	**Person**	**Plant**	**Sheep**	**Sofa**	**Train**	**TV**	**MAP**
Accuracy	SIFT	92.6	91.5	89.9	93.2	88.4	95.0	84.5	86.2	83.5	95.9	91.2	**83.1**	**93.0**	92.4	61.8	91.2	**95.5**	90.0	92.7	91.6	89.1
	ORB	92.9	90.5	88.8	92.4	88.8	94.0	82.0	84.6	81.9	**96.0**	91.0	82.4	**93.0**	92.1	57.0	90.5	95.2	89.9	91.9	91.2	88.3
	CPP	**94.0**	**92.4**	**90.2**	**93.7**	**88.9**	**95.8**	**85.5**	**86.4**	**84.5**	95.4	**92.0**	82.7	92.9	**93.4**	**62.3**	**91.6**	**95.5**	**90.8**	**93.9**	**92.4**	**89.7**
Precision	SIFT	37.4	17.6	21.0	20.5	8.9	30.2	22.9	**25.3**	22.8	9.4	17.1	**22.7**	11.4	13.8	50.1	6.6	11.0	15.4	21.6	18.2	20.2
	ORB	39.3	8.8	11.9	12.9	9.7	20.1	14.0	16.8	17.6	8.5	14.6	18.6	8.3	14.5	43.8	6.0	4.4	11.3	12.6	14.0	15.4
	CPP	**49.7**	**21.8**	**23.9**	**22.7**	**12.6**	**42.2**	**27.9**	**25.3**	**30.0**	**11.1**	**23.2**	21.0	**19.3**	**25.1**	**50.7**	**12.6**	**13.4**	**19.3**	**34.9**	**24.3**	**25.6**
Recall	SIFT	35.1	19.3	20.9	**19.8**	8.9	29.4	20.6	**24.5**	20.9	6.5	15.2	**20.3**	9.8	13.4	50.3	6.5	9.7	16.4	21.1	18.6	19.3
	ORB	33.3	9.7	11.5	13.1	9.2	21.8	14.1	16.4	17.5	5.2	13.0	16.3	6.5	15.7	43.6	6.8	3.9	11.0	12.0	14.2	14.7
	CPP	**50.0**	**20.0**	**24.1**	19.4	**12.7**	**41.2**	**24.7**	23.2	**30.2**	**10.4**	**18.9**	19.1	**21.6**	**24.1**	**52.2**	**12.5**	**12.9**	**18.8**	**34.6**	**23.7**	**24.3**
F-Measure	SIFT	36.2	18.4	20.9	20.2	8.9	29.8	21.7	**24.9**	21.8	7.7	16.1	**21.4**	10.6	13.6	50.2	6.5	10.3	15.9	21.3	18.4	19.7
	ORB	36.1	9.2	11.7	13.0	9.4	20.9	14.1	16.6	17.5	6.5	13.8	17.4	7.3	15.1	43.7	6.4	4.1	11.1	12.3	14.1	15.0
	CPP	**45.5**	**20.9**	**24.0**	**20.9**	**12.7**	**41.7**	**26.2**	24.2	**30.1**	**10.7**	**20.8**	20.0	**20.4**	**24.6**	**51.4**	**12.6**	**13.2**	**19.0**	**34.7**	**24.0**	**24.9**

Chapter 6

Object Localization by using Generic Edge Features to Optimize Convolutional Neural Network Detection Scores [1]

6.1 Introduction

Object recognition is the task of finding all the objects in an image along with their x and y locations in the image. This means that object recognition consists of two major steps of object detection and object localization as discussed in Chapter 2. Because of the strength of deep learning techniques in representing images, and their recent progress in classifying images with similar or even better accuracy compared with human, object detection task is almost solved. Despite this progress, the object recognition still has room for improvement in the object localization task.

In this research, an object localization method is introduced which relies on the fact that objects in the image have corresponding edge segments in the edge map of the image. This method applies a Best First Search algorithm [122] on the edge segments around the candidate objects, which are detected by an object detection module, one at a time. In each iteration, the current candidate object is merged with all its overlapping edge segments, and the detection score for each merged box is obtained by feeding the Convolutional Neural Network (CNN) representation into an SVM classifier. The merged box with the maximum score is selected as an improved candidate object and is fed into the next iteration. This routine continues until there is no more edge or no more improvement. The main flowchart of the proposed method is represented in Figure 6.1.

The main difference of the proposed method from the current object localization methods such as bounding box regression [59] is its independence from any information about the training dataset. This means the proposed method solely relies on the information obtained from the current image. This feature creates the ability to

[1] The contents of this chapter is partially published in [43, 41].

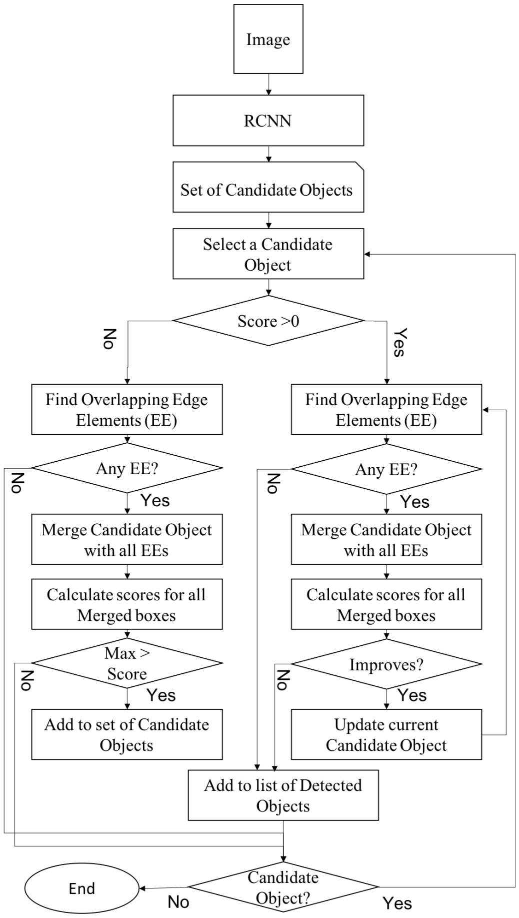


Image
RCNN
Set of Candidate Objects
Select a Candidate Object
Score >0
No
Yes
Find Overlapping Edge Elements (EE)
Any EE?
Yes
No
Merge Candidate Object with all EEs
Calculate scores for all Merged boxes
Max > Score
Yes
No
Add to set of Candidate Objects
Find Overlapping Edge Elements (EE)
Any EE?
Yes
No
Merge Candidate Object with all EEs
Calculate scores for all Merged boxes
Improves?
Yes
No
Update current Candidate Object
Add to list of Detected Objects
Candidate Object?
End
No
Yes

Figure 6.1: Main flowchart of the proposed object localization method.

apply the proposed method to the applications where a trained network exists, and the training data either does not exist or is expensive to obtain.

This chapter is organized in the following order: The proposed method is explained in Section 6.2. First the detection score (Sub-section 6.2.1) is defined, then the candidate object detection module is discussed in Sub-section 6.2.2 followed by finding overlapping edge segments (Sub-section 6.2.3) and merging them with the candidate object (Sub-section 6.2.4). At the end the optimization algorithm (Sub-section 6.2.5) is elaborated. The evaluation and discussion on the performance of the proposed method are illustrated in Section 6.3. Section 6.4 concludes this chapter along with some possible areas of future work and the limitations of the proposed method.

6.2 Proposed Method

The proposed object localization method is applied to the result of the object detection method which is a set of candidate objects with their corresponding types and detection scores in the image. It modifies the locations, (x, y) coordinates, of these candidate objects to improve their detection scores for their corresponding types, or for other types of objects with higher detection scores.

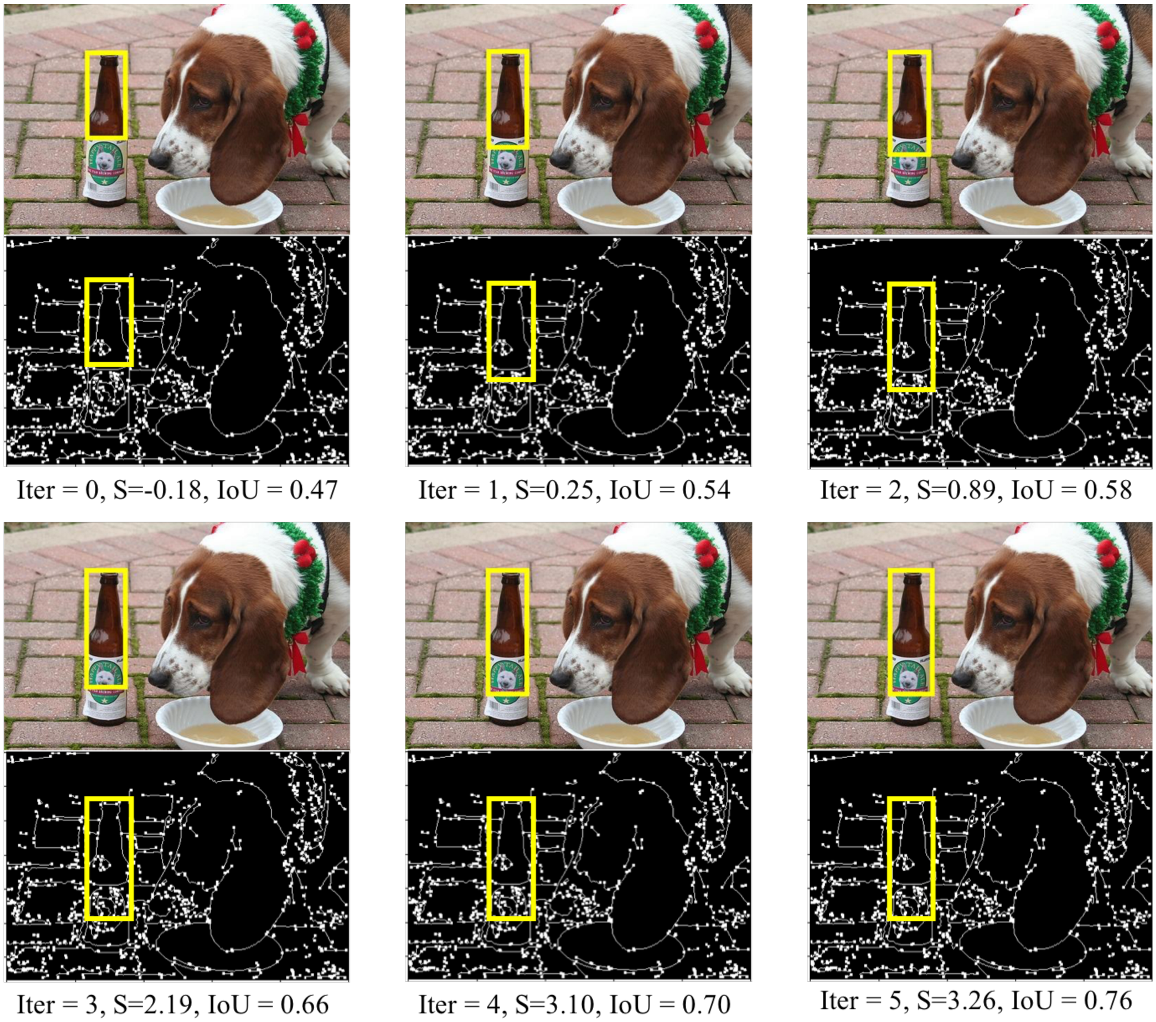
The proposed method (Figure 6.1) applies a Best First Search on the set of edge segments, which represents the object boundaries in the image. The search space for this method is edge segments of the image extracted from its edge map. This method searches for the locations of objects in the image where the detection score of CNN is maximized. The improved candidate object in each iteration is a modified box whose detection score is higher than the original one, and is the input to the next iteration of the searches. This search continues until the improvement is stabilized or there is no more edge segment with positive overlap with the current candidate box. Figure 6.2 shows several iterations of GET_Loc and its positive impact on the object localization.

6.2.1 Calculate Detection Score

Calculating score for an area in an image is the base module in the proposed object localization method which is illustrated in this section. For the purpose of score calculation, the feature vector that represents the specified area in the image is obtained, normalized and fed into a classifier, all of these steps is elaborated in this section.

For finding the feature vector for the specified area of the image, it is scaled to the suitable size for the selected Convolutional Neural Network (CNN). The scaling technique which is used in this research is image warping using the bilinear interpolation method. The bilinear interpolation is a method for resampling that uses the distance weighted average of four nearest pixel values to estimate a new pixel value [2] as is represented in Figure 6.3a.

The warped image using the bilinear interpolation is fed into a CNN network for extracting the representation for the specified area in the image. This feature vector has to be normalized to be ready for using in the SVM classification. This



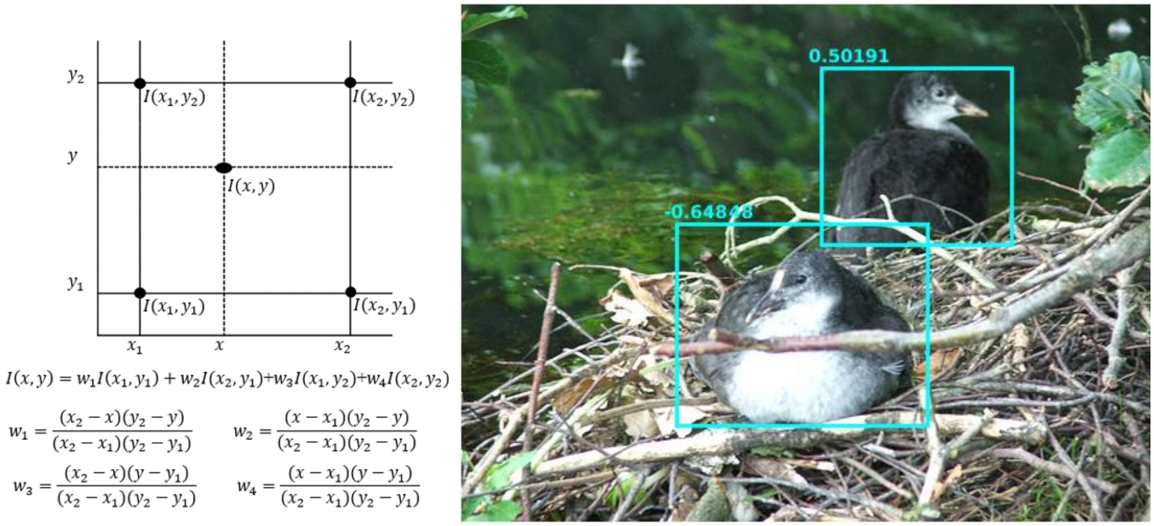
Iter = 0, S=-0.18, IoU = 0.47
Iter = 1, S=0.25, IoU = 0.54
Iter = 2, S=0.89, IoU = 0.58
Iter = 3, S=2.19, IoU = 0.66
Iter = 4, S=3.10, IoU = 0.70
Iter = 5, S=3.26, IoU = 0.76

Figure 6.2: Improved bounding boxes after several iterations of the Best First Search on GETs. The detection score and IoU with ground truth object has improved.

normalization is done using the Eq. (6.1). In this equation, T is the training set, N is its size and C is a constant value.

$$NormalizedFeature = C \times \frac{Feature}{\frac{1}{N} \times \sum_T Feature} \tag{6.1}$$

The last step in calculating the detection score of an area is feeding the normalized feature into a classifier. In the proposed method, we have used the Liblinear [50] classifier for this purpose. The optimization equation of this classifier is represented in Eq. (6.2). In this equation, $f \subseteq \mathcal{F}$ is the matrix of feature vectors, $l \subseteq \mathcal{L}$ is the vector of labels, and K is a reqularization weight.


y_2
y
y_1
$I(x_1, y_2)$
$I(x_2, y_2)$
$I(x, y)$
$I(x_1, y_1)$
$I(x_2, y_1)$
x_1
x
x_2
$I(x, y) = w_1 I(x_1, y_1) + w_2 I(x_2, y_1) + w_3 I(x_1, y_2) + w_4 I(x_2, y_2)$
$w_1 = \frac{(x_2 - x)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)}$
$w_2 = \frac{(x - x_1)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)}$
$w_3 = \frac{(x_2 - x)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)}$
$w_4 = \frac{(x - x_1)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)}$
0.50191
-0.64848

(a)

(b)

Figure 6.3: (a) The bilinear interpolation resampling method. (b) RCNN with correct object and negative score.

$$w = \min_{\acute{w}} \sum_{(f,l)\in T} \ell(\acute{w}; (f, l)) + Kr(\acute{w}) \tag{6.2}$$

For using the liblinear SVM classifier, we have used the ℓ_2 regularized - ℓ_1 hinge loss setting for the regularizer and the loss function [106]. This setting specifies the loss function as $\ell(w; (f, l) = \max(0, 1 + \max_{\acute{l}\neq l} \sum_{f\in F} w(f, \acute{l}) - \sum_{f\in F} w(f, l))$ and the regularizer as $r_2(w) = \sum_{f\in F} \sum_{l\in \mathcal{L}} w^2(f_1, l)$.

When the classifier has trained, the obtained weights of the classifier are used for finding the detection score ($\varphi(A, T)$) of the specified area according to the Eq. (6.3). In this equation, $f(A)$ is the normalized feature of CNN for specified area A, $w(T)$ is the weights trained in Liblinear SVM, and $b(T)$ is a predefined bias, both corresponding to class T, which Liblinear handle it by adding a dimension to its feature vector and weight matrix [50].

$$\varphi(A, T) \quad = \quad f(A) \times w(T) + b(T) \tag{6.3}$$

6.2.2 Candidate Object Detection

Before applying the object localization method, it is required to find candidate objects in the image. For this purpose, we use the object detection method similar to RCNN. This method, uses the Selective Search(SS) algorithm [137] to find a number of bounding boxes that potentially contain objects. The SS algorithm finds superpixels of the image and merge them hierarchically to come up with larger areas in the image.

The bounding boxes obtained from the SS algorithm are fed into the module of Section 6.2.1 for calculating their detection scores. In this step, the bounding boxes whose scores are greater than a specified threshold τ are chosen as detected objects and the others would be ignored.

At the end, since it is unpleasant to have multiple bounding boxes around a single object, a Non-Maximum Suppression method (NMS) is applied to the bounding boxes of detected objects. This technique selects the one with the highest detection score among all the bounding boxes whose overlaps are greater than a specified threshold ν and belong to similar object types. This technique is represented in Figure 6.4. The metric that is used for measuring the overlap between two bounding boxes is Intersection over Union (IoU) which for boxes A and B is calculated using the Eq. (6.4).

$$IoU(A, B) = \frac{Area(A \cap B)}{Area(A \cup B)} \tag{6.4}$$

We have performed an investigation on the detection scores of the ignored bounding boxes to determine the threshold of candidate objects generation module. The

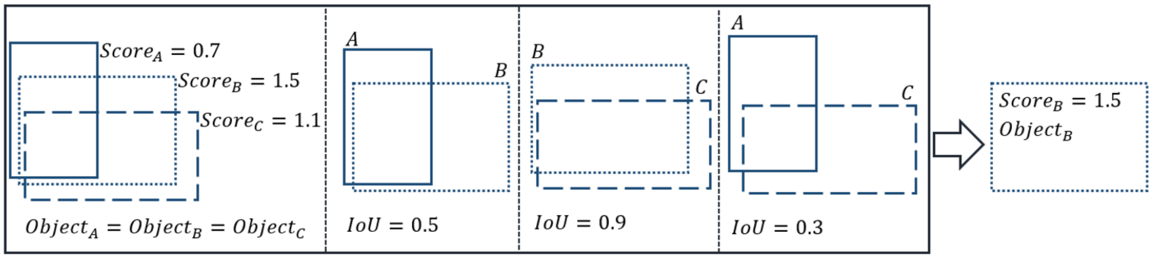
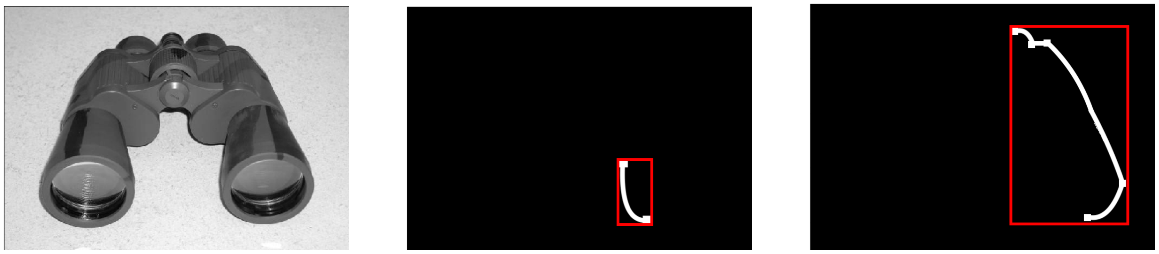


Figure 6.4: The general diagram of the NMS technique in keeping the objects bounding boxes.



(a) Image (b) GET (c) Trace

Figure 6.5: Sample Image, GET, and Trace. The area around each edge segment is represented with a red box.

results show that there are many bounding boxes with the negative scores that contain objects and this happens because of their inadequate localizations. Figure 6.3b shows a bounding box with negative score that is ignored from detection.

While Object detection module of RCNN keeps the bounding boxes with the positive score ($\tau > 0$) as detected objects, in the proposed method we chose a negative threshold τ for determining the candidate objects in each image. The bounding boxes whose scores are greater than τ form the set of candidate objects for further improvement in their localizations.

6.2.3 Edge Segments

The edge map of an image contains information about the boundary of objects in that image. This information is a beneficial source for improving the object localization, specially when the training images are not available. Using the PCPG package [56], the edge map of the image are obtained and its traces and GETs are classified (Figure 6.5). The detail explanation on the PCPG method is elaborated in Section 2.3.2.

Each trace obtained from PCPG is a group of connected edge points tracked from a single starting point (Figure 6.5). Each trace has a starting point and an ending point that specify the bounding box around it.

These traces are investigated to find the points where the curvature of the edge changes, Curve Partitioning Points (CPPs). The curves between CPPs form the set of GETs of the image. Similar to traces in the image, GETs are used as edge segments of object boundaries for improving the object localization. Each GET has a bounding

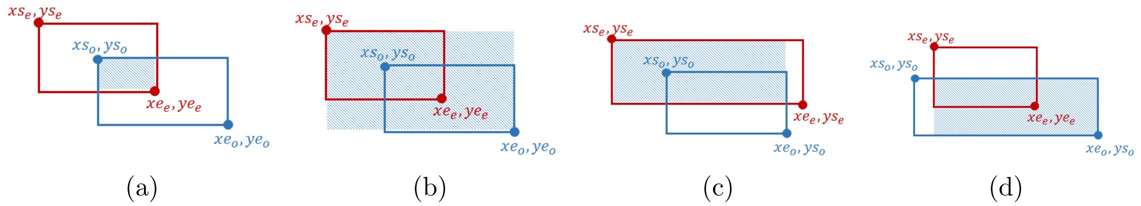


Figure 6.6: Merge a candidate object with an overlapping edge segment. (a) Equation 6.5, (b) Equation 6.6, (c) Equation 6.7, (d) Equation 6.8

box around it which is defined using its starting and ending points.

These edge segments provide guidance for better object localization. In one side by considering the traces as search space, the area around the candidate object is investigated with the larger step size while using the GETs as search space provides smaller step size. In this research, we tested three different sets of edge segments as search space for more precise localization. We have a set of Trace edge segments, a set of GET edge segments, and a set consists of both types of edge segments to have a variety of step sizes. We named each version of the proposed method, which are different in their search space, by refering to the type of its edge segments and defined TraceLoc, GETLoc, and GT(GET and Trace)Loc.

6.2.4 Merge Candidate Object with Edge Segments

For each bounding box in the set of candidate objects, all of the edge segments whose overlaps with that candidate object are greater than zero ($IoU > 0$) are selected for merging. Each of these edge segments is merged with the candidate object in four different ways, according to Figure 6.6 and Algorithm 6.1, and four different merged boxes are created for a single edge segment.

For creating these merged boxes, four different scenarios are considered which are shown in Figure 6.6 and elaborated here respectively. In all of these equations $[(xs_o, ys_o), (xe_o, ye_o)]$ is the coordination of the candidate object, $[(xs_e, ys_e), (xe_e, ye_e)]$ is the coordination of edge segment, and $[(xs_m, ys_m), (xe_m, ye_m)]$ is the coordination of the merged box.

The first scenario is the case that the object exists in the intersection area of the candidate object and the edge segment. In this case the coordination of the merged

Algorithm 6.1. Merge candidate object with the overlapped edge segments

```
procedure FindMergedBoxes($(CandidBox, EdgeMap)$)
    ▷ Input: Candidate object
    ▷ Input: List of edge segments in the edge map
    ▷ Output: List of merged bounding boxes
    $CandidObj = [(xs_o, ys_o), (xe_o, ye_o)]$
    $Edgesegment = [(xs_e, ys_e), (xe_e, ye_e)]$
    $MBox = [\ ]$
    for Each Edge segment $i$ do
        ▷ First Merged Box
        Calculate $[(xs_m, ys_m), (xe_m, ye_m)]$ according to Equation 6.5
        $MBox = [MBox; [(xs_m, ys_m), (xe_m, ye_m)]]$
        ▷ Second Merged Box
        Calculate $[(xs_m, ys_m), (xe_m, ye_m)]$ according to Equation 6.6
        $MBox = [MBox; [(xs_m, ys_m), (xe_m, ye_m)]]$
        ▷ Third Merged Box
        Calculate $[(xs_m, ys_m), (xe_m, ye_m)]$ according to Equation 6.7
        $MBox = [MBox; [(xs_m, ys_m), (xe_m, ye_m)]]$
        ▷ Fourth Merged Box
        Calculate $[(xs_m, ys_m), (xe_m, ye_m)]$ according to Equation 6.8
        $MBox = [MBox; [(xs_m, ys_m), (xe_m, ye_m)]]$
        $MBox = [MBox; MBS_{X,Y}, MBE_{X,Y}]$
```

box is calculated using Eq. (6.5).

$$\begin{aligned}(xs_m, ys_m) &= \max(xs_o, xs_e), \max(ys_o, ys_e) \\ (xe_m, ye_m) &= \min(xe_o, xe_e), \min(ye_o, ye_e)\end{aligned} \tag{6.5}$$

The second scenario, is the merged box that contains both the candidate object and the edge segment entirely and its coordination is calculated using Eq. (6.6).

$$\begin{aligned}(xs_m, ys_m) &= \min(xs_o, xs_e), \min(ys_o, ys_e) \\ (xe_m, ye_m) &= \max(xe_o, xe_e), \max(ye_o, ye_e)\end{aligned} \tag{6.6}$$

Considering the situation that the candidate object contains the entire object along with some extra area of the image, defines the third scenario in merging the candidate object with the edge segment. This merged box is calculated using the Eq. (6.7).

$$\begin{aligned}(xs_m, ys_m) &= \min(xs_o, xs_e), \min(ys_o, ys_e) \\ (xe_m, ye_m) &= \min(xe_o, xe_e), \min(ye_o, ye_e)\end{aligned} \tag{6.7}$$

The last scenrio occures when the third scenario exists in one hand and the candidate object does not have the entire object as well. For this situation, we use the Eq. (6.8) to calculate the coordinates of the merged box.

$$\begin{aligned}(xs_m, ys_m) &= \max(xs_o, xs_e), \max(ys_o, ys_e) \\ (xe_m, ye_m) &= \max(xe_o, xe_e), \min(ye_o, ye_e)\end{aligned} \tag{6.8}$$

6.2.5 Optimization

For improving the precision of objects locations in the images, the Best First Search (BFS) algorithm has been utilized in this research. The main method of the proposed method is represented in Algorithm 6.2. BFS is an informed heuristic tree-based search method which in each iteration chooses the node which is closest to the search's goal. This method defines an evaluation function which determines how close is the current node to the goal [122]. To adapt the BFS method into the object localization task, the search space, objective function, and the finishing conditions should be specified.

In the application of object localization, we are looking for bounding boxes that determine the location of the candidate object more precisely. This means that our search space is a set o bounding boxes around that candidate object which may provide higher precision. For creating this search space, in each iteration, the edge segments whose bounding boxes have overlap with the candidate object are selected for creating the search space.

In Figure 6.6, a candidate object with coordination of $[xs_o, ys_o, xe_o, ye_o]$ is combined in four different ways with an edge segment with a coordination of $[xs_e, ys_e, xe_e, ye_e]$ whose overlap with the candidate object is positive. As a result, for each overlapping edge segment, a set of four merged boxes is created. The shaded areas in Figure 6.6 represent these merged boxes.

Algorithm 6.2. Object localization using the Generic Edge Tokens of the image

```
procedure GETLoc($Image, CanObj$)
    ▷ Input: Image
    ▷ Input: List of candidate boxes with their detection scores
    ▷ Output: List of detected boxes with their detection scores
    for $Each\ CandidBox_i$ do
        while $Detection\ Score\ Improves$ do
            $FindMergedBoxes(CandidBox, EdgeMap)$
            for $Each\ Merged\ Box\ j$ do
                ▷ Calculate Detection Score $DS_{i,j}$
                $DS_{i,j} = CNNScore(MergedBox_j)$
            ▷ Find the best merged box
            $SelectedBox = arg\ max_{j \in MergedBox} DS_{i,j}$
            $CandidBox_i = SelectedBox$
```

All the merged boxes from all the edge segments whose overlaps with the current candidate object is greater than zero create the search space for finding a more precise location. For each segment in this search space, each merged box, the evaluation function must be calculated. We have chosen the detection score of each merged box as an evaluation metric. The higher detection score, the higher confidence in detecting object, and the better location around the candidate object. As a result, the goal of the BFS search is to find the maximum detection score in each iteration of the search. Defining $\varphi(C, T)$ as the detection score for an area inside the bounding box C to be from class T, the optimization problem of each iteration is represented in Eq. (6.9).

$$\varphi(B, T_o) = arg \max_{C_e \in E} \varphi(C_e, T_i) \tag{6.9}$$

In this equation, C_e is a merged box obtained from merging the current candidate object with an edge segment from the edge map E. The result of this optimization in each iteration (or level in the search tree) is a modified candidate object's bounding box B, its class type T_o, and its detection score $\varphi(B, T_o)$. This modified candidate object is fed into the next iteration of the BFS algorithm for further improvements.

In search for improving each candidate object, two different cases may occur. As first case, if the candidate objects score is positive, the BFS search iterates on the merged boxes around that to improve its detection score. This search continues until there is no more edge segment with overlap, the search space is empty, or the detection score does not improve in several iterations.

As another case, if the candidate object has a negative score, we search among all merged boxes around it to find a bounding box with positive score. If this kind of bounding box has found, the candidate object is replaced with the merged box with the highest positive score. Eventually, this modified candidate object will endure the BFS search of the first case. Otherwise, if there is no merged box with positive score, this candidate object is ignored as it probably does not have any recognizable object.

6.3 Experimental Results

We have done some experiments on the proposed method to evaluate its performance. In this section, the framework for these experiments is defined and an extensive amount of results are represented. In defining the framework, we introduce the datasets that we have evaluated on, the parameter settings that we have used, and the task and measurement that we have chosen for reporting our results. Then the obtained results on this framework are represented and discussed along with comparing the outputs of the proposed method with ground truth and the baseline method on some test images. Finally, some interesting observations that motivated us for future areas of improvements are represented.

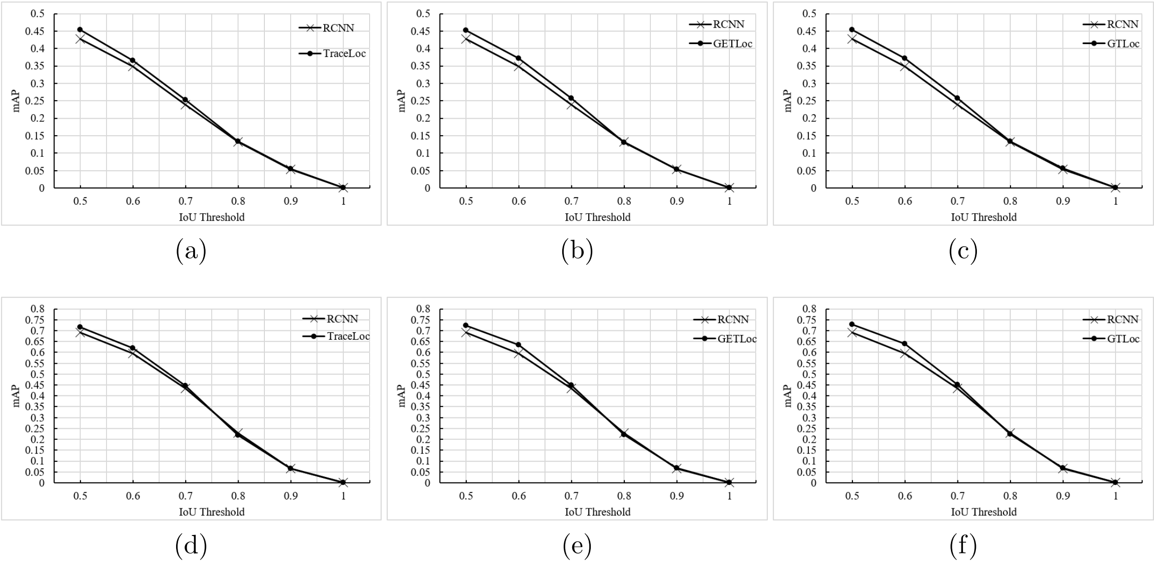
6.3.1 Experimental Framework

We have performed our evaluation on three datasets of Pascal VOC 2007 [47] test set, Pascal VOC 2012 [48] test set, and Pascal VOC 2007 validation set. These are the standard datasets for Pascal VOC competition that was held annually from 2005 to 2012 [49].

This competition contained different computer vision tasks such as image classification, object detection, image segmentation, action recognition, and person layout; from which we have used object detection competition for our evaluation. The submitted methods for this competition have to predict the bounding boxes of each object

Table 6.1: Comparison of mAP for different classes of (a) Pascal VOC 2007 test set (b) Pascal VOC 2012 test set, and (c) Pascal VOC 2007 validation set using the baseline RCNN model and different versions of the proposed model.

(a) Test 2007	**Aero**	**Bike**	**Bird**	**Boat**	**Bottle**	**Bus**	**Car**	**Cat**	**Chair**	**Cow**	**Table**	**Dog**	**Horse**	**Mbike**	**Person**	**Plant**	**Sheep**	**Sofa**	**Train**	**TV**	**MAP**
RCNN	49.8	**61.7**	32.8	25.2	24.2	**53.1**	61.5	49.0	22.8	48.8	33.2	39.4	51.4	51.5	48.4	15.6	**50.2**	35.0	**49.5**	51.2	42.7
GET_Loc	49.5	60.9	37.7	31.0	30.3	51.2	61.4	**54.4**	27.8	**53.7**	32.6	46.1	57.5	58.4	48.5	**20.8**	48.2	34.1	47.9	51.6	45.2
Trace_Loc	50.3	61.3	**39.8**	31.6	30.8	51.9	61.9	48.6	28.9	47.6	**34.3**	**47.1**	**58.7**	59.6	48.5	20.6	49.3	**35.5**	49.0	**52.2**	45.4
GT_Loc	**50.4**	61.3	39.4	**31.8**	**32.0**	52.3	**62.0**	48.9	**29.2**	47.8	33.3	46.8	58.4	**59.6**	**48.6**	20.7	48.4	35.4	49.2	51.9	**45.4**
(b) Test 2012	**Aero**	**Bike**	**Bird**	**Boat**	**Bottle**	**Bus**	**Car**	**Cat**	**Chair**	**Cow**	**Table**	**Dog**	**Horse**	**Mbike**	**Person**	**Plant**	**Sheep**	**Sofa**	**Train**	**TV**	**MAP**
RCNN	56.4	49.3	31.4	15.4	19.4	43.3	46.1	52.4	13.6	31.9	23.8	48.7	41.1	51.8	44.0	12.8	42.9	20.4	33.7	34.4	35.6
GET_Loc	**59.2**	52.7	**35.5**	**18.8**	22.7	46.0	49.0	55.1	17.2	**38.1**	26.4	51.3	**44.5**	53.8	47.0	**14.9**	44.7	**23.3**	**38.3**	**39.1**	**38.9**
Trace_Loc	58.4	**53.3**	35.2	**18.8**	22.5	46.5	48.6	54.9	16.6	37.8	25.8	**51.9**	43.7	**54.5**	**47.3**	13.8	44.3	22.2	37.8	38.4	38.6
GT_Loc	58.8	52.8	35.0	18.7	**23.1**	**46.8**	**49.1**	**55.2**	**17.5**	37.8	**26.5**	51.4	44.4	54.1	47.1	14.7	**45.3**	23.1	**38.3**	**39.1**	**38.9**
(c) Val 2007	**Aero**	**Bike**	**Bird**	**Boat**	**Bottle**	**Bus**	**Car**	**Cat**	**Chair**	**Cow**	**Table**	**Dog**	**Horse**	**Mbike**	**Person**	**Plant**	**Sheep**	**Sofa**	**Train**	**TV**	**MAP**
RCNN	**81.1**	80.1	70.2	53.7	43.0	71.2	**71.3**	**80.1**	61.5	**81.3**	62.7	**81.1**	81.6	80.5	50.7	33.8	**70.8**	72.7	**81.5**	72.6	69.1
GET_Loc	80.0	80.2	79.0	70.0	47.8	71.0	71.0	79.1	**66.9**	81.3	71.4	80.0	80.1	79.5	**56.9**	**41.7**	69.4	80.3	79.7	**81.4**	72.3
Trace_Loc	79.5	80.1	69.7	**70.4**	**50.8**	**71.3**	70.7	79.5	58.8	80.7	**71.6**	79.6	80.5	79.6	56.4	40.5	70.3	**80.7**	79.5	81.2	71.6
GT_Loc	79.6	**80.5**	**79.5**	70.2	48.6	71.1	71.0	79.2	60.0	80.7	71.4	80.1	**88.3**	**87.6**	56.5	40.9	70.0	80.3	79.2	81.2	**72.8**



(a) (b) (c)

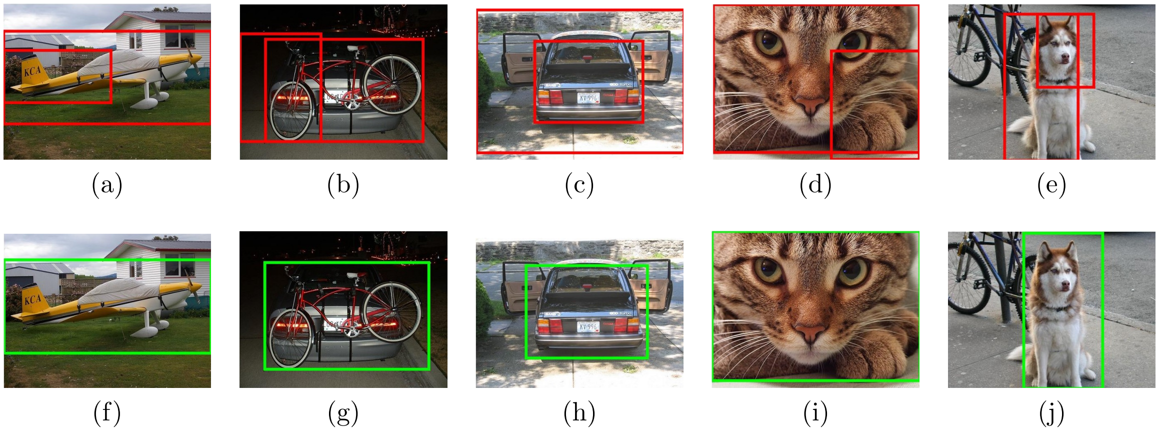
(d) (e) (f)

Figure 6.7: Compare mAP of the proposed methods and the baseline model RCNN for different IoU thresholds on Pascal VOC 2007 test set and validation set. (a), (b), (c) represent the comparison of Trace_Loc, GET_Loc, and GT_Loc on test set, while (d), (e), and (f) show the same result on validation set.

from any type if it exists in the test image, with a confidence value for this prediction.

All of these datasets contain 20 classes of object types with annotation files for images in the dataset. The annotation file for each image contains the ground truth bounding boxes around all the objects in that image. Pascal VOC 2007 contains 9,963 images and 24,640 annotated objects. These images are divided into three sets of train, validation, and test with 2501, 2510, and 4952 images respectively. Pascal VOC 2012 is a larger dataset with 11,530 images for train (5717) and validation (5823) and 10991 images in the test set. This dataset has 27,450 annotated objects just in its train and validation images.

Our experimental results utilize two measurement of mean Average Precision (mAP) and IoU for the purpose of evaluation, as they are norm among researchers in the object recognition field. The mAP is calculated using the Eq. (6.10) and calculates average precision for each class of objects and the final mAP would be the mean value of all the average precisions of classes.



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 6.8: RCNN baseline model detects multiple bounding boxes around a single object which some samples of that are represented in (a), (b), (c), (d), and (e). The output of the proposed GT_Loc method is also represented in the second row showing that this issue is fixed.

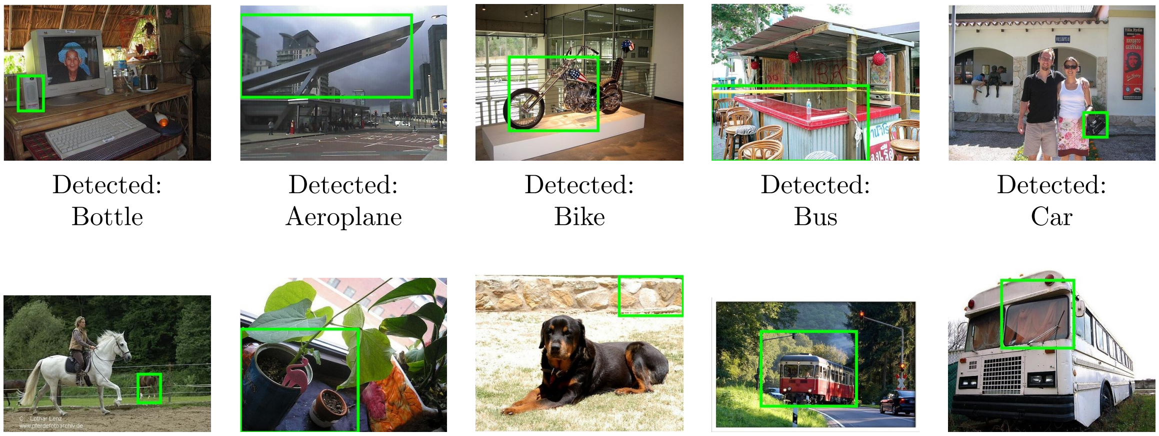
$$
\begin{aligned}
AP &= \frac{number \quad of \quad detected \quad objects}{total \quad number \quad of \quad objects} \\
mAP &= \frac{\sum_N AP}{N}, \quad N = number \quad of \quad classes
\end{aligned}
\tag{6.10}
$$

We also calculate the IoU (Eq. (6.4)) between the bounding box around the detected object and the ground truth bounding box in the annotation file. Using this metric, we are able to compare the effectiveness of our proposed method in finding bounding boxes closer to the bounding boxes annotated by human and evaluate our strength in improving object localization.

For evaluating the proposed method, we have selected the RCNN [59] with AlexNet Convolutional Neural Network [90] as a baseline model in all the experiments. We have used Caffe [80] toolbox for training and implementing the proposed method. To find the edge segments, we have used the PCPG [56] package with the gradient threshold of 10, scanning interval of 8, and minimum edge length of 11 pixels.

6.3.2 Comparison

We compared the performance of our proposed methods, such as RCNN with GET localization (GET_Loc), RCNN with trace localization (Trace_Loc), and RCNN with



Detected: Bottle

Detected: Aeroplane

Detected: Bike

Detected: Bus

Detected: Car

Detected: Cow

Detected: MotorBike

Detected: Sheep

Detected: Train

Detected: TV Monitor

Figure 6.9: Samples of objects whose types are detected wrongly.

GET and trace localization (GT_Loc), with RCNN as a baseline method. The Average Precision (AP) of the proposed methods and the baseline for different classes of these datasets, along with the mAP for the entire datasets are represented in Table 6.1.

By comparing the overall mAP of the proposed method and the baseline model in these datasets, an improvement of approximately 3% is noticeable. The results show that the proposed method has improved the precision significantly for classes such as *'bird'*, *'boat'*, *'bottle'*, *'chair'*, *'dog'*, *'horse'*, *'motorbike'*, and *'plant'* where the edge information is precise in the images. This is concluded from the improved mAP of around 10% for these classes.

The results of Table 6.1 show that the proposed methods can improve mAP for the Pascal VOC 2007 validation set as well in which the RCNN baseline model represents its best performance, since the images in this validation set are used for finetuning the image representations. To conclude this paragraph, the proposed methods are able to improve the best performance of the RCNN baseline model, without having any knowledge about the undergoing image.

We compared the mAP of the proposed and the baseline methods for different threshold values of IoU and represented the resulted diagrams in Figure 6.7. These diagrams illustrate that the proposed methods are more precise compared with the RCNN baseline method since in any of the overlap thresholds the proposed method has a higher average precision. This observation occures since our proposed method not only improves the localization of objects detected by RCNN, but also detects objects missed by RCNN because of their poor localizations.

Some examples of the objects detected by GT_Loc are shown in Figure 6.12 along with the outputs of the baseline RCNN model, and the ground truth annotation. These are some of the samples that RCNN has ignored due to their negative detection score, while the proposed method has improved their locations' precision which resulted in their positive detection scores and yield to detection by the proposed method. Besides this much improvement, the proposed method takes about 5 seconds to process each object which is the major drawback of the proposed method that should be improved in the future.

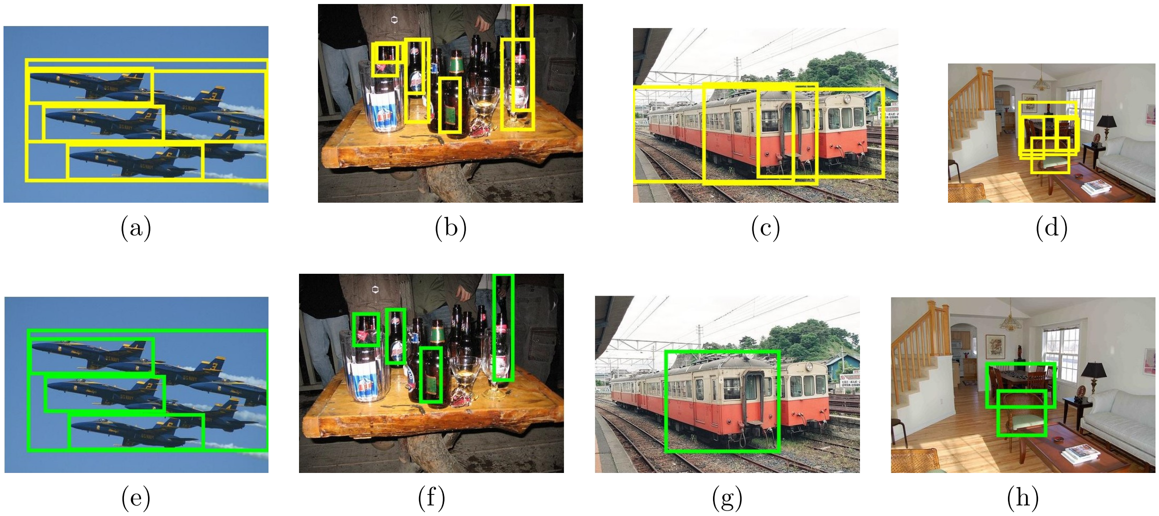
Our experiments show that the RCNN baseline model finds several bounding boxes around a single object which is because of their inefficient localization. The proposed GT_Loc method has solved this issue where it is due to insufficient precision of localization. Some sample images from this matter are represented in Figure 6.8.

Some images from the Pascal VOC 2007 test set are represented in Figure 6.11 along with the bounding boxes around their objects which is produced by the proposed GT_Loc method. These images show that the proposed method is able to detect multiple instance of an object in a single image, and also is able to find different objects in the image. The last row of this figure, shows the multiple instance detection while the others have multiple objects and multiple instances detection.

6.3.3 Observations

While conducting experiments, we faced with some issues in the baseline method which also exists in the proposed method. These issues opened new ideas to improve the performance of our proposed methods following in our research. In this section, these issues with some samples of them are represented.

Despite all the claims and news on the strength of deep learning methods, specifically convolutional neural networks, in representing images, we faced with some cases where their representation was not adequate enough to result into a correct classification of the object types. Some samples of this case are represented in Figure 6.9. By looking at these images, a similarity between detected type and the annotated



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 6.10: The effect of NMS on ignoring correctly detected objects. First row is before applying NMS and second row is after this technique has applied.

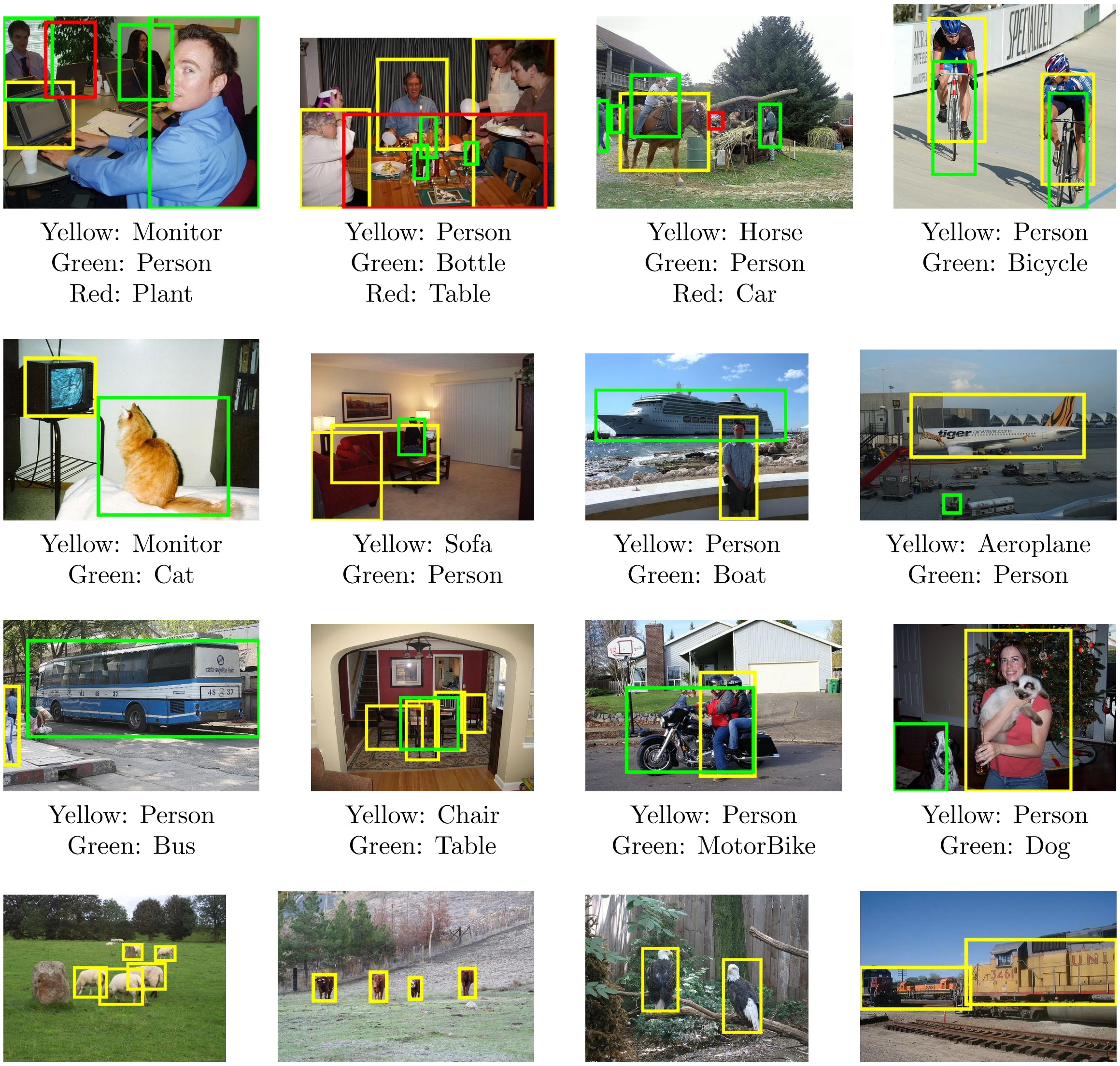
object exists. This shows that it is possible to improve the image representation by adding more contextual information. This motivates us to enhance the image representation of CNN based deep learning methods by using some local and global image descriptors.

Considering images from the first row of Figure 6.4, it can be seen that some of the object bounding boxes are detected correctly before applying the NMS technique in the second row. But after applying the NMS technique, some of them are ignored because they have overlap with other bounding boxes with the same object type. This is really an issue with NMS technique. However, it works well in many cases for ignoring multiple object boundary for a single image, in cases where different instances really exist in the image and are occluded by each other this technique does not work properly and results in losing some of the detected objects. To solve this issue, we propose to have a supervised version of NMS which can handle this condition that occures frequently in the images.

6.4 Summary

In this method, we proposed an object localization technique for improving the performance of object recognition. The proposed method applies a BFS search for the object localization task and its search space, objective function and finishing conditions are specified. The search space is a set of bounding boxes around the candidate object that we want to improve its location. These bounding boxes are obtained by merging the edge segments of the image extracted from its edge map with that candidate object. The search iterates to optimize the detection score of the candidate object as its objective function. This detection score is calculated by using the weights of a linear SVM which is trained on the object representation obtained from CNN. This search for the better localization continues until there is no edge segment for creating new merge box, new search space, or the improvement on the candidate object is stabilized.

One advantage of the proposed method is the fact that it relies on each image's content for improving the localization and does not require any training set or any information about other images in the dataset. This makes the method special for cases where there is not enough training samples and the pretrained CNN model are used for object representation. Another advantage is the reliance of the object localization module on perceptual features of the image which guides the machine learning algorithms to be tuned with the human vision perception of the objects in the image to bridge the gap between human and computer understanding from a scene. The proposed method has been tested on object detection datasets and represents overall improvement compared with the baseline method of RCNN, while individual improvements for some classes are significant. Some sample output of the proposed method are also represented to prove the effectiveness of the proposed method.



Yellow: Monitor
Green: Person
Red: Plant

Yellow: Person
Green: Bottle
Red: Table

Yellow: Horse
Green: Person
Red: Car

Yellow: Person
Green: Bicycle

Yellow: Monitor
Green: Cat

Yellow: Sofa
Green: Person

Yellow: Person
Green: Boat

Yellow: Aeroplane
Green: Person

Yellow: Person
Green: Bus

Yellow: Chair
Green: Table

Yellow: Person
Green: MotorBike

Yellow: Person
Green: Dog

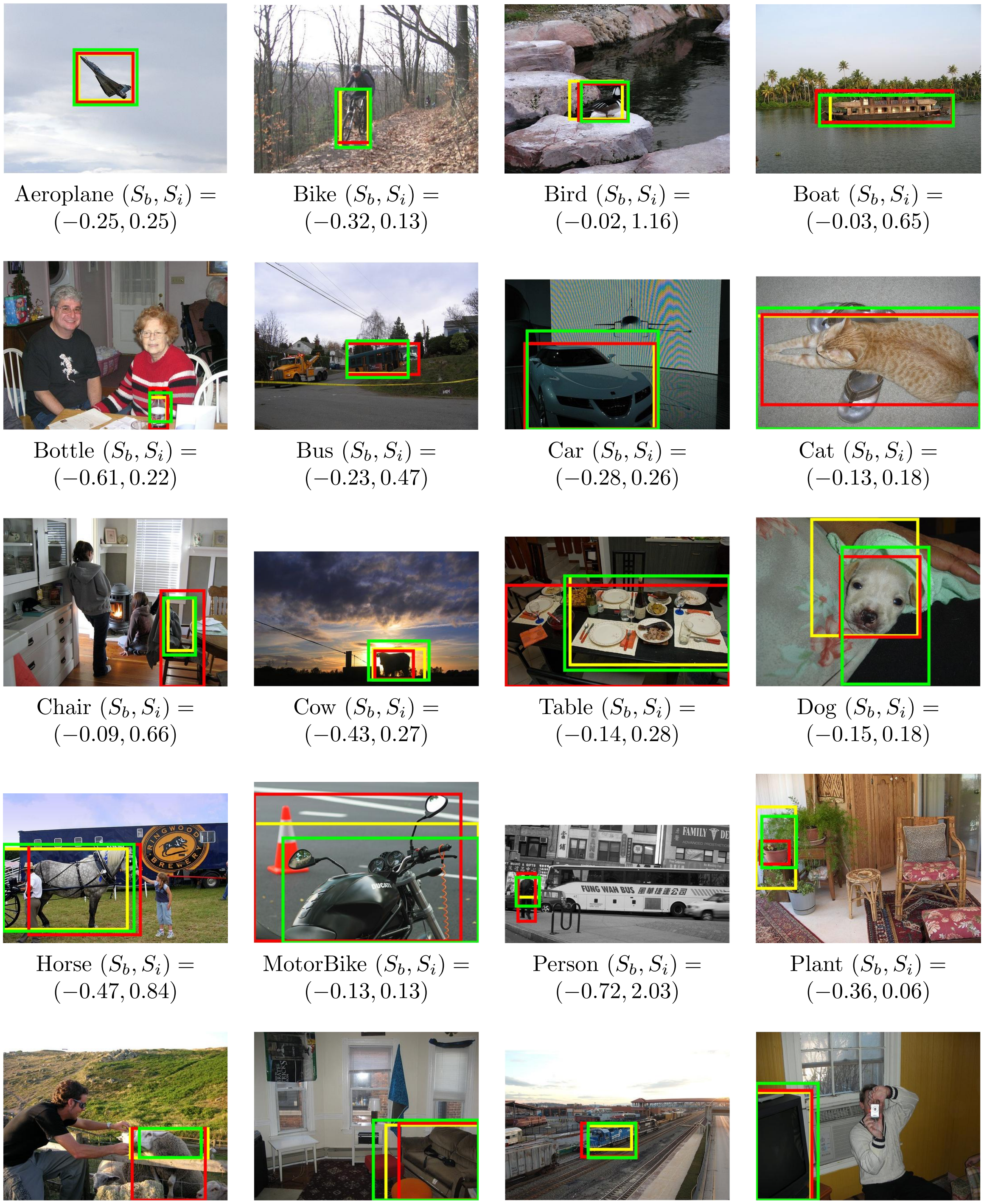
Yellow: Sheep

Yellow: Cow

Yellow: Bird

Yellow: Train

Figure 6.11: Samples of images from Pascal VOC 2007 test set along with their annotations which is generated by the GT_Loc proposed method. Objects with the same type are marked with the same color.



Aeroplane $(S_b, S_i) = (-0.25, 0.25)$

Bike $(S_b, S_i) = (-0.32, 0.13)$

Bird $(S_b, S_i) = (-0.02, 1.16)$

Boat $(S_b, S_i) = (-0.03, 0.65)$

Bottle $(S_b, S_i) = (-0.61, 0.22)$

Bus $(S_b, S_i) = (-0.23, 0.47)$

Car $(S_b, S_i) = (-0.28, 0.26)$

Cat $(S_b, S_i) = (-0.13, 0.18)$

Chair $(S_b, S_i) = (-0.09, 0.66)$

Cow $(S_b, S_i) = (-0.43, 0.27)$

Table $(S_b, S_i) = (-0.14, 0.28)$

Dog $(S_b, S_i) = (-0.15, 0.18)$

Horse $(S_b, S_i) = (-0.47, 0.84)$

MotorBike $(S_b, S_i) = (-0.13, 0.13)$

Person $(S_b, S_i) = (-0.72, 2.03)$

Plant $(S_b, S_i) = (-0.36, 0.06)$

Sheep $(S_b, S_i) = (-0.26, 0.27)$

Sofa $(S_b, S_i) = (-0.16, 0.10)$

Train $(S_b, S_i) = (-0.57, 0.27)$

TV Monitor $(S_b, S_i) = (-0.05, 0.15)$

Figure 6.12: Samples of ground truth object (Red) along with output images from RCNN (Yellow) and GT_Loc (Green). Comparing the detection scores of RCNN (S_b) and GT_Loc (S_i) represents the affect of more precise localization of the proposed method in the object detection.

Chapter 7

Conclusion and Future Work

Object recognition is one of the main tasks in computer vision whose improvement can be achieved by advancing any of its subtasks such as object proposal generation, object detection including image representation and classification, and object localization. In this thesis, we have focused on image representation and object localization tasks and made some improvements. Besides, we have utilized perceptual characteristics of the human vision perception to make these advances. Following in this chapter, brief summary of the proposed methods (Table 7.1) along with some areas of the future work are addressed.

7.1 Conclusion

At first, we have proposed a multilayer image representation method which utilizes the perceptual information of the image's shape to create a hierarchical representation. This method utilizes the PCPG package to extract perceptual features and use the N-gram notation to create the dictionary of visual words. We have created a hierarchy of dictionaries, where the higher level contains visual words with more abstract representation power, and the lower level contains words with more detailed descriptive capabilities. This hierarchy is called the Shape pyramid. To capture the location distribution of each visual word, we applied the Spatial Pyramid technique to our proposed shape pyramid structure and introduced a Spatio-Shape pyramid for describing the image. The experimental results show high accuracy on the benchmark datasets for image classification. Despite the performance gain, this method is limited to a certain visual words difined statically by human supervision which makes it less scalable and limited shape encoding coverage.

To improve this representation method, we have introduced the hybrid image representation method which creates a dictionary of visual words dynamically but without considering the human vision perception. This method creates a scaling octave for each image and extract its edge maps by applying Canny edge detection algorithm. The Hough transform is applied on these edge maps to extract line segment constituent of the edge. The center point of these lines are selected as descriptive areas in the image and described using SIFT and SURF descriptors. Finally Kmeans clustering algorithm is applied on the descibed keypoints to find the bag of visual words which is then utilized for image encoding. This method has shown great performance in comparison to other human-made representation methods on the benchmark datasets. As mentioned earlier, despite its performance, this method does not use the perceptual characteristics of the human vision to improve the fidelity of its representation.

To augment the previous method with perceptual characteristics of the human vision system, we have utilized Gestalt laws of grouping to find the descriptive areas in the image. In this method, we have improved the PCPG edge tracking module by imposing the Gestalt laws of grouping to find Curve Partitioning Points which are used as visual words for describing the image. In this method, we have utilized proximity, similarity, and continuity laws of Gestalt to ignore CPPs with less importance and keep the crucial ones. This proposed method is tested on benchmark datasets and has shown promising performance against existing human-engineered image representation methods.

After improving the object detection task by proposing perceptual image representation methods, we targetted the object localization task for further improvement on the object recognition pipeline. In this method, we extract a set of candidate objects using an object recognition module similar to RCNN. Then we optimize the locations of those candidate objects. The proposed method extracts the edge segments of the image, using the PCPG package. These edge segments are combined with the candidate object, if they have overlap. The obtained merged boxes are evaluated using CNN to extract their representation, and SVM to provide similarity score for each of them. This search on the edge segments are done using the best first search algorithm, while the goal of this algorithm is optimizing the calculated similarity score. The experimental results show the improvement that our proposed method provides for the baseline object recognition method.

Table 7.1: Summary of the proposed image representation (N-gram, hybrid, BoC) and object localization (GT-Loc) methods.

Method	Advantages	Disadvantages	Applications
N-gram	• Perceptual features • Spatial pyramid matching	• Fixed BoWs • Not scalable • Limited shapes	• Sketch retrieval • Character recognition
Hybrid	• Dynamic BoWs • Scale invariance	• Lack perceptual features	• Dark image representation
BoC	• Dynamic BoWs • Perceptual features • Gestalt laws	• Slow representation	• Human-made object recognition
GT-Loc	• Perceptual features • No training data	• Slow localization	• Small data localization

7.2 Future Work

Following in our research, we are going to complete our object recognition pipeline and apply it in an Augmented Reality application dataset. As our future work, we are aiming to do the following researches:

- Using the perceptual features and proposed image representation methods to describe image's content semantically. The current representations are only meaningful to computers and are similar to black boxes to human. Combining perceptual features and deep representation models makes deep representations understandable for human as well.

- Proposing a supervised method to perform non-maximum suppression (NMS) and solve the problem of current object recognition pipelines on the occluded objects.

- Combining the human intelligence in creating the handcrafted local and global features, with the deep learning representation obtained from the computer intelligence, since their combination may solve both of their deficiencies and result in better performances.

- Completing the object recognition pipeline by merging the proposed perceptual image representation methods, and object recognition technique.

Bibliography

[1] The Floyd-Warshall Algorithm, author=Cormen, Thomas H and Leiserson, Charles E and Rivest, Ronald L, journal=Introduction to Algorithms, volume=558, pages=565, year=1990, publisher=MIT Press.

[2] Tinku Acharya and Ping-Sing Tsai. Computational Foundations of Image Interpolation Algorithms. *ACM Ubiquity*, 8(5):1–17, 2007.

[3] V Adithya, PR Vinod, and Usha Gopalakrishnan. Artificial Neural Network based Method for Indian Sign Language Recognition. In *Information and Communication Technologies (ICT)*, pages 1080–1085. IEEE, 2013.

[4] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. Freak: Fast Retina Keypoint. In *Computer Vision and Pattern Recognition (CVPR)*, pages 510–517. IEEE, 2012.

[5] Bogdan Alexe, Thomas Deselaers, and Vittorio Ferrari. What is an Object? In *Computer Vision and Pattern Recognition (CVPR)*, pages 73–80. IEEE, 2010.

[6] Alexander Andreopoulos, Stephan Hasler, Heiko Wersing, Herbert Janssen, John K Tsotsos, and Edgar Korner. Active 3D Object Localization using a Humanoid Robot. *IEEE Transactions on Robotics*, 27(1):47–64, 2011.

[7] Alexander Andreopoulos and John K Tsotsos. 50 Years of Object Recognition: Directions Forward. *Computer Vision and Image Understanding*, 117(8):827–891, 2013.

[8] Alexander Andreopoulos and John K Tsotsos. A Computational Learning Theory of Active Object Recognition under Uncertainty. *International Journal of Computer Vision (IJCV)*, 101(1):95–142, 2013.

[9] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up Robust Features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[10] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up Robust Features. In *European Conference on Computer vision (ECCV)*, pages 404–417. Springer, 2006.

[11] Jerome R Bellegarda. Statistical Language Model Adaptation: Review and Perspectives. *Speech Communication*, 42(1):93–108, 2004.

[12] Samy Bengio, Fernando Pereira, Yoram Singer, and Dennis Strelow. Group Sparse Coding. In *Advances in Neural Information Processing Systems (NIPS)*, pages 82–89, 2009.

[13] Alexander Berengolts and Michael Lindenbaum. On the Distribution of Saliency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(12):1973–1990, 2006.

[14] Stefano Berretti, Alberto Del Bimbo, and Pietro Pala. Retrieval by Shape Similarity with Perceptual Distance and Effective Indexing. *IEEE Transactions on Multimedia*, 2(4):225–239, 2000.

[15] Marco Bertamini and Johan Wagemans. Processing Convexity and Concavity along a 2-D Contour: Figure–ground, Structural Shape, and Attention. *Psychonomic Bulletin and Review*, 20(2):191–207, 2013.

[16] Hakan Bilen, Marco Pedersoli, and Tinne Tuytelaars. Weakly Supervised Object Detection with Posterior Regularization. In *British Machine Vision Conference (BMVC)*, volume 3, pages 1–12, 2014.

[17] Hakan Bilen, Marco Pedersoli, and Tinne Tuytelaars. Weakly Supervised Object Detection with Convex Clustering. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1081–1089, 2015.

[18] Bernhard E Boser, Isabelle M Guyon, and Vladimir N Vapnik. A Training Algorithm for Optimal Margin Classifiers. In *Workshop on Computational Learning Theory*, pages 144–152. ACM, 1992.

[19] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[20] Rodney A Brooks. Symbolic Reasoning among 3-D Models and 2-D Images. *Artificial Intelligence*, 17(1):285–348, 1981.

[21] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: Binary Robust Independent Elementary Features. *European Conference on Computer vision (ECCV)*, pages 778–792, 2010.

[22] John Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):679–698, 1986.

[23] Joao Carreira, Rui Caseiro, Jorge Batista, and Cristian Sminchisescu. Semantic Segmentation with Second-order Pooling. In *European Conference on Computer vision (ECCV)*, pages 430–443. Springer, 2012.

[24] Joao Carreira and Cristian Sminchisescu. Constrained Parametric Min-cuts for Automatic Object Segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3241–3248. IEEE, 2010.

[25] Claudio Castellini, Tatiana Tommasi, Nicoletta Noceti, Francesca Odone, and Barbara Caputo. Using Object Affordances to Improve Object Recognition. *IEEE Transactions on Autonomous Mental Development*, 3(3):207–215, 2011.

[26] Vijay Chandrasekhar, Gabor Takacs, David Chen, Sam Tsai, Radek Grzeszczuk, and Bernd Girod. CHOG: Compressed Histogram of Gradients a Low Bit-rate Feature Descriptor. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2504–2511. IEEE, 2009.

[27] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a Library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[28] Dempsey Chang, Keith V Nesbitt, and Kevin Wilkins. The Gestalt Principles of Similarity and Proximity Apply to Both the Haptic and Visual Grouping of Elements. In *Australasian Conference on User Interface*, volume 64, pages 79–86. Australian Computer Society, Inc., 2007.

[29] Tianshui Chen, Liang Lin, Xian Wu, and Xiaonan Luo. Learning to Segment Object Proposals via Recursive Neural Networks. *ArXiv Preprint arXiv:1612.01057*, 2016.

[30] Xiaozhi Chen, Huimin Ma, Xiang Wang, and Zhichen Zhao. Improving Object Proposals with Multi-thresholding Straddling Expansion. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2587–2595, 2015.

[31] Gong Cheng and Junwei Han. A Survey on Object Detection in Optical Remote Sensing Images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 117:11–28, 2016.

[32] Gong Cheng, Junwei Han, Lei Guo, and Tianming Liu. Learning Coarse-to-fine Sparselets for Efficient Object Detection and Scene Classification. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1173–1181, 2015.

[33] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Weakly Supervised Object Localization with Multi-fold Multiple Instance Learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(1):189–203, 2017.

[34] Adam Coates and Andrew Y Ng. The Importance of Encoding Versus Training with Sparse Coding and Vector Quantization. In *International Conference on Machine Learning (ICML)*, pages 921–928, 2011.

[35] Jerome Paul N Cruz, Ma Lourdes Dimaala, Laurene Gaile L Francisco, Erica Joanna S Franco, Argel A Bandala, and Elmer P Dadios. Object Recognition and Detection by Shape and Color Pattern Recognition Utilizing Artificial Neural Networks. In *Information and Communication Technologies (ICT)*, pages 140–144. IEEE, 2013.

[36] Navneet Dalal and Bill Triggs. Histograms of Oriented Gradients for Human Detection. In *Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 886–893. IEEE, 2005.

[37] Nasser H Dardas and Nicolas D Georganas. Real-time Hand Gesture Detection and Recognition using Bag-of-features and Support Vector Machine Techniques. *IEEE Transactions on Instrumentation and Measurement*, 60(11):3592–3607, 2011.

[38] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A Large-scale Hierarchical Image Database. In *Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. IEEE, 2009.

[39] Shiv Ram Dubey and Anand Singh Jalal. Apple Disease Classification using Color, Texture and Shape Features from Images. *Signal, Image and Video Processing*, pages 1–8, 2015.

[40] Richard O Duda and Peter E Hart. Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Communications of the ACM*, 15(1):11–15, 1972.

[41] Elham Etemad and Qigang Gao. Optimizing Convolutional Neural Network Detection Score using Generic Edge Features for Object Localization. ***Submitted to:*** *Computer Vision and Image Understanding.*

[42] Elham Etemad and Qigang Gao. Perceptual Image Representation Using Bag of Curve Partitioning Points. ***Accepted in:*** *Digital Image Computing: Techniques and Applications (**DICTA**), year=September 21, 2018,.*

[43] Elham Etemad and Qigang Gao. Object Localization by Optimizing Convolutional Neural Network Detection Score using Generic Edge Features. In *IEEE International Conference on Image Processing (ICIP)*, pages 675–679. IEEE, 2017.

[44] Elham Etemad and Qigang Gao. Hybrid Image Representation Method based on Bag of Edge Tokens from Octaves of Edge Elements. In *International Conference on Pattern Recognition and Artificial Intelligence (ICPRAI)*. IEEE, 2018.

[45] Elham Etemad, Gang Hu, and Qigang Gao. Perceptual Shape Words based Representation for Image Classification. ***Submitted to:*** *International Journal of Pattern Recognition and Artificial Intelligence, year=2018, publisher=Springer.*

[46] Elham Etemad, Gang Hu, and Qigang Gao. A Shape Feature based BOVW Method for Image Classification using N-gram and Spatial Pyramid Coding Scheme. In *IEEE International Conference on Image Processing (ICIP)*, pages 504–508. IEEE, 2016.

[47] M Everingham, L Van Gool, CKI Williams, J Winn, and A Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC 2007) Results (2007), Accessed: January 2016.

[48] M Everingham, L Van Gool, CKI Williams, J Winn, and A Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012), Accessed: January 2016.

[49] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The PASCAL Visual Object Classes Challenge: A Retrospective. *International Journal of Computer Vision (IJCV)*, 111(1):98–136, 2015.

[50] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. LIBLINEAR: A Library for Large Linear Classification. *Journal of Machine Learning Research*, 9(Aug):1871–1874, 2008.

[51] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories. *Computer Vision and Image Understanding*, 106(1):59–70, 2007.

[52] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object Detection with Discriminatively Trained Part-based Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010.

[53] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient Graph-based Image Segmentation. *International Journal of Computer Vision (IJCV)*, 59(2):167–181, 2004.

[54] Leandro AF Fernandes and Manuel M Oliveira. Real-time Line Detection Through an Improved Hough Transform Voting Scheme. *Pattern Recognition*, 41(1):299–314, 2008.

[55] Itzhak Fogel and Dov Sagi. Gabor Filters as Texture Discriminator. *Biological Cybernetics*, 61(2):103–113, 1989.

[56] Qi-Gang Gao and AKC Wong. Curve Detection based on Perceptual Organization. *Pattern Recognition*, 26(7):1039–1046, 1993.

[57] Theo Gevers. Color in image database, 1998.

[58] Ross Girshick. Fast R-CNN. In *International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[59] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014.

[60] Hanlin Goh, Nicolas Thome, Matthieu Cord, and Joo-Hwee Lim. Learning Deep Hierarchical Visual Feature Coding. *IEEE Transactions on Neural Networks and Learning Systems*, 25(12):2212–2225, 2014.

[61] Robert M Gray. Vector Quantization. *Acoustic Speach and Signal Processing Magazine (ASSP)*, 1(2):4–29, 1984.

[62] Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 Object Category Dataset. 2007.

[63] Chunhui Gu, Joseph J Lim, Pablo Arbeláez, and Jitendra Malik. Recognition using Regions. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1030–1037. IEEE, 2009.

[64] Lie Guo, Ping-Shu Ge, Ming-Heng Zhang, Lin-Hui Li, and Yi-Bing Zhao. Pedestrian Detection for Intelligent Transportation Systems Combining AdaBoost Algorithm and Support Vector Machine. *Expert Systems with Applications*, 39(4):4274–4286, 2012.

[65] Zhenhua Guo, Lei Zhang, and David Zhang. A Completed Modeling of Local Binary Pattern Operator for Texture Classification. *IEEE Transactions on Image Processing*, 19(6):1657–1663, 2010.

[66] Neetika Gupta and Mukesh Kumar Rohil. Image Feature Detection using an Improved Implementation of Maximally Stable Extremal Regions for Augmented Reality Applications. *International Journal of Image and Data Fusion*, pages 1–20, 2017.

[67] Chris Harris and Mike Stephens. A Combined Corner and Edge Detector. In *Alvey Vision Conference*, volume 15, page 50. Citeseer, 1988.

[68] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In *European Conference on Computer vision (ECCV)*, pages 346–361. Springer, 2014.

[69] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving Deep into Rectifiers: Surpassing Human-level Performance on Imagenet Classification. In *International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.

[70] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[71] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation*, 18(7):1527–1554, 2006.

[72] Gang Hu and Qigang Gao. A Non-parametric Statistics based Method for Generic Curve Partition and Classification. In *IEEE International Conference on Image Processing (ICIP)*, pages 3041–3044. IEEE, 2010.

[73] Ming-Kuei Hu. Visual Pattern Recognition by Moment Invariants. *IRE Transactions on Information Theory*, 8(2):179–187, 1962.

[74] Yongzhen Huang, Zifeng Wu, Liang Wang, and Tieniu Tan. Feature Coding in Image Classification: A Comprehensive Study. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(3):493–506, 2014.

[75] David H Hubel. *Eye, Brain, and Vision.* Scientific American Library/Scientific American Books, 1995.

[76] Daniel P Huttenlocher and Shimon Ullman. Recognizing Solid Objects by Alignment with an Image. *International Journal of Computer Vision (IJCV)*, 5(2):195–212, 1990.

[77] Kashif Iqbal, Michael O Odetayo, and Anne James. Content-based Image Retrieval Approach for Biometric Security using Colour, Texture and Shape Features Controlled by Fuzzy Heuristics. *Journal of Computer and System Sciences*, 78(4):1258–1277, 2012.

[78] Qasim Iqbal and Jake K Aggarwal. Retrieval by Classification of Images Containing Large Manmade Objects using Perceptual Grouping. *Pattern Recognition*, 35(7):1463–1479, 2002.

[79] Mathews Jacob and Michael Unser. Design of Steerable Filters for Feature Detection using Canny-like Criteria. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(8):1007–1019, 2004.

[80] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional Architecture for Fast Feature Embedding. In *ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.

[81] Lu Jin, Shenghua Gao, Zechao Li, and Jinhui Tang. Hand-crafted Features or Machine Learnt Features? Together they Improve RGB-D Object Recognition. In *International Symposium on Multimedia (ISM)*, pages 311–319. IEEE, 2014.

[82] George H Joblove and Donald Greenberg. Color Spaces for Computer Graphics. In *ACM SIGGRAPH Computer Graphics*, volume 12, pages 20–25. ACM, 1978.

[83] Ian Jolliffe. *Principal Component Analysis.* Wiley Online Library, 2002.

[84] Sabine Kastner and Leslie G Ungerleider. The Neural Basis of Biased Competition in Human Visual Cortex. *Neuropsychologia*, 39(12):1263–1276, 2001.

[85] Yan Ke and Rahul Sukthankar. PCA-SIFT: A more Distinctive Representation for Local Image Descriptors. In *Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II–506. IEEE, 2004.

[86] Alireza Khotanzad and Yaw Hua Hong. Invariant Image Recognition by Zernike Moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):489–497, 1990.

[87] Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *ArXiv Preprint arXiv:1412.6980*, 2014.

[88] Ron Kohavi et al. A Study of Cross Validation and Bootstrap for Accuracy Estimation and Model Selection. In *International Joint Conference on Artificial Intelligence (IJCAI)*, volume 14, pages 1137–1145, 1995.

[89] Tao Kong, Anbang Yao, Yurong Chen, and Fuchun Sun. HyperNet: Towards Accurate Region Proposal Generation and Joint Object Detection. In *Computer Vision and Pattern Recognition (CVPR)*, pages 845–853, 2016.

[90] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.

[91] P Kshirsagar and N Rathod. Artificial Neural Network. *International Journal of Computer Applications*, 2012.

[92] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 2169–2178. IEEE, 2006.

[93] Svetlana Lazebnik, Cordelia Schmid, Jean Ponce, et al. Spatial Pyramid Matching. *Object Categorization: Computer and Human Vision Perspectives*, 3(4), 2009.

[94] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[95] Honglak Lee, Alexis Battle, Rajat Raina, and Andrew Y Ng. Efficient Sparse Coding Algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, pages 801–808, 2006.

[96] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. BRISK: Binary Robust Invariant Scalable Keypoints. In *International Conference on Computer Vision (ICCV)*, pages 2548–2555. IEEE, 2011.

[97] Guang-Hai Liu and Jing-Yu Yang. Content-based Image Retrieval using Color Difference Histogram. *Pattern Recognition*, 46(1):188–198, 2013.

[98] Ruijun Liu, Yi Chen, Xiaobin Zhu, and Kun Hou. Image Classification using Label Constrained Sparse Coding. *Multimedia Tools and Applications*, pages 1–15, 2015.

[99] Margaret S Livingstone and David H Hubel. Anatomy and Physiology of a Color System in the Primate Visual Cortex. *The Journal of Neuroscience*, 4(1):309–356, 1984.

[100] David G Lowe. The Viewpoint Consistency Constraint. *International Journal of Computer Vision (IJCV)*, 1(1):57–72, 1987.

[101] David G Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision (IJCV)*, 60(2):91–110, 2004.

[102] Matei Mancas, Bernard Gosselin, and Benoît Macq. Perceptual Image Representation. *EURASIP Journal on Image and Video Processing*, 2007(1):098181, 2007.

[103] David Marr. Vision: A Computational Investigation into the Human Representation and Processing of Visual Information. 1982.

[104] Jose L Marroquin and Federico Girosi. Some Extensions of the K-means Algorithm for Image Segmentation and Pattern Classification. Technical report, Massachusetts Institute of Technology, Cambridge Artificial Intelligence Lab, 1993.

[105] Majid Mirmehdi and Radhakrishnan Perissamy. Perceptual Image Indexing and Retrieval. *Journal of Visual Communication and Image Representation*, 13(4):460–475, 2002.

[106] Robert Moore and John DeNero. L1 and L2 Regularization for Multiclass Hinge Loss Models. In *Machine Learning in Spoken Language Processing (MLSLP)*, pages 1–5, 2011.

[107] Albina Mukanova, Qigang Gao, and Gang Hu. N-gram based Image Representation and Classification using Perceptual Shape Features. In *Canadian Conference on Computer and Robot Vision*, pages 349–356. IEEE, 2014.

[108] Scott O Murray, Daniel Kersten, Bruno A Olshausen, Paul Schrater, and David L Woods. Shape Perception Reduces Activity in Human Primary Visual Cortex. *Proceedings of the National Academy of Sciences*, 99(23):15164–15169, 2002.

[109] Institute of Electrical and Electronics Engineers. *IEEE Standards Glossary of Image Processing and Pattern Recognition Terminology*. IEEE, 1990.

[110] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Multiresolution Gray-scale and Rotation Invariant Texture Classification with Local Binary Patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[111] Bruno A Olshausen and JD Field. What is the Other 85 Percent of V1 Doing. *L. van Hemmen, and T. Sejnowski (Eds.)*, 23:182–211, 2006.

[112] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and Transferring Mid-level Image Representations using Convolutional Neural Networks. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1717–1724, 2014.

[113] Wanli Ouyang, Xiaogang Wang, Xingyu Zeng, Shi Qiu, Ping Luo, Yonglong Tian, Hongsheng Li, Shuo Yang, Zhe Wang, Chen-Change Loy, et al. DeepID-Net: Deformable Deep Convolutional Neural Networks for Object Detection. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2403–2412, 2015.

[114] Greg Pass, Ramin Zabih, and Justin Miller. Comparing Images using Color Coherence Vectors. In *ACM International Conference on Multimedia*, pages 65–73. ACM, 1997.

[115] Prashan Premaratne and Q Nguyen. Consumer Electronics Control System based on Hand Gesture Moment Invariants. *Computer Vision, IET*, 1(1):35–41, 2007.

[116] Waseem Rawat and Zenghui Wang. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Computation*, 29(9):2352–2449, 2017.

[117] Ali Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN Features Off-the-shelf: an Astounding Baseline for Recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 806–813, 2014.

[118] Roberto Rigamonti, Matthew A Brown, and Vincent Lepetit. Are Sparse Representations Really Relevant for Image Classification? In *Computer Vision and Pattern Recognition (CVPR)*, pages 1545–1552. IEEE, 2011.

[119] Edward Rosten and Tom Drummond. Machine Learning for High-speed Corner Detection. In *European Conference on Computer vision (ECCV)*, pages 430–443. Springer, 2006.

[120] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: an Efficient Alternative to SIFT or SURF. In *International Conference on Computer Vision (ICCV)*, pages 2564–2571. IEEE, 2011.

[121] Dennis W Ruck, Steven K Rogers, Matthew Kabrisky, Mark E Oxley, and Bruce W Suter. The Multilayer Perceptron as an Approximation to a Bayes Optimal Discriminant Function. *IEEE Transactions on Neural Networks*, 1(4):296–298, 1990.

[122] Stuart Jonathan Russell, Peter Norvig, John F Canny, Jitendra M Malik, and Douglas D Edwards. *Artificial Intelligence: a Modern Approach*, volume 2. Prentice Hall Upper Saddle River, 2003.

[123] Cordelia Schmid. Constructing Models for Content-based Image Retrieval. In *Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II–II. IEEE, 2001.

[124] Mark W Schmidt, Kevin P Murphy, Glenn Fung, and Rómer Rosales. Structure Learning in Random Fields for Heart Motion Abnormality Detection. In *Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 2, 2008.

[125] David Sculley. Web-scale k-means Clustering. In *International Conference on World Wide Web*, pages 1177–1178. ACM, 2010.

[126] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[127] Miaojing Shi and Vittorio Ferrari. Weakly Supervised Object Localization Using Size Estimates. In *European Conference on Computer vision (ECCV)*, pages 105–121. Springer, 2016.

[128] Patrick Y Shinzato, Valdir Grassi, Fernando S Osorio, and Denis F Wolf. Fast Visual Road Recognition and Horizon Detection using Multiple Artificial Neural Networks. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 1090–1095. IEEE, 2012.

[129] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-scale Image Recognition. *ArXiv Preprint ArXiv:1409.1556*, 2014.

[130] Manimala Singha and K Hemachandran. Content based Image Retrieval using Color and Texture. *International Journal on Signal and Image Processing (SIPIJ)*, 3(1):39–57, 2012.

[131] Paul Smolensky. Information Processing in Dynamical Systems: Foundations of Harmony Theory. Technical report, DTIC Document, 1986.

[132] Louise Stark and Kevin Bowyer. Function-based Generic Recognition for Multiple Object Categories. *CVGIP: Image Understanding*, 59(1):1–21, 1994.

[133] Michael Stark, Philipp Lies, Michael Zillich, Jeremy Wyatt, and Bernt Schiele. Functional Object Class Detection based on Learned Affordance Cues. In *International Conference on Computer Vision Systems*, pages 435–444. Springer, 2008.

[134] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In *Association for the Advancement of Artificial Intelligence (AAAI)*, pages 4278–4284, 2017.

[135] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going Deeper with Convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[136] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the Inception Architecture for Computer Vision. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.

[137] Jasper RR Uijlings, Koen EA van de Sande, Theo Gevers, and Arnold WM Smeulders. Selective Search for Object Recognition. *International Journal of Computer Vision (IJCV)*, 104(2):154–171, 2013.

[138] Vladimir Naumovich Vapnik. An Overview of Statistical Learning Theory. *IEEE Transactions on Neural Networks*, 10(5):988–999, 1999.

[139] Paul Viola and Michael J Jones. Robust Real-time Face Detection. *International Journal of Computer Vision (IJCV)*, 57(2):137–154, 2004.

[140] Johan Wagemans, James H Elder, Michael Kubovy, Stephen E Palmer, Mary A Peterson, Manish Singh, and Rüdiger von der Heydt. A Century of Gestalt Psychology in Visual Perception: I. Perceptual Grouping and Figure–ground Organization. *Psychological Bulletin*, 138(6):1172, 2012.

[141] Chaoyang Wang, Long Zhao, Shuang Liang, Liqing Zhang, Jinyuan Jia, and Yichen Wei. Object Proposal by Multi-branch Hierarchical Segmentation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3873–3881, 2015.

[142] Fei Wang, Noah Lee, Jimeng Sun, Jianying Hu, and Shahram Ebadollahi. Automatic Group Sparse Coding. In *Association for the Advancement of Artificial Intelligence (AAAI)*, pages 495–500, 2011.

[143] James Z Wang, Jia Li, and Gio Wiederhold. SIMPLIcity: Semantics-sensitive Integrated Matching for Picture Libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9):947–963, 2001.

[144] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. Locality-constrained Linear Coding for Image Classification. In *Computer Vision and Pattern Recognition (CVPR)*, pages 3360–3367. IEEE, 2010.

[145] Xiang-Yang Wang, Yong-Jian Yu, and Hong-Ying Yang. An Effective Image Retrieval Scheme using Color, Texture and Shape Features. *Computer Standards and Interfaces*, 33(1):59–68, 2011.

[146] M Wertheimer. Experimental Studies on the Seeing of Motion. *Psychologia*, 61:161–265, 1912.

[147] Andrew P Witkin. Scale-space Filtering, April 14 1987. US Patent 4,658,372.

[148] Shangxuan Wu, Ying-Cong Chen, Xiang Li, An-Cong Wu, Jin-Jie You, and Wei-Shi Zheng. An Enhanced Deep Feature Representation for Person Re-identification. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–8. IEEE, 2016.

[149] Tong Tong Wu and Kenneth Lange. Coordinate Descent Algorithms for Lasso Penalized Regression. *The Annals of Applied Statistics*, pages 224–244, 2008.

[150] Yao Xiao, Cewu Lu, Efstratios Tsougenis, Yongyi Lu, and Chi-Keung Tang. Complexity-adaptive Distance Metric for Object Proposals Generation. In *Computer Vision and Pattern Recognition (CVPR)*, pages 778–786, 2015.

[151] Junjie Yan, Yinan Yu, Xiangyu Zhu, Zhen Lei, and Stan Z Li. Object Detection by Labeling Superpixels. In *Computer Vision and Pattern Recognition (CVPR)*, pages 5107–5116, 2015.

[152] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear Spatial Pyramid Matching using Sparse Coding for Image Classification. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1794–1801. IEEE, 2009.

[153] Ming Yuan and Yi Lin. Model Selection and Estimation in Regression with Grouped Variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[154] Jun Yue, Zhenbo Li, Lu Liu, and Zetian Fu. Content-based Image Retrieval using Color and Texture Fused Features. *Mathematical and Computer Modelling*, 54(3):1121–1127, 2011.

[155] Matthew D Zeiler and Rob Fergus. Visualizing and Understanding Convolutional Networks. In *European Conference on Computer vision (ECCV)*, pages 818–833. Springer, 2014.

[156] Xiaofen Zheng, Scott A Sherrill-Mix, and Qigang Gao. Perceptual Shape-based Natural Image Representation and Retrieval. In *International Conference on Semantic Computing (ICSC)*, pages 622–629. IEEE, 2007.

[157] Long Zhu, Yuanhao Chen, Alan Yuille, and William Freeman. Latent Hierarchical Structural Learning for Object Detection. In *Computer Vision and Pattern Recognition (CVPR)*, pages 1062–1069. IEEE, 2010.

[158] Yukun Zhu, Raquel Urtasun, Ruslan Salakhutdinov, and Sanja Fidler. SegDeepM: Exploiting Segmentation and Context in Deep Neural Networks for Object Detection. In *Computer Vision and Pattern Recognition (CVPR)*, pages 4703–4711, 2015.

[159] C Lawrence Zitnick and Piotr Dollár. Edge Boxes: Locating Object Proposals from Edges. In *European Conference on Computer vision (ECCV)*, pages 391–405. Springer, 2014.

IEEE COPYRIGHT AND CONSENT FORM

To ensure uniformity of treatment among all contributors, other forms may not be substituted for this form, nor may any wording of the form be changed. This form is intended for original material submitted to the IEEE and must accompany any such material in order to be published by the IEEE. Please read the form carefully and keep a copy for your files.

TITLE OF PAPER/ARTICLE/REPORT, INCLUDING ALL CONTENT IN ANY FORM, FORMAT, OR MEDIA (hereinafter, "the Work"):

A shape feature based bovw method for image classification using N-gram and spatial pyramid coding scheme

COMPLETE LIST OF AUTHORS:

Elham Etemad, Dalhousie University, Canada; Gang Hu, Dalhousie University, Canada; Qigang Gao, Dalhousie University, Canada

IEEE PUBLICATION TITLE (Journal, Magazine, Conference, Book):

2016 IEEE International Conference on Image Processing (ICIP)

COPYRIGHT TRANSFER

1. The undersigned hereby assigns to The Institute of Electrical and Electronics Engineers, Incorporated (the "IEEE") all rights under copyright that may exist in and to: (a) the above Work, including any revised or expanded derivative works submitted to the IEEE by the undersigned based on the Work; and (b) any associated written or multimedia components or other enhancements accompanying the Work.

CONSENT AND RELEASE

2. In the event the undersigned makes a presentation based upon the Work at a conference hosted or sponsored in whole or in part by the IEEE, the undersigned, in consideration for his/her participation in the conference, hereby grants the IEEE the unlimited, worldwide, irrevocable permission to use, distribute, publish, license, exhibit, record, digitize, broadcast, reproduce and archive, in any format or medium, whether now known or hereafter developed: (a) his/her presentation and comments at the conference; (b) any written materials or multimedia files used in connection with his/her presentation; and (c) any recorded interviews of him/her (collectively, the "Presentation"). The permission granted includes the transcription and reproduction of the Presentation for inclusion in products sold or distributed by IEEE and live or recorded broadcast of the Presentation during or after the conference.

3. In connection with the permission granted in Section 2, the undersigned hereby grants IEEE the unlimited, worldwide, irrevocable right to use his/her name, picture, likeness, voice and biographical information as part of the advertisement, distribution and sale of products incorporating the Work or Presentation, and releases IEEE from any claim based on right of privacy or publicity.

4. The undersigned hereby warrants that the Work and Presentation (collectively, the "Materials") are original and that he/she is the author of the Materials. To the extent the Materials incorporate text passages, figures, data or other material from the works of others, the undersigned has obtained any necessary permissions. Where necessary, the undersigned has obtained all third party permissions and consents to grant the license above and has provided copies of such permissions and consents to IEEE.

□ Please check this box if you do not wish to have video/audio recordings made of your conference presentation.

See reverse side for Retained Rights/Terms and Conditions, and Author Responsibilities.

GENERAL TERMS

- The undersigned represents that he/she has the power and authority to make and execute this assignment.
- The undersigned agrees to indemnify and hold harmless the IEEE from any damage or expense that may arise in the event of a breach of any of the warranties set forth above.
- In the event the above work is not accepted and published by the IEEE or is withdrawn by the author(s) before acceptance by the IEEE, the foregoing copyright transfer shall become null and void and all materials embodying the Work submitted to the IEEE will be destroyed.
- For jointly authored Works, all joint authors should sign, or one of the authors should sign as authorized agent for the others.

(1) Elham Etemad ______________________ February 2, 2016 ______________

Author/Authorized Agent for Joint Authors **Date**

U.S. GOVERNMENT EMPLOYEE CERTIFICATION (WHERE APPLICABLE)

This will certify that all authors of the Work are U.S. government employees and prepared the Work on a subject within the scope of their official duties. As such, the Work is not subject to U.S. copyright protection.

(2)______________________________ ______________________

Authorized Signature **Date**

(Authors who are U.S. government employees should also sign signature line (1) above to enable the IEEE to claim and protect its copyright in international jurisdictions.)

CROWN COPYRIGHT CERTIFICATION (WHERE APPLICABLE)

This will certify that all authors of the Work are employees of the British or British Commonwealth Government and prepared the Work in connection with their official duties. As such, the Work is subject to Crown Copyright and is not assigned to the IEEE as set forth in the first sentence of the Copyright Transfer Section above. The undersigned acknowledges, however, that the IEEE has the right to publish, distribute and reprint the Work in all forms and media.

(3)______________________________ ______________________

Authorized Signature **Date**

(Authors who are British or British Commonwealth Government employees should also sign line (1) above to indicate their acceptance of all terms other than the copyright transfer.)

IEEE COPYRIGHT FORM *(continued)*

RETAINED RIGHTS/TERMS AND CONDITIONS

General

1. Authors/employers retain all proprietary rights in any process, procedure, or article of manufacture described in the Work.

2. Authors/employers may reproduce or authorize others to reproduce the Work, material extracted verbatim from the Work, or derivative works for the author's personal use or for company use, provided that the source and the IEEE copyright notice are indicated, the copies are not used in any way that implies IEEE endorsement of a product or service of any employer, and the copies themselves are not offered for sale.

3. In the case of a Work performed under a U.S. Government contract or grant, the IEEE recognizes that the U.S. Government has royalty-free permission to reproduce all or portions of the Work, and to authorize others to do so, for official U.S. Government purposes only, if the contract/grant so requires.

4. Although authors are permitted to re-use all or portions of the Work in other works, this does not include granting third-party requests for reprinting, republishing, or other types of re-use. The IEEE Intellectual Property Rights office must handle all such third-party requests.

5. Authors whose work was performed under a grant from a government funding agency are free to fulfill any deposit mandates from that funding agency.

Author Online Use

6. Personal Servers. Authors and/or their employers shall have the right to post the accepted version of IEEE-copyrighted articles on their own personal servers or the servers of their institutions or employers without permission from IEEE, provided that the posted version includes a prominently displayed IEEE copyright notice and, when published, a full citation to the original IEEE publication, including a link to the article abstract in IEEE Xplore. Authors shall not post the final, published versions of their papers.

7. Classroom or Internal Training Use. An author is expressly permitted to post any portion of the accepted version of his/her own IEEE-copyrighted articles on the author's personal web site or the servers of the author's institution or company in connection with the author's teaching, training, or work responsibilities, provided that the appropriate copyright, credit, and reuse notices appear prominently with the posted material. Examples of permitted uses are lecture materials, course packs, e-reserves, conference presentations, or in-house training courses.

8. Electronic Preprints. Before submitting an article to an IEEE publication, authors frequently post their manuscripts to their own web site, their employer's site, or to another server that invites constructive comment from colleagues. Upon submission of an article to IEEE, an author is required to transfer copyright in the article to IEEE, and the author must update any previously posted version of the article with a prominently displayed IEEE copyright notice. Upon publication of an article by the IEEE, the author must replace any previously posted electronic versions of the article with either (1) the full citation to the IEEE work with a Digital Object Identifier (DOI) or link to the article abstract in IEEE Xplore, or (2) the accepted version only (not the IEEE-published version), including the IEEE copyright notice and full citation, with a link to the final, published article in IEEE Xplore.

INFORMATION FOR AUTHORS

Author Responsibilities

The IEEE distributes its technical publications throughout the world and wants to ensure that the material submitted to its publications is properly available to the readership of those publications. Authors must ensure that their Work meets the requirements as stated in section 8.2.1 of the IEEE PSPB Operations Manual, including provisions covering originality, authorship, author responsibilities and author misconduct. More information on IEEE's publishing policies may be found at http://www.ieee.org/publications_standards/publications/rights/pub_tools_policies.html. Authors are advised especially of IEEE PSPB Operations Manual section 8.2.1.B12: "It is the responsibility of the authors, not the IEEE, to determine whether disclosure of their material requires the prior consent of other parties and, if so, to obtain it." Authors are also advised of IEEE PSPB Operations Manual section 8.1.1B: "Statements and opinions given in work published by the IEEE are the expression of the authors."

Author/Employer Rights

If you are employed and prepared the Work on a subject within the scope of your employment, the copyright in the Work belongs to your employer as a work-for-hire. In that case, the IEEE assumes that when you sign this Form, you are authorized to do so by your employer and that your employer has consented to the transfer of copyright, to the representation and warranty of publication rights, and to all other terms and conditions of this Form. If such authorization and consent has not been given to you, an authorized representative of your employer should sign this Form as the Author.

IEEE Copyright Ownership

It is the formal policy of the IEEE to own the copyrights to all copyrightable material in its technical publications and to the individual contributions contained therein, in order to protect the interests of the IEEE, its authors and their employers, and, at the same time, to facilitate the appropriate re-use of this material by others. The IEEE distributes its technical publications throughout the world and does so by various means such as hard copy, microfiche, microfilm, and electronic media. It also abstracts and may translate its publications, and articles contained therein, for inclusion in various compendiums, collective works, databases and similar publications.

THIS FORM MUST ACCOMPANY THE SUBMISSION OF THE AUTHOR'S MANUSCRIPT.
Questions about the submission of the form or manuscript must be sent to the publication's editor.
Please direct all questions about IEEE copyright policy to:
IEEE Intellectual Property Rights Office, copyrights@ieee.org, +1-732-562-3966 (telephone)

IEEE COPYRIGHT AND CONSENT FORM

To ensure uniformity of treatment among all contributors, other forms may not be substituted for this form, nor may any wording of the form be changed. This form is intended for original material submitted to the IEEE and must accompany any such material in order to be published by the IEEE. Please read the form carefully and keep a copy for your files.

OBJECT LOCALIZATION BY OPTIMIZING CONVOLUTIONAL NEURAL NETWORK DETECTION SCORE USING GENERIC EDGE FEATURES
Elham Etemad, Qigang Gao
2017 IEEE International Conference on Image Processing (ICIP)

COPYRIGHT TRANSFER

The undersigned hereby assigns to The Institute of Electrical and Electronics Engineers, Incorporated (the "IEEE") all rights under copyright that may exist in and to: (a) the Work, including any revised or expanded derivative works submitted to the IEEE by the undersigned based on the Work; and (b) any associated written or multimedia components or other enhancements accompanying the Work.

GENERAL TERMS

1. The undersigned represents that he/she has the power and authority to make and execute this form.
2. The undersigned agrees to indemnify and hold harmless the IEEE from any damage or expense that may arise in the event of a breach of any of the warranties set forth above.
3. The undersigned agrees that publication with IEEE is subject to the policies and procedures of the IEEE PSPB Operations Manual.
4. In the event the above work is not accepted and published by the IEEE or is withdrawn by the author(s) before acceptance by the IEEE, the foregoing copyright transfer shall be null and void. In this case, IEEE will retain a copy of the manuscript for internal administrative/record-keeping purposes.
5. For jointly authored Works, all joint authors should sign, or one of the authors should sign as authorized agent for the others.
6. The author hereby warrants that the Work and Presentation (collectively, the "Materials") are original and that he/she is the author of the Materials. To the extent the Materials incorporate text passages, figures, data or other material from the works of others, the author has obtained any necessary permissions. Where necessary, the author has obtained all third party permissions and consents to grant the license above and has provided copies of such permissions and consents to IEEE

You have indicated that you DO wish to have video/audio recordings made of your conference presentation under terms and conditions set forth in "Consent and Release."

CONSENT AND RELEASE

1. In the event the author makes a presentation based upon the Work at a conference hosted or sponsored in whole or in part by the IEEE, the author, in consideration for his/her participation in the conference, hereby grants the IEEE the unlimited, worldwide, irrevocable permission to use, distribute, publish, license, exhibit, record, digitize, broadcast, reproduce and archive, in any format or medium, whether now known or hereafter developed: (a) his/her presentation and comments at the conference; (b) any written materials or multimedia files used in connection with his/her presentation; and (c) any recorded interviews of him/her (collectively, the "Presentation"). The permission granted includes the transcription and reproduction of the Presentation for inclusion in products sold or distributed by IEEE and live or recorded broadcast of the Presentation during or after the conference.
2. In connection with the permission granted in Section 1, the author hereby grants IEEE the unlimited, worldwide, irrevocable right to use his/her name, picture, likeness, voice and biographical information as part of the advertisement, distribution and sale of products incorporating the Work or Presentation, and releases IEEE from any claim based on

right of privacy or publicity.

BY TYPING IN YOUR FULL NAME BELOW AND CLICKING THE SUBMIT BUTTON, YOU CERTIFY THAT SUCH ACTION CONSTITUTES YOUR ELECTRONIC SIGNATURE TO THIS FORM IN ACCORDANCE WITH UNITED STATES LAW, WHICH AUTHORIZES ELECTRONIC SIGNATURE BY AUTHENTICATED REQUEST FROM A USER OVER THE INTERNET AS A VALID SUBSTITUTE FOR A WRITTEN SIGNATURE.

Elham Etemad	06-02-2017
Signature	**Date (dd-mm-yyyy)**

Information for Authors

AUTHOR RESPONSIBILITIES

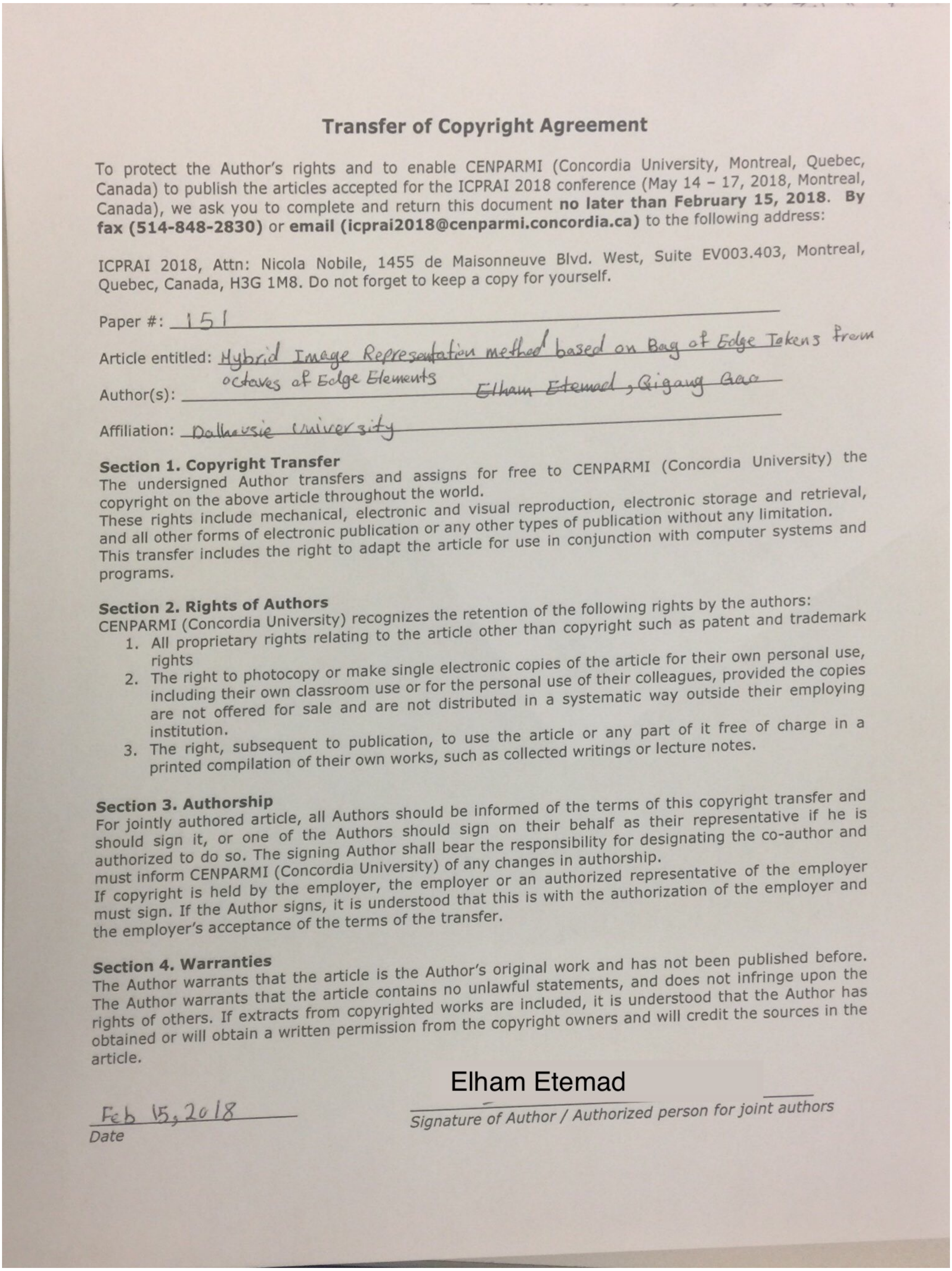
The IEEE distributes its technical publications throughout the world and wants to ensure that the material submitted to its publications is properly available to the readership of those publications. Authors must ensure that their Work meets the requirements as stated in section 8.2.1 of the IEEE PSPB Operations Manual, including provisions covering originality, authorship, author responsibilities and author misconduct. More information on IEEE's publishing policies may be found at http://www.ieee.org/publications_standards/publications/rights/authorrightsresponsibilities.html Authors are advised especially of IEEE PSPB Operations Manual section 8.2.1.B12: "It is the responsibility of the authors, not the IEEE, to determine whether disclosure of their material requires the prior consent of other parties and, if so, to obtain it." Authors are also advised of IEEE PSPB Operations Manual section 8.1.1B: "Statements and opinions given in work published by the IEEE are the expression of the authors."

RETAINED RIGHTS/TERMS AND CONDITIONS

- Authors/employers retain all proprietary rights in any process, procedure, or article of manufacture described in the Work.
- Authors/employers may reproduce or authorize others to reproduce the Work, material extracted verbatim from the Work, or derivative works for the author's personal use or for company use, provided that the source and the IEEE copyright notice are indicated, the copies are not used in any way that implies IEEE endorsement of a product or service of any employer, and the copies themselves are not offered for sale.
- Although authors are permitted to re-use all or portions of the Work in other works, this does not include granting third-party requests for reprinting, republishing, or other types of re-use.The IEEE Intellectual Property Rights office must handle all such third-party requests.
- Authors whose work was performed under a grant from a government funding agency are free to fulfill any deposit mandates from that funding agency.

AUTHOR ONLINE USE

- **Personal Servers**. Authors and/or their employers shall have the right to post the accepted version of IEEE-copyrighted articles on their own personal servers or the servers of their institutions or employers without permission from IEEE, provided that the posted version includes a prominently displayed IEEE copyright notice and, when published, a full citation to the original IEEE publication, including a link to the article abstract in IEEE Xplore. Authors shall not post the final, published versions of their papers.
- **Classroom or Internal Training Use.** An author is expressly permitted to post any portion of the accepted version of his/her own IEEE-copyrighted articles on the author's personal web site or the servers of the author's institution or company in connection with the author's teaching, training, or work responsibilities, provided that the appropriate copyright, credit, and reuse notices appear prominently with the posted material. Examples of permitted uses are lecture materials, course packs, e-reserves, conference presentations, or in-house training courses.
- **Electronic Preprints.** Before submitting an article to an IEEE publication, authors frequently post their manuscripts to their own web site, their employer's site, or to another server that invites constructive comment from colleagues. Upon submission of an article to IEEE, an author is required to transfer copyright in the article to IEEE, and the author must update any



Transfer of Copyright Agreement

To protect the Author's rights and to enable CENPARMI (Concordia University, Montreal, Quebec, Canada) to publish the articles accepted for the ICPRAI 2018 conference (May 14 – 17, 2018, Montreal, Canada), we ask you to complete and return this document **no later than February 15, 2018**. **By fax (514-848-2830)** or **email (icprai2018@cenparmi.concordia.ca)** to the following address:

ICPRAI 2018, Attn: Nicola Nobile, 1455 de Maisonneuve Blvd. West, Suite EV003.403, Montreal, Quebec, Canada, H3G 1M8. Do not forget to keep a copy for yourself.

Paper #: 151

Article entitled: Hybrid Image Representation method based on Bag of Edge Tokens from octaves of Edge Elements

Author(s): Elham Etemad, Qigang Gao

Affiliation: Dalhousie University

Section 1. Copyright Transfer
The undersigned Author transfers and assigns for free to CENPARMI (Concordia University) the copyright on the above article throughout the world.
These rights include mechanical, electronic and visual reproduction, electronic storage and retrieval, and all other forms of electronic publication or any other types of publication without any limitation.
This transfer includes the right to adapt the article for use in conjunction with computer systems and programs.

Section 2. Rights of Authors
CENPARMI (Concordia University) recognizes the retention of the following rights by the authors:

1. All proprietary rights relating to the article other than copyright such as patent and trademark rights
2. The right to photocopy or make single electronic copies of the article for their own personal use, including their own classroom use or for the personal use of their colleagues, provided the copies are not offered for sale and are not distributed in a systematic way outside their employing institution.
3. The right, subsequent to publication, to use the article or any part of it free of charge in a printed compilation of their own works, such as collected writings or lecture notes.

Section 3. Authorship
For jointly authored article, all Authors should be informed of the terms of this copyright transfer and should sign it, or one of the Authors should sign on their behalf as their representative if he is authorized to do so. The signing Author shall bear the responsibility for designating the co-author and must inform CENPARMI (Concordia University) of any changes in authorship.
If copyright is held by the employer, the employer or an authorized representative of the employer must sign. If the Author signs, it is understood that this is with the authorization of the employer and the employer's acceptance of the terms of the transfer.

Section 4. Warranties
The Author warrants that the article is the Author's original work and has not been published before.
The Author warrants that the article contains no unlawful statements, and does not infringe upon the rights of others. If extracts from copyrighted works are included, it is understood that the Author has obtained or will obtain a written permission from the copyright owners and will credit the sources in the article.

Feb 15, 2018
Date

Elham Etemad
Signature of Author / Authorized person for joint authors

IEEE COPYRIGHT AND CONSENT FORM

To ensure uniformity of treatment among all contributors, other forms may not be substituted for this form, nor may any wording of the form be changed. This form is intended for original material submitted to the IEEE and must accompany any such material in order to be published by the IEEE. Please read the form carefully and keep a copy for your files.

Image Representation Using Bag of Perceptual Curve Features

Elham Etemad, Dalhousie University, Canada; and Qigang Gao, Dalhousie University, Canada

2018 International Conference on Digital Image Computing Techniques and Applications (DICTA)

COPYRIGHT TRANSFER

The undersigned hereby assigns to The Institute of Electrical and Electronics Engineers, Incorporated (the "IEEE") all rights under copyright that may exist in and to: (a) the Work, including any revised or expanded derivative works submitted to the IEEE by the undersigned based on the Work; and (b) any associated written or multimedia components or other enhancements accompanying the Work.

GENERAL TERMS

1. The undersigned represents that he/she has the power and authority to make and execute this form.
2. The undersigned agrees to indemnify and hold harmless the IEEE from any damage or expense that may arise in the event of a breach of any of the warranties set forth above.
3. The undersigned agrees that publication with IEEE is subject to the policies and procedures of the IEEE PSPB Operations Manual.
4. In the event the above work is not accepted and published by the IEEE or is withdrawn by the author(s) before acceptance by the IEEE, the foregoing copyright transfer shall be null and void. In this case, IEEE will retain a copy of the manuscript for internal administrative/record-keeping purposes.
5. For jointly authored Works, all joint authors should sign, or one of the authors should sign as authorized agent for the others.
6. The author hereby warrants that the Work and Presentation (collectively, the "Materials") are original and that he/she is the author of the Materials. To the extent the Materials incorporate text passages, figures, data or other material from the works of others, the author has obtained any necessary permissions. Where necessary, the author has obtained all third party permissions and consents to grant the license above and has provided copies of such permissions and consents to IEEE

You have indicated that you DO wish to have video/audio recordings made of your conference presentation under terms and conditions set forth in "Consent and Release."

CONSENT AND RELEASE

1. In the event the author makes a presentation based upon the Work at a conference hosted or sponsored in whole or in part by the IEEE, the author, in consideration for his/her participation in the conference, hereby grants the IEEE the unlimited, worldwide, irrevocable permission to use, distribute, publish, license, exhibit, record, digitize, broadcast, reproduce and archive, in any format or medium, whether now known or hereafter developed: (a) his/her presentation and comments at the conference; (b) any written materials or multimedia files used in connection with his/her presentation; and (c) any recorded interviews of him/her (collectively, the "Presentation"). The permission granted includes the transcription and reproduction of the Presentation for inclusion in products sold or distributed by IEEE and live or recorded broadcast of the Presentation during or after the conference.
2. In connection with the permission granted in Section 1, the author hereby grants IEEE the unlimited, worldwide, irrevocable right to use his/her name, picture, likeness, voice and biographical information as part of the advertisement, distribution and sale of products incorporating the Work or Presentation, and releases IEEE from any claim based on right of privacy or publicity.

BY TYPING IN YOUR FULL NAME BELOW AND CLICKING THE SUBMIT BUTTON, YOU CERTIFY THAT SUCH ACTION CONSTITUTES YOUR ELECTRONIC SIGNATURE TO THIS FORM IN ACCORDANCE WITH UNITED STATES LAW, WHICH AUTHORIZES ELECTRONIC SIGNATURE BY AUTHENTICATED REQUEST FROM A USER OVER THE INTERNET AS A VALID SUBSTITUTE FOR A WRITTEN SIGNATURE.

Elham Etemad	06-10-2018
Signature	**Date (dd-mm-yyyy)**

Information for Authors

AUTHOR RESPONSIBILITIES

The IEEE distributes its technical publications throughout the world and wants to ensure that the material submitted to its publications is properly available to the readership of those publications. Authors must ensure that their Work meets the requirements as stated in section 8.2.1 of the IEEE PSPB Operations Manual, including provisions covering originality, authorship, author responsibilities and author misconduct. More information on IEEE's publishing policies may be found at http://www.ieee.org/publications_standards/publications/rights/authorrightsresponsibilities.html Authors are advised especially of IEEE PSPB Operations Manual section 8.2.1.B12: "It is the responsibility of the authors, not the IEEE, to determine whether disclosure of their material requires the prior consent of other parties and, if so, to obtain it." Authors are also advised of IEEE PSPB Operations Manual section 8.1.1B: "Statements and opinions given in work published by the IEEE are the expression of the authors."

RETAINED RIGHTS/TERMS AND CONDITIONS

- Authors/employers retain all proprietary rights in any process, procedure, or article of manufacture described in the Work.
- Authors/employers may reproduce or authorize others to reproduce the Work, material extracted verbatim from the Work, or derivative works for the author's personal use or for company use, provided that the source and the IEEE copyright notice are indicated, the copies are not used in any way that implies IEEE endorsement of a product or service of any employer, and the copies themselves are not offered for sale.
- Although authors are permitted to re-use all or portions of the Work in other works, this does not include granting third-party requests for reprinting, republishing, or other types of re-use.The IEEE Intellectual Property Rights office must handle all such third-party requests.
- Authors whose work was performed under a grant from a government funding agency are free to fulfill any deposit mandates from that funding agency.

AUTHOR ONLINE USE

- **Personal Servers**. Authors and/or their employers shall have the right to post the accepted version of IEEE-copyrighted articles on their own personal servers or the servers of their institutions or employers without permission from IEEE, provided that the posted version includes a prominently displayed IEEE copyright notice and, when published, a full citation to the original IEEE publication, including a link to the article abstract in IEEE Xplore. Authors shall not post the final, published versions of their papers.
- **Classroom or Internal Training Use.** An author is expressly permitted to post any portion of the accepted version of his/her own IEEE-copyrighted articles on the author's personal web site or the servers of the author's institution or company in connection with the author's teaching, training, or work responsibilities, provided that the appropriate copyright, credit, and reuse notices appear prominently with the posted material. Examples of permitted uses are lecture materials, course packs, e-reserves, conference presentations, or in-house training courses.
- **Electronic Preprints.** Before submitting an article to an IEEE publication, authors frequently post their manuscripts to their own web site, their employer's site, or to another server that invites constructive comment from colleagues. Upon submission of an article to IEEE, an author is required to transfer copyright in the article to IEEE, and the author must update any previously posted version of the article with a prominently displayed IEEE copyright notice. Upon publication of an article by the IEEE, the author must replace any previously posted electronic versions of the article with either (1) the full citation to the

IEEE work with a Digital Object Identifier (DOI) or link to the article abstract in IEEE Xplore, or (2) the accepted version only (not the IEEE-published version), including the IEEE copyright notice and full citation, with a link to the final, published article in IEEE Xplore.

Questions about the submission of the form or manuscript must be sent to the publication's editor.
Please direct all questions about IEEE copyright policy to:
IEEE Intellectual Property Rights Office, copyrights@ieee.org, +1-732-562-3966