Discriminative Shape Feature Pooling in Deep Convolutional Networks for Visual Classification

by

Chahna Dixit

Submitted in partial fulfilment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
December 2016

Table of Contents

List of Figures

List of Tables

Abstract

Unlike conventional handcrafted feature extractors, deep learning approach is able to extract generic image features without relying on explicit domain knowledge. More recently, there is a trend of combining handcrafted features with learned deep networks to leverage benefits of both. However, the usage of handcrafted features in existing methods are either by naïve concatenation or brute force from deep networks, and lack in actually addressing the issues of parameter quality in the network. In this research, we propose a method that enriches the deep network features by utilizing the injected perceptual shape features - Generic Edge Tokens and Curve Partitioning Points, to adjust network's internal parameter updating process. Thus, the modified convolutional neural network (CNN) is learned under the guidance of domain specific knowledge, and able to produce image representation that tightly embraces benefits from both handcrafted and deep learned features. Our experiments on several benchmark datasets show improved performance compared to the models using either handcrafted features or deep network representations alone, with reduced computation and faster convergence rate.

Acknowledgements

I would like to express my sincere gratitude to my supervisor Dr. Qigang Gao for his continuous support and motivation throughout the research work. His excellent guidance helped me achieve my goal for this thesis. Also, I would like to thank Dr. Gang Hu for his contribution in the research work in terms of experiments; I appreciate his genuine efforts in helping me during the time of the research and writing of the thesis as well.

Further, I am also grateful to Elham Etemad – PhD candidate of the IPAMI lab at Dalhousie University, who helped me learn Caffe and gave her time in providing solutions for the problems I faced during my research. A very special thanks to my family and friends for their encouragement throughout my work.

Chapter 1

Introduction

1.1 General Domain Background

Humans have the ability to understand their surrounding environment and interpret the structure of objects visible to the eye, which is known as visual perception. Humans can differentiate between several objects by gaining a thorough knowledge of its characteristics. For instance, given a flower, they can learn about its shape, color, texture and other such features which help them perceive the information. Hence, the human brain tends to capture an image of what they see and process information for learning purposes, so that the next time any similar object is seen, they can identify it immediately. This principle of the human visual system is adopted by computer vision, a field concerned with how the computers can process information from images and gain better understanding. The goal of computer vision is to make the computers to emulate human vision by learning and making inferences from visual input provided in the form of images or a sequence of images (video). Some typical tasks of computer vision include scene understanding, object recognition, automated medical image analysis for diagnosis, visual tracking, etc. To perform such tasks, computer vision employs image processing and machine learning algorithms which are similar to humans recognizing peculiar characteristics of the object/image and learning about its unique structure for future identification.

The results from machine learning algorithms solely rely on image processing techniques used. With the development in technology, various mobile phones with high resolution cameras and powerful lenses for the digital cameras are available today, which contribute to the increasing number of image data daily. The growing volume of image repository provides room for research to exploit the information rich image content. To retrieve meaningful information from images, image processing algorithms and techniques are fundamental. Image processing aims at providing better visual information for visual computing tasks. The tasks of image processing are either related to better image looks, noise removal or compact and meaningful representation, called image feature representation. The success of any computer vision task largely relies on image feature representation; hence image processing should extract features that contain high-level information and have semantic meaning.

Feature representation is a reduced representation of images that the computer can use for solving an application-specific computational task. Features are actually descriptors which describe the image. The features obtained through feature extraction are supposed to be informative in nature, non-redundant and providing concise information about the image content. The performance of any visual task highly depends on how effectively image features are represented. Hence, feature representation becomes an important step for various computer vision tasks like image retrieval, object recognition, image classification, scene understanding etc.

Image retrieval is a problem of browsing and retrieving images from a large database of images. When a query image is given as input to the image retrieval system, features of the query image are extracted using a feature extraction technique. These features are compared to a feature database corresponding to the image database using a similarity metric to obtain the images similar to query image. On the other hand, scene understanding is about describing the scene in general indicating what objects are present in the scene along with the semantic meaning. The objects in the scene are identified by extracting its features. This shows that importance of extracting appropriate features.

This research aims at improving feature representation for image classification, a task of assigning labels to the images from among a predefined set of categories. The process of image classification involves image pre-processing, segmentation, feature extraction and classification. It consists of a training phase and a testing phase. In the training phase, the input image data called the training set builds a model through learning, for prediction of unseen images from the test set. The learning is done with the help of feature extraction. The unique features of each image are extracted and the model learns according to the specified labels or groups objects sharing similar properties. Later, in the testing phase when an unseen image is passed to the model, the label for that image is predicted based on its features. Image classification aims at achieving higher accuracy in terms of labelling the image, and feature representation of the image is an integral part contributing greatly to better prediction results. The application areas of image classification include human-computer interaction, video surveillance, biometrics etc.

1.2 Problem Statement

Over the past few decades, researchers have been working on improving the results on image classification. To achieve better classification results, most of them emphasize on improving the way features are represented. The challenging part of this task is adopting the appropriate feature extraction technique. The feature representation techniques are normally categorized into two different groups: Handcrafted feature representation and Learned feature representation e.g. via deep learning [1].

The workflow of handcrafted feature representation technique is shown in Figure 1.1 describing the handcrafted feature representation obtained through handcrafted feature extraction technique from input image data and domain knowledge, which is then utilized for any computer vision task. Handcrafted features are the features of the image that are designed manually and the technique is further divided into local image descriptors and global image descriptors. The local image descriptor describes patches within the image; whereas the global image descriptor describes the image as a whole [2]. Some of the local image descriptors are SIFT [3], SURF [4], HOG [5]. The local image descriptors like SIFT, SURF work by selecting keypoints in the images and then describing them. These descriptors initially select a set of interest points from the entire image which are called keypoints. The keypoints are some peculiar points in the image like point of change in color intensity or corner points of the objects. Then, the image is divided in the form of patches, where each patch is a small area around the detected keypoints [3]. The patches are described with a descriptor using some specific characteristics such as color, texture, shape, etc.

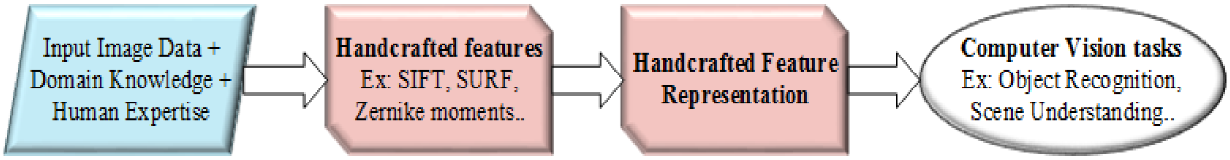


Figure 1.1 Workflow of handcrafted feature representation technique.

The local image descriptors are used in Bag of keypoints [6] technique (inspired from text categorization) for constructing semantic image representations. Bag of keypoints is also known as Bag of Visual Words (BoVW). The model involves constructing large vocabulary of many visual words and then representing each image as a histogram of frequency words that are present in the image. More details of the BoVW model are given in Chapter 2.

Apart from the BoVW model, global representation of image can be obtained with the help of global image descriptors. Color, shape and texture are the global image descriptors. Color is one of the most important features of images and it plays a major role for humans in recognizing objects. Color features can be extracted based on the color space RGB, HSV, HSL or any other [7]. Color histogram, color moments, color coherence vectors are the different color features, of which color histogram is the most common method [8]. Texture is a group of pixels possessing certain characteristics. The texture features can be classified as spatial texture features and spectral texture features [7] [9]. The different types of texture features include Gabor filter, wavelet transform, co-occurrence matrices, random fields. Shape of an object allows humans to identify the structure and shape feature extraction techniques can be classified as contour-based or region-based. The contour-based methods focus on the boundary of the object while the region-based methods consider the entire region which includes boundary as well as the interior [8]. Shape-based feature extraction techniques include Hough transform [10], Zernike moments [11] etc.

The handcrafted feature representation requires explicit prior domain knowledge of input data, and so they are robust in terms of any rotational or translational variance in the image. The encoding of extracted local features remove noise and the dimensionality reduction or pooling down-samples the feature representation while preserving spatial layout of the object in the image. The handcrafted feature representation techniques have dominated in the past by achieving huge success in image classification. However, they require human expertise and are domain driven and application-specific; meaning if one type of feature works well for a particular application, the same feature may not work well for any other application.

The recent developments in deep learning and the phenomenal success of deep networks in various applications like character recognition, object recognition, etc., have replaced the handcrafted feature representation techniques [12]. The workflow of deep learning techniques is shown in Figure 1.2 in which the learned feature representation from deep networks is given to any computer vision task.

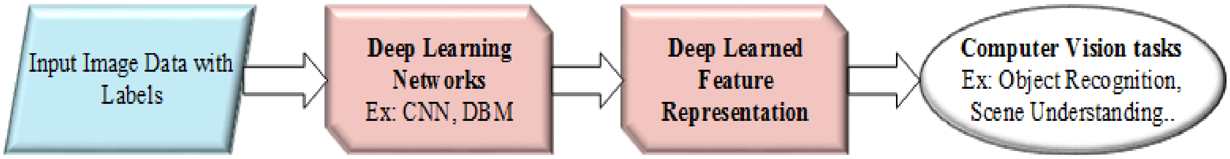

Input Image Data with Labels
Deep Learning Networks Ex: CNN, DBM
Deep Learned Feature Representation
Computer Vision tasks Ex: Object Recognition, Scene Understanding..

Figure 1.2 Workflow of deep learning technique.

The deep learning technique for feature representation is different than the handcrafted feature representation techniques discussed above. The deep networks consist of many hidden layers (more than 2) that learn hierarchy of features directly from raw pixels of input image. The low-level features of image are extracted from the earlier layers of the network, and high-level features are extracted from the later layers. Among various deep learning architectures including Convolutional Neural Networks (CNN), Deep Belief Networks (DBN), Recurrent Neural Networks (RNN) etc. [13], CNNs seem to work better for learning features for image classification or recognition [14].

Recent works have shown deep networks being able to extract generic features [15] which are beneficial in classification and recognition tasks [16] [17]. The deep learning techniques do not require explicit prior domain knowledge; instead it can directly or indirectly learn the features from labeled data or pre-trained models that share intrinsic properties the handcrafted features cannot fetch. However, unlike handcrafted features that have clear definition about how the visual cues are constructed, deep networks remain a 'mystery' in terms of internal parameter updating process and hierarchy feature learning. In spite of giving remarkable performance, the unclear process of tuning millions of parameters on raw image pixels due to the internally shuffled order of input images, creates barriers for us to understand how the features are learned, and what steps are required to improve the performance. For example, the computational cost for current hardware is high because of the need to train millions of parameters, of which some might not be necessary; this not only increases the convergence time, but also causes overfitting. In order to overcome overfitting, dropout method is often applied to reduce the parameters. But it is difficult to tell whether the dropped parameters are innocent or not. To compensate the victim parameters, using very deep networks with millions of more parameters may ease the problem, but it brings even higher computational cost.

In summary, the limitations and benefits of both the representation techniques makes it difficult to select an appropriate technique that will be suitable for any given application domain. This arises the need to exploit the techniques to achieve better performance.

1.3 Research Motivation and Objectives

Considering the benefits of both the methods as discussed in the previous sub-section, some researchers realized that handcrafted features can be combined with deep learned features to achieve better performance [18] [12]. However, existing methods of combining the features mainly simply concatenate both the final features for classification without considering the intermediate features of deep networks and hence, these methods lack in actually addressing the issues of deep networks. Therefore, their achievements are limited in some specific domains.

In the same spirit, we combine handcrafted features with the features from deep learning technique attempting to overcome the issues of deep networks. We propose a novel method 'Discriminative Shape Feature Pooling' (DSFP) that utilizes the power from injected handcrafted features into the deep network, to adjust the internal parameter updating process. An overview of the proposed feature combination approach is shown in Figure 1.3, in which the higher level pooling layer of the deep network is modified with the domain knowledge of handcrafted features to obtained the modified learned representation. The modified feature representation of the image is then used for classification of the image. The pooling mechanism for the higher-level image features is similar to the visual feature selection processing in human brain [19]. Shape-based perceptual features – Generic Edge Tokens (GETs) and Curve Partitioning Points (CPPs) [20] are used in Convolutional Neural Network's (CNN's) new pooling strategy. More details of proposed method can be found in Chapter 3.

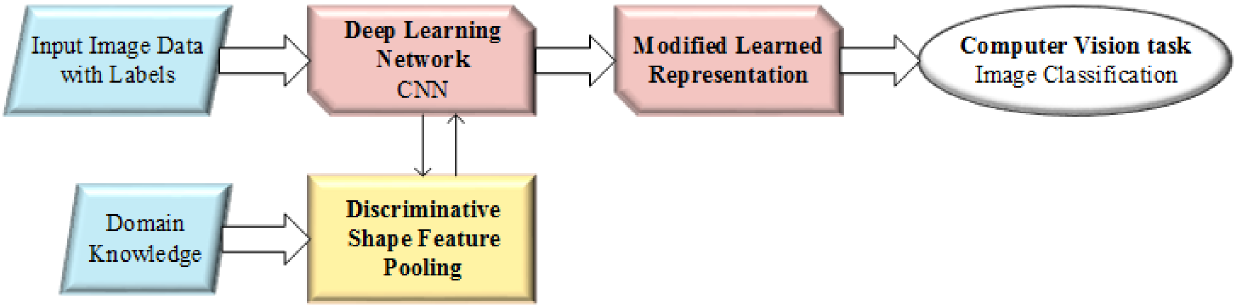

Input Image Data with Labels
Deep Learning Network CNN
Modified Learned Representation
Computer Vision task Image Classification
Domain Knowledge
Discriminative Shape Feature Pooling

Figure 1.3 Workflow of proposed feature combination approach.

While most of the researchers combine features from the last fully connected layer of CNN with handcrafted features, our method is different in two ways: (i) We modify the higher-level pooling layer of the network which is the layer just before the fully connected layers. The features obtained from the fully connected layers are in the form of vectors, whereas the higher-level pooling layer represents important objects of the image. (ii) Unlike other researchers simply concatenating the handcrafted features with deep learned features, we modify the pooled feature map with the help of injected handcrafted features.

1.4 Research Contributions

In summary, the proposed method has following contributions:

- The experimental results on image classification show improvement in performance, especially for natural scenes and living beings' categories.
- With the guidance of handcrafted features, the deep network model has reduced learning curve, i.e. fast convergence.

- The framework is generic; i.e. open to other handcrafted features and deep network architectures too.
- The modified network has relatively fewer parameters in comparison to other deep networks with many layers showing almost similar performance as ours.

The rest of the thesis is organized as follows:

Chapter 2 provides detailed explanation about the background and work done on handcrafted features and deep learning methods. It also gives a literature survey and comments about the research work based on combination of both the methods. Chapter 3 explains about the Alexnet based convolutional neural network, GET-CPP details and proposed methodology along with the algorithm and system design. In Chapter 4, we show the experiment results on different datasets and a detailed analysis of the obtained results. A comparison with already existing methods is also shown. In the end, Chapter 5 gives a conclusion regarding the results of proposed method and possible scope of improvement for future.

Chapter 2

Background and Related Work

The performance of an image classification problem is inherently constrained by the feature representation method used to represent the images. An image is considered as raw data in the form of pixels. When dealing with the classification of an image, raw input data in the form of pixels is not helpful to directly tackle the problem. Due to this, image needs to be represented in a better way such that unique characteristics of image are extracted and this information can then be used to classify it.

Researchers have emphasized on improving the representation of images in order to gain high classification performance, and even today good amount of work is being done for the same purpose. A considerable improvement in results has been obtained in the last ten years. The researchers adopt one or both among the two trends of image feature representation: Handcrafted feature representation and/or Learned feature representation (Deep Learning) [1].

This chapter gives a survey on the image classification methods and a detailed explanation of handcrafted feature representation techniques and deep learning techniques. While most of the researchers use either the handcrafted feature representation or deep learning technique, few of them have attempted to combine both the representation methods. The approach of combining features is discussed along with the importance of pooling in representing the feature.

2.1 Image Classification

Image classification is the task of classifying images into corresponding categories based on its visual content. A training set consisting of many input images along with the image labels is used to train a classifier; i.e. the classifier learns unique characteristics specific to the images and their corresponding labels. The classifier then predicts labels for an unseen set of images and the performance of classifier is evaluated by comparing the predicted labels to actual labels of the image.

Being related to the concept of human vision, image classification or any other visual task becomes challenging in various aspects. Some of these aspects could be illumination changes, occlusion, scale, image deformation, background clutter, intra-class variation and viewpoint variation [21]. When images are captured from different devices, the resolution of even the same images may vary. Moreover, the lighting conditions differ with place and time. Same category images may not be classified correctly because of difference in their viewpoints. Sometimes, the background of image could be more dominating than the object of interest itself, or it is also possible that only a small part of object is visible which makes it difficult to identify the image. Thus, accurate representation of an image becomes challenging, which ultimately affects the prediction results.

2.2 Handcrafted Feature Representation

Many researchers tend to design the features of an image manually. In that, the researchers scrutinize peculiar characteristics of the image and find ways to represent them. Since, the handcrafted features are designed by humans, the techniques of representation rely on the knowledge and expertise of the humans. However, the techniques do not depend on labeled data and have efficient training algorithms [1].

Handcrafted features can be categorized into two different feature types: Local image features and Global image features, which are discussed in the following sub-sections.

2.2.1 Local Image Features

The local features are associated with different parts or regions of an image, which implies that these features are computed at different points in the image [2]. A local feature can be considered as an image pattern that differs from its immediate neighborhood [22]. The approach for extraction of local features involve detecting several points of interest from the image, called keypoints and describing those keypoints to obtain feature vector of image which is matched across images [23]. Feature detectors are used for detecting keypoints in the image, and these keypoints are typically either corners or centers of blob-like structures. In order to match the detected keypoints across images, feature descriptors are computed. Feature descriptors are some kind of vectors of values that represent patches around interest points [2].

Normally, the keypoints are desired to be repeatable across multiple views of an object and should be easy to extract. The feature descriptors should be such that they are distinctive, robust to occlusion and clutter [24]. Efforts have been made to extract relevant and useful local features and for that purpose, several detectors and descriptors have been designed and used in different ways for classifying images.

A naïve approach for feature detection was proposed by C. Harris and M. Stephens [25], called the Harris corner detector for detecting the corners and edges of objects in the image. The basic idea behind the Harris corner detector is that it gives a mathematical approach to determine whether the image patches represent an edge, corner or a flat region. The type is identified by calculating the change in the intensity of different patches obtained by slightly shifting the original image patch in different directions. H. Kim et al. [26] used an improved version of Harris corner detector to classify the breast mammogram image as normal or abnormal. Since medical image processing focuses on the intensity values in the form of features, the use of Harris corner detector in the classification improved the results. A few other researchers have also worked upon the modification of the detector for better performance. During the time when Harris corner detector was proposed, it seemed to be a well-known method for detecting interest points in an image, but its popularity decreased with the proposal of other better feature detectors, some of which worked as both, detectors and descriptors.

D. Lowe [23] proposed a new approach which extracts highly distinctive features called Scale Invariant Feature Transform abbreviated as SIFT. The features extracted by SIFT are invariant to scale and rotation and robust to changes in viewpoints, illumination, noise or clutter.

In order to detect the keypoints, SIFT detector first generates a scale space with the help of scale space filtering [27], so that the detected locations are invariant to scale. The generated scale space extrema is approximated using the Laplacian of Gaussian function and the keypoints are detected by calculating the difference-of-Gaussian from the difference of two nearby scales. The keypoints that are detected are still large in numbers, out of which some of them are not good enough and need to be discarded by comparing to its eight neighbors in the current image and nine neighbors in the scale above and below. The filtered out keypoints are assigned an orientation and the keypoint descriptor can be represented relative to this orientation which is helpful to achieve rotational invariance. The keypoint descriptor computes the gradient magnitude and orientation of each image sample point and assigns a weight by Gaussian weighting function. The samples are then accumulated into orientation histograms to create final feature descriptor.

However, the computation of SIFT feature vector is time consuming and requires more storage space because of its length. Also, SIFT does not work well in conditions of large illumination changes and non-rigid deformation. To overcome these problems, Y. Ke et al. [28] proposed PCA-SIFT which computes the local image gradient of a patch and projects it using the Eigen space to give a compact feature vector. The use of Principal Component Analysis (PCA) in PCA-SIFT reduces the size of the feature vector. The PCA-SIFT based local descriptors were more compact than the original SIFT.

Yet another feature detector and descriptor was introduced by H. Bay et al. [4] – Speeded Up Robust Features (SURF) with some changes as compared to that of SIFT. The SURF detector detects the keypoints using Hessian matrix approximation. The computation time of SURF detector is reduced because of the use of integral images to build the scale space. Whereas, Haar wavelet responses are used to assign the orientation and then the keypoint descriptor is formed centered around the keypoint.

N. Dalal and B. Triggs [5] proposed a feature descriptor called Histogram of Gradients (HOG) which divides the image into small spatial regions called cells, wherein a histogram of gradient directions is formed for each pixel in the cell. These histograms are concatenated to describe the image. The key reason behind using HOG is that the local object appearance and shape can be characterized well by distribution of local intensity gradients or edge directions. Later on, V. Chandrasekhar et al. [29] came up with Compressed Histogram of Gradients (CHOG), a low bit-rate descriptor similar to HOG in which they represent the histogram of gradients as tree structures that can be compressed efficiently. They performed Vector Quantization of the gradient distribution into a small set of bins.

High quality features are detected by a high-speed, machine learning based corner detection method known as FAST (Features from Accelerated Segment Test) detector [30]. This detector works on the principle of Segment Test Criterion in which an image pixel is selected to be identified as interest point or not, and the test decides whether the selected pixel point represents a corner or not based on the intensity value, neighboring pixel values and some threshold selected by the user. The detected corners are then classified based on the decision tree classifier. While FAST detector is suitable for real-time applications, faster than few other corner detectors; it is not robust to high level noise and is dependent on a threshold value.

The Binary Robust Independent Elementary Features (BRIEF) [31], proposed by M. Calonder et al. gives better or similar performance and is faster compared to the SURF descriptor. The BRIEF descriptor computes binary strings by comparing the pixel intensities of the location pairs chosen from an image patch. These descriptors in the form of binary strings can be matched using Hamming distance which is an efficient computation method.

S. Leutenegger et al. proposed a novel keypoint detector and descriptor called Binary Robust Invariant Scalable Keypoints (BRISK) [32] which reduced the computational cost in comparison to the SIFT and SURF. BRISK is associated with FAST-based detector, and it computes binary descriptor strings using the circular sampling patterns. The sampling patterns are applied to the neighborhood of each keypoint to get gray values and the feature characteristic direction is determined by processing the local intensity gradients. The oriented sampling patterns are used to obtain pairwise brightness comparison results.

An efficient alternative to SIFT and SURF was proposed by E. Rublee et al. based on the FAST keypoint detector and a modified form of the BRIEF descriptor which was called Oriented FAST and Rotated BRIEF (ORB) [33]. Considering the low cost and good performance of both the techniques, ORB tries to modify the techniques to eliminate some limitations, mainly focusing on the issue of rotational invariance in BRIEF descriptor. In order to tackle that issue, ORB uses an orientation compensation mechanism and learns optimal sampling pairs.

The Fast Retina Keypoint (FREAK) [34] descriptor proposed by A. Alahi et al. is inspired by the human visual system, especially the retina which computes a cascade of binary strings by comparing the image intensities over a retinal sampling pattern. It is faster in computation and requires less memory. The authors claim that FREAK descriptor is more robust than SIFT, SURF, or BRISK.

In order to represent images for classification, the local image features are used in the Bag-of-Visual-Words technique (BoVW) [6], which is a technique inspired from text categorization [35]. The basic pipeline for the BoVW model is shown in Figure 2.1.

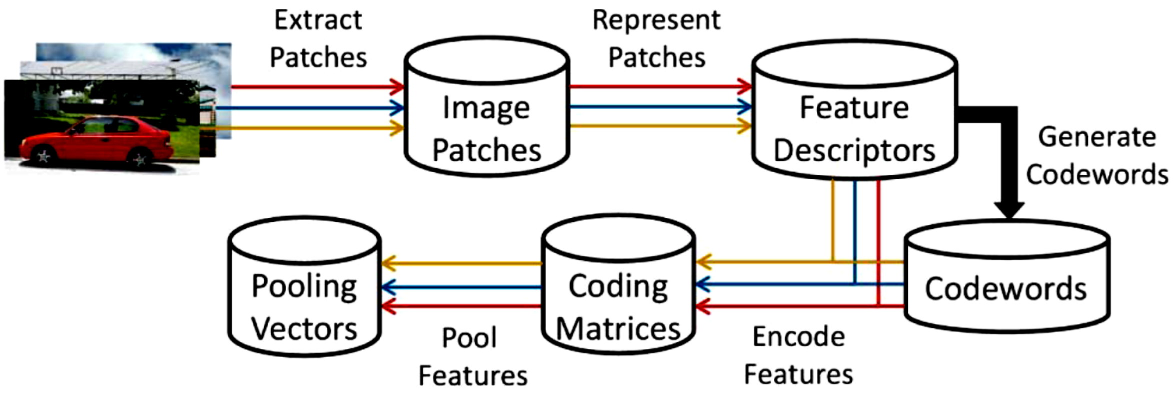


Figure 2.1 Bag of visual words model pipeline [36].

The five basic steps of BoVW model as shown in Figure 4 are: extracting image patches, representing the image patches, generating codewords, encoding features and pooling the features. Given an image, the image patches can be extracted in many ways like regular grid method [37], feature detectors [25] [23], random sampling or segmentation-based patches. The pixels of the image patches are then represented with the help of feature descriptors like SIFT, HOG, SURF etc. A subset of codewords is sampled from the set of all the feature descriptors and a ‘visual vocabulary’ or a ‘codebook’ is formed by clustering, for instance K-means clustering [38], in which the cluster centers act as visual words (or codewords). Each feature descriptor then generates a code vector/feature vector using feature encoding methods like super-vector coding [39], sparse coding methods [40] etc. The length of the feature vector is equal to the number of visual words in the codebook. The final feature representation is obtained by pooling the feature vectors. As a result of the BoVW model, the images are represented as frequencies of visual words. Using this representation, a classifier is learned using machine learning. Given a new image, the descriptors are extracted, and for each descriptor the model computes its nearest neighbor in the codebook and creates histogram containing the frequencies of visual words.

The benefits of using the BoVW model are that minor changes in the position and orientation of the object in the image does not affect the results, and it produces a fixed length feature vector irrespective of the number of keypoint detections [41]. The use of feature descriptors in the model makes the visual words invariant to illumination and affine transformation. The BoVW model has shown commendable performance for classification of images. T. Deselaers et al. [42] have used the BoVW model for classification of adult images to filter them from the network traffic, and their results show that the model outperforms state-of-art methods in that particular task. On the other hand, T. Li et al. [43] have proposed a modified version of BoVW known as Contextual Bag-of-words (CBOW) representation for image classification which considers contextual relations between the local patches.

2.2.2 Global Image Features

The image classification systems use global features to represent the entire image. These features produce compact representations of images and each image corresponds to a point in a highdimensional feature space [2]. The global features of an image include color, texture, shape or perceptual information.

Colors are an important part of the human vision and is a wavelength-dependent perception [44]. The images can be represented using various color models such as RGB, HSV, YCrCb etc. The different types of color descriptors include color histogram, color moments and color coherence vector. The most common among them is color histogram, which is the representation of distribution of colors in the image based on the color space. The histogram is a plot of the intensity values according to the color space versus the number of pixels at that particular value. In [45], the author used features of color histogram on the YCbCr color space to classify the images and claimed the approach to be efficient, quick and robust.

While color is a property of the single image pixel, the texture of an image is measured using group of pixels. The analysis of texture can be done using techniques like Gabor filter, Local Binary Pattern (LBP), Markov Random field model etc. [46]. M. Yang et al. [47] considered Gabor features for sparse representation based classification for face recognition. Gabor filters are group of wavelets and can be viewed as a sinusoidal plane of particular frequency and orientation, modulated by a Gaussian function [48]. The use of Gabor features in their method reduced the computational cost producing higher recognition rates.

Apart from color and texture, shape is also an essential cue for human vision and thus as an image feature for classification. The techniques include Hough transform, co-occurrence matrix, moment invariants etc. A. Aggarwal et al. [11] proposed a method to represent medical images by extracting the features using Zernike moments. The use of Zernike moments is beneficial for representing information which possess minimum redundancy, insensitive to noise and hence becomes a good choice as a global feature descriptor.

Perceptual features are high-level features which relate to the perception of human visual system. The authors of [20] introduced a package called the Perceptual Curve Partitioning and Grouping (PCPG) which extracts the generic edge tokens (GETs) and curve partitioning points (CPPs) by scanning the objects in the image. Generic edge tokens (GETs) are perceptually unique curve segments, and an object can be represented in the form of GETs by considering its edge information. Two adjacent GETs are connected with a curve partitioning point (CPP); the point which indicates the change in monotonicity. More details about the GET/CPP is provided in Chapter 3.

Because a single type of feature from among the color, texture and shape cannot represent the image completely, the researchers combine some or all of these global features to represent the image. D. Sudarvizhi [49] represents images by combining all three features using color HSV histogram, color moments, color correlogram, Zernike moments as well as Daubechies wavelet transform. The image representation was used for image retrieval systems and it provided improved results for retrieval purposes.

2.2.3 Combining Local and Global Image Features

The fusion of local and global image features is also a trend for image classification. [50] uses a hierarchical approach to combine the local and global features for classification of remote sensing images. The authors use Gabor filters as local descriptor to extract the texture features, along with SIFT descriptor to get the local image features. The combined approach showed improved results for classifying the images. Rather than proposing a new method by combining the local and global features, some other researchers propose descriptors based on the combination which can be used for various applications. For instance, CSIFT [51] is a SIFT descriptor which has the property of color variance. The SIFT descriptor was designed for grayscale images and hence, to obtain better image information in terms of color, CSIFT was proposed that combines the color as well as geometrical information and it proved to be more robust that the original SIFT descriptor in terms of color and photometric variations.

2.3 Learned Feature Representation

While handcrafted feature representation focuses on designing the features by hand, the other technique to represent the features makes the system to learn the feature on its own. This technique is known as the learned feature representation; for example, via deep learning. Deep learning is about training the computer systems in such a way that it can automatically extract the features of the image for its representation. The multiple processing layers in the deep learning architecture are responsible for extracting feature hierarchy as we go from lower layer to the higher layers [52]. Each successive layer in the architecture uses the output of its previous layer as the input. In the recent times, learning feature representation through deep learning architecture has gained popularity, mainly because of its automatic feature-learning ability and improved performance results.

The current trend for feature representation is deep learning which seems to have replaced the handcrafted feature representation method [53] because when the features are learned directly by the machines, human expertise and prior knowledge of data is barely needed. Also, the features learned in a deep fashion are helpful in providing better results. Various deep learning architectures include Deep Neural Networks (DNN), Deep Belief Network (DBN), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN), Deep Boltzmann Machines (DBM), Deep Auto-encoders [54]. A major breakthrough occurred in the field of learning features via deep networks from the work done by G. Hinton [55], and later many other researchers also gave their contribution in the field.

Among the different deep learning architectures, the Convolutional Neural Networks (CNNs) seem to work better for learning features in case of image classification or recognition [14]. Several competitions for image classification task are being held annually, the results of which provides the fact that CNNs are a trend today. The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [56] is a benchmark for classification of images on the ImageNet dataset [57] containing more than hundred object categories and millions of images. One other challenge which was being held every year until 2012 was the Pascal Visual Object Classes Challenge [58], also known as the Pascal VOC Challenge that held competitions for classification and detection tasks on the Pascal VOC dataset. The details of CNN and its contribution in terms of providing remarkable results for various challenges of image classification are provided in the next section.

2.2.1 Convolutional Neural Networks (CNNs)

Convolutional neural networks are a type of artificial neural networks having a specialized connectivity structure, with multiple stages extracting a hierarchy of features. Given the raw input data in the form of image, the CNN extracts features from low-level to high-level through several functions which are then classified in the end by the classifier in the CNN architecture. To extract the features, the CNN consists of layers of trainable convolutions and spatial subsampling which could be transformed using non-linearity functions [59]. The deeper layers of the network produce high-level, global and invariant features. CNN is considered to be a supervised training of convolutional filters by back-propagating the classification error. The basic building blocks of a CNN include convolutional layer, pooling layer, fully connected layer and loss layer.

The convolution layer is considered the core layer of the CNN and it performs convolution operation over the input volume by applying a filter (convolution kernel) to produce activation maps. The activation maps refer to the regions where the features specific to the applied filter have been detected in the input volume [60]. Since the same filter is applied to entire image, CNN needs fewer parameters and the weights are shared. During the training process of CNN, these filters are learned after every iteration. The result of convolution operation on an image is shown in Figure 2.2.

A non-linear activation function is applied to the output of the convolution layer to introduce non-linear properties in the network. The most commonly used non-linear activation function is the Rectified Linear Unit (ReLU), whereas few others could be tanh or sigmoid functions.

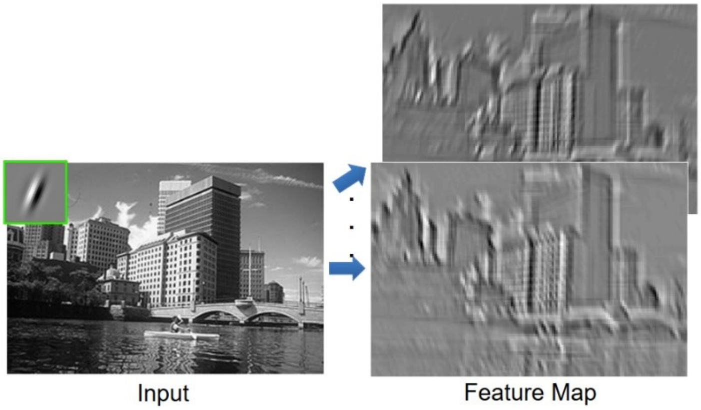


Figure 2.2 Convolution operation [61].

The pooling layer, also known as sub-sampling or down-sampling layer reduces the dimensions of feature maps while preserving important information [17]. Reduced dimensions of feature maps indicate fewer parameters in the network with decrease in computation. The pooling operation depends on type of pooling used. The reduced size feature map obtained through pooling operation is shown in Figure 2.3.

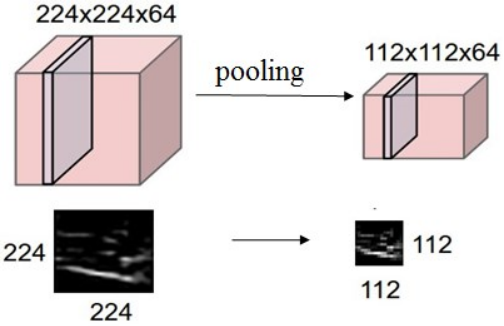


Figure 2.3 Pooling operation.

The pooling layer may be followed by a normalization layer that could be applied within or across the feature maps. Local Response Normalization (LRN) that performs a kind of 'lateral inhibition' by normalizing over local input regions is normally used.

The fully connected layer considers the input in the form a vector and produces output as a vector. All the neurons in the fully-connected layer are connected to all the neurons in the previous layer. The fully-connected layer is responsible for classification of images from the high-level features into suitable labels based on the training set. In order to train the CNN, the network uses a loss function in the loss layer which will be chosen according to the task to be performed. The loss function calculates the loss during training based on the difference in the predicted and correct labels and updates the weights of the network accordingly for training purpose.

The deep networks with multiple intermediate layers are responsible for learning complex relationships between the inputs and outputs. Although, with limited data available and more layers in the network leading to millions of parameters, the complex model perfectly fits the training data; but when the model is evaluated on any new data, the performance is poor. This problem is known as overfitting, which is an important challenge in the deep convolutional neural networks as well. Overfitting prevents the network from building generic models. The methods to reduce overfitting are called regularization methods. The various regularization methods include weight decay or L2 regularization, L1 regularization, dropout, data augmentation [62]. L2 regularization refers to adding an extra term to the cost function known as regularization term which helps in shrinking the weights. L1 regularization is similar to L2 regularization but the weights shrink by a constant amount unlike L2 regularization. On the other hand, dropout tends to randomly drop some hidden or visible units from the network [63]. Data augmentation is artificially increasing the size of the training data by various operations like cropping, flipping etc. One other important way to regularize the deep networks is transfer learning or more often known as pre-training [64]. When the CNN is initialized with random weights, the network is said to be trained from scratch. A huge amount of data is required to train the CNN from scratch and since the weight initialization is random, the training takes longer time and there are chances that the network gets stuck in poor local minima. To avoid such issues, the CNN training is done by initializing with weights from an already trained model on a large dataset like ImageNet or the model is directly used as a feature extractor for some other dataset. This phenomenon is called pre-training, a kind of regularizer that gives good weight initialization to the network leading to better local minima [65].

The benefits of convolutional neural networks like weight sharing, reduced parameters and possessing translation invariant characteristics along with higher accuracy has motivated many researchers to explore these networks for image classification. The first successful CNN architecture called LeNet-5 [16] was used for recognizing digits, postal codes etc. The LeNet-5 architecture is shown in Figure 2.4.

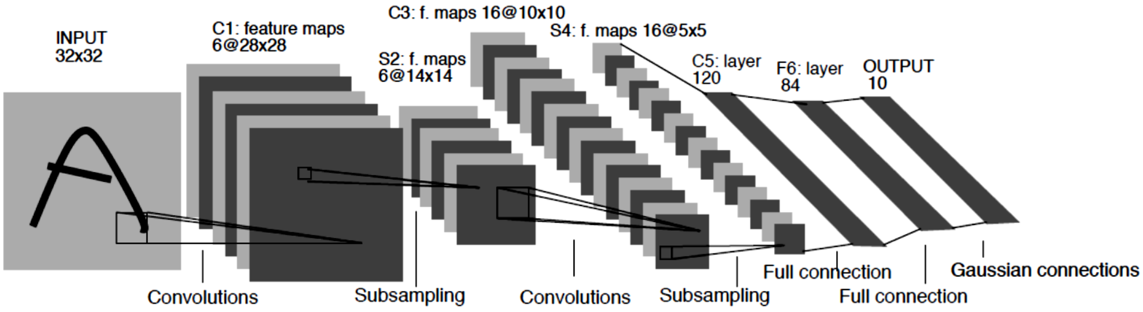


Figure 2.4 LeNet-5 CNN model [16].

A similar, but deeper architecture was proposed by A. Krizhevsky et al. [17], the first work that popularized CNNs in the field of Computer Vision; and the architecture is well known as Alexnet. The Alexnet model worked well for the classification of the ImageNet dataset, and achieved second-best position in the ILSVRC-2012 competition with significant results. Our proposed method uses the Alexnet model as baseline model and is described in detail in Chapter 3. Later on, several other CNN models such as ZF Net [66], VGGNet [67] were proposed which were still deeper or had more parameters than Alexnet. All these proposed networks aimed at obtaining better representation of images in terms of extracting better high-level features from deeper layers.

2.4 Combining Handcrafted and Learned Feature Representation

The recently emerging deep learning methods represent images by learning the features automatically from the raw input data, whereas the traditional handcrafted feature representation uses prior knowledge of the domain and humans design these features with their expertise. Considering that both these representation techniques emphasize on different aspects of data, few researchers have attempted to combine handcrafted features with machine learnt features to take benefits of both the methods.

Representing the images by combining handcrafted features and learned features is a better idea for classifying images instead of using any one type of the features. To prove this, a model was proposed for recognizing images which used Alexnet model trained by the ImageNet dataset [12]. The feature representation from second fully connected layer was extracted and combined with handcrafted features that used local SIFT features encoded using the Locality-Constrained Linear Coding (LLC) [68] method based on the Spatial Pyramid matching (SPM) [69]. The model outperformed baseline handcrafted and deep learning techniques by considering the prior knowledge of the data as well as the information of data distribution.

In terms of medical image recognition, handcrafted features were combined with CNN features for mitosis detection [70]. Considering the benefits of handcrafted features that they are inspired by domain and application specific along with the fact that although CNN gives remarkable performance for classification, its architecture is computationally complex and it requires lots of labelled data, a CNN model was proposed which was computationally simple and a cascaded strategy was developed to combine the handcrafted features in the form of color, texture and morphology features with the features from the last layer of CNN. This approach was faster and required fewer computing resources.

Grayscale images were provided as input into the CNN which decreased the number of filters in the network, since the color information was separated [18]. The decrease in number of filters reduced the network parameters. The separated color information was then exploited by using the color histogram of the images. The final feature vector of the CNN model was combined with the color histogram and then given to the classifier for classification. This way, the proposed CNN model was compact and required lesser time for computation because of reduced number of parameters. Moreover, the method gave similar results to that of the state-of-art in reduced time. Such straightforward concatenation approaches only add additional data to the classifier, rather than providing a coherent way to improve the deep learning feature representation.

Instead of simply concatenating handcrafted features with the final feature from deep network, some CNN-based approaches utilized the handcrafted features in a more comprehensive way. Handcrafted features can be supportive to CNN features; and to show this, a novel deep network called Feature Fusion Net (FFN) was proposed that adds two additional layers on top of the 7 layer CNN, which fuses color histogram and Gabor features with the last layer output of Alexnet model [71]. In this way, handcrafted features were combined into the later stage of CNN to provide supportive role for feature regularization.

Handcrafted features were combined with CNNs in a different manner to detect tumor cells in histology images [72]. The spatially constrained CNN (SC-CNN) [73] was modified in which the color and texture features were computed through scattering transform [74] and given as input along with the raw data of pixel intensities to the SC-CNN. The detection results of tumor cells were better with this combination rather than the results with CNN alone.

A recently proposed DEFEATnet [75] is a novel deep network representing sequential layers, each consisting of SIFT feature extraction followed by sparse coding and local max pooling. The final representation of the network improved performance compared to traditional methods, but the original deep learning techniques still perform much better.

2.5 Pooling Strategies

Pooling strategies play a crucial role in performance of deep network. Many researchers have tried to modify the pooling policy of the deep network to improve the representation of image for higher results. K. He et al. [19] introduced SPP-net (CNNs + Spatial Pyramid Pooling Layer) which accepts images of multiple input sizes and scales, and produces a fixed length representation of images. It showed improved results for classification and detection. Y. Gong et al. [76] performed orderless VLAD pooling on CNN representation of image patches at multiple scales to obtain generic feature for classification.

2.6 Summary

Both the feature representation techniques have their own benefits and limitations. The handcrafted features require explicit prior domain knowledge and human expertise for their designing, and once designed the feature representation can be visualized. For instance, the BoVW model can represent the visual words in the form of histograms. The semantic information can be obtained by using handcrafted features. However, these features are application specific so if a particular feature works well for one type of application, it may not work well for some other application.

On the other hand, learned feature technique, especially CNN has shown great results for image classification. CNNs are capable of learning generic features; but the success of CNN or any other learned feature technique relies on availability of large amounts of data and also it needs powerful hardware due to the computational complexity. Moreover, CNN is a complete black box, meaning the learned features are not easily interpretable.

In order to leverage the advantages from both the techniques, in the current times researchers have attempted to combine handcrafted features along with the CNN features. In combining the techniques, most researchers consider features from last layer of CNN to combine handcrafted feature techniques. While combining with the handcrafted features, the researchers try to build a lighter model of CNN which has fewer parameters so that the computation time is decreased and better results are obtained. Few other researchers modify the input of CNN with handcrafted features while, others train the CNN along with the combined features of CNN. Although these methods contribute in improving the image representation for better performance, they fail to address the issues of deep networks.

In summary, combined feature representations are able to gain better classification performance. Unlike other approaches, we inject handcrafted features into the highest level pooling layer representation of CNN which extracts important objects of images, and hence domain knowledge can be utilized as a guidance to the final CNN feature construction through the crucial pooling process.

Chapter 3

Proposed Methodology and System Design

The detailed survey on handcrafted and learned feature representation techniques in Chapter 2 highlights the recent trend of combining both the techniques for better representation of image. The idea of combination has led to higher classification results. Taking into accounts the benefits of such combination, in this research we follow similar trend to improve the feature representation of the image.

This Chapter provides explanation of the basic system components: Perceptual shape features, Alexnet architecture and importance of pooling layer. The proposed method framework along with algorithm and implementation details is explained in the later part of the chapter.

3.1 Basic Concepts and System Components

3.1.1 Perceptual Feature Representation

Various types of handcrafted features are available, which can be utilized for feature representation. In our method, we use perceptual features that carry contextual information along with semantic meaning. They utilize the laws of perceptual grouping, derived from Gestalt psychology [77] and hence are based on the characteristics of human vision. The Gestalt psychology is based on the ability of humans to visualize objects as a whole, rather than in parts or regions. Thus, the perceptual features produce more meaningful representation of images.

Perceptual feature representation has been an important area for researchers for improving the image representation. Long ago, global perceptual features were extracted from edge images that followed Gestalt principles [78]. The psychologically important features like edges, junctions were exploited to obtain saliency map that could detect the shapes based on properties of the objects. Later, in order to improve recognition accuracy of images and to employ invariance, a perceptual feature called generalized robust invariant feature (G-RIF) was proposed [79]. The G-RIF detector was a combination of a radial symmetry detector and a corner-like structure detector, while the descriptor encodes edge and hue information. G-RIF was computationally efficient and it outperformed SIFT in terms of recognition results.

Perceptual texture features along with Gabor wavelet features were used for image classification [80]. Based on the human perception, the three different perceptual texture features introduced were directionality (horizontal, vertical, diagonal), contrast of image and the granularity measurement defined as coarseness. Along with reduction in feature dimensions, the classification accuracy improved. Recently, perceptual features were utilized to distinguish between objects of different sizes in the real world [81]. Examining the perceptual properties of images contribute to better understanding about the way objects are recognized and that the perceptual information can influence the classification results as well.

Generic Edge Tokens (GETs) and Curve Partitioning Points (CPPs)

The perceptual shape features are based on the principles of Gestalt psychology, the elements of which are Generic Edge Tokens (GETs) and Curve Partitioning Points (CPPs) [20] . GETs and CPPs are considered as basic elements that describe the shape of any object and are based on human visual perception. Any object can be represented in the form of GETs and CPPs with the help of curve partitioning rules. The Perceptual Curve Partitioning and Grouping (PCPG) model extracts the GETs and CPPs [20]. Any smooth planar curve can be partitioned into connected GETs where the GETs are connected with CPPs. In general, the GETs are perceptually unique and these edge tokens could be curve segments or straight line segments. A perceptual feature hierarchy of the edge tokens is shown in Figure 3.1.

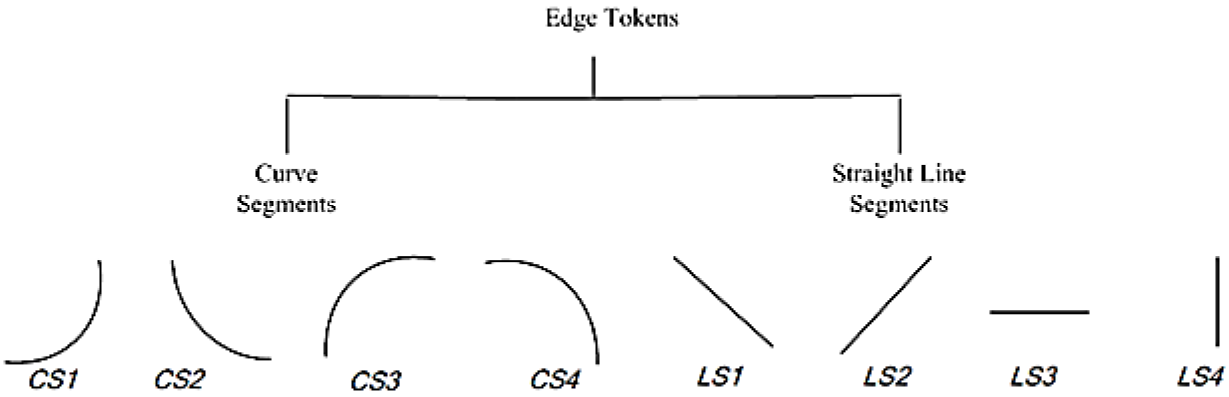


Figure 3.1 Perceptual feature hierarchy of GET.
CS – Curve Segment, LS – Line Segment

The eight categories of GETs are defined by considering the curvature and slopes of the different edge tokens. Two GETs are connected by a CPP. The set of CPPs track the path along the edges. Each CPP is a junction of two adjacent GETs, wherein the CPP indicates shape salience, i.e. the transition of perceptual shape geometry monotonicity. Based on the eight different GETs, there are eight types of CPPs as proposed by [20] which are shown in Figure 3.2. which consist of connection between two curve segments, two straight line segments and one curve segment connected to one straight line segment.

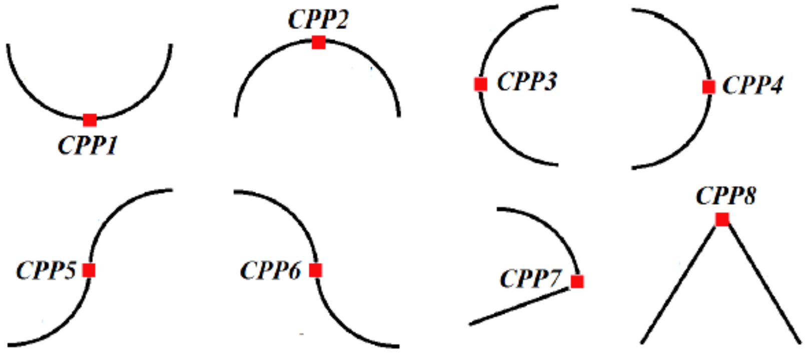


Figure 3.2 Representation of different categories of CPP.

The PCPG model extracts groups of the basic elements of the image based on the Gestalt psychology of grouping characteristics as per the human visual system. The extracted elements possess important properties of the Gestalt psychology which are similarity, continuity, closure, symmetry, proximity and simplicity [77]. An improved curve detection version was introduced by G. Hu et al. [82] called Order Preserving Arctangent Bin Sequence (OPABS) scheme, which was able to extract more precise salient shape features. Based on the monotonic properties of the different GETs, G. Hu et al. claimed that some CPPs are Strong CPPs while others are Weak CPPs (Figure 3.3), in which the Weak CPPs are hard to detect in the image and its presence is not convincing. Such Weak CPPs which would have been ignored in the PCPG model, were then detected using the novel OPABS scheme that detected better CPPs and ultimately led to better GETs corresponding to those CPPs.

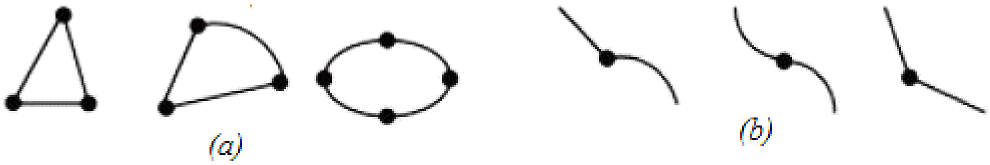


Figure 3.3 (a) Strong CPPs (b) Weak CPPs.

In our proposed method, we used the OPABS package [82] to obtain the GET and CPP information. The package provides the location of GETs and CPPs, their types and length of GET. The information regarding the CPPs and their corresponding GETs is also obtained from the package. An edge map of an example image represented in the form of GETs and CPPs according to the OPABS package is shown in Figure 3.4.

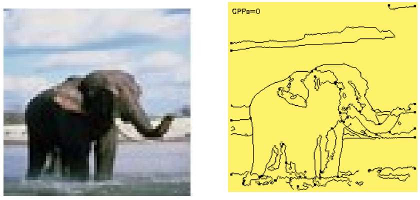


Figure 3.4 GET-CPP representation of an example image.

Having said that perceptual GET and CPP features are able to convey shape semantics meaning, our approach is open to any other handcrafted discriminative features.

3.1.2 Baseline Alexnet Architecture

The first work that popularized the use of CNNs in the field of computer vision was proposed by A. Krizhevsky et al. [17] which is very well known as the Alexnet architecture. The model consists of five convolutional layers and three fully connected layers. The first, second and fifth convolutional layers are followed by max-pooling layers for spatial subsampling. Each convolutional layer and fully-connected layer in the network is followed by a non-linearity layer, Rectified Linear Unit (ReLU) which introduces non-linearity to the network. Some of the ReLU layers are followed by a local response normalization (LRN) layer which helps in optimization and in obtaining generalization in the results. The output of the last fully connected layer is fed to a softmax layer which produces the normalized exponential probability of class observations represented as neuron activations.

The Alexnet architecture was trained on the ImageNet dataset with about 1.2 million images divided among 1000 classes. The CNN model split for training on two different GPUs is shown in Figure 3.5.

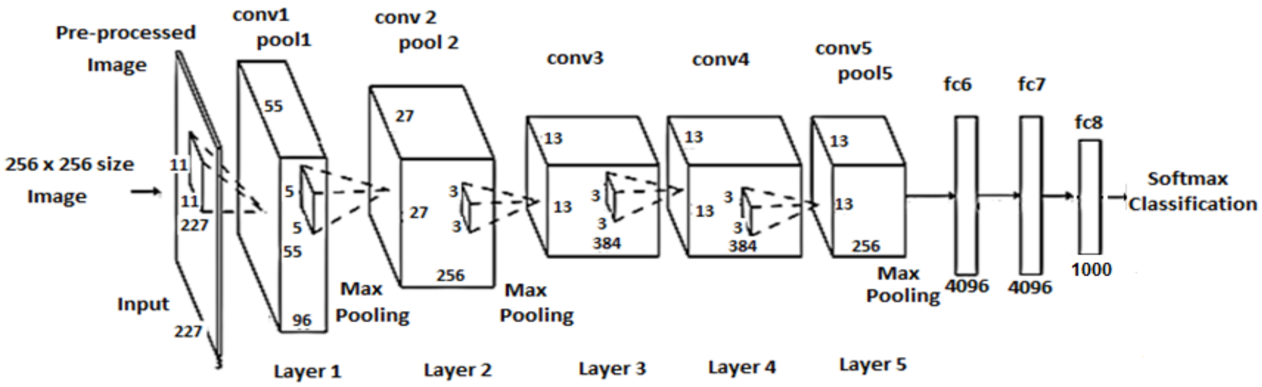


Figure 3.5 Alexnet architecture proposed by [17].

The parameter details of the Alexnet architecture are shown in Table 3.1. The convolution and pooling layer consists of a hyper parameter stride, which refers to the number of pixels the filter or the window should be slid.

Layer No.	Layers	Input	Parameters	Output
1	Conv-1	227 x 227 x 3	filters 96, filter size 11 x 11, stride 4	55 x 55 x 96
	ReLU-1	55 x 55 x 96	-	55 x 55 x 96
	Pool-1	55 x 55 x 96	pool type MAX, window size 3 x 3, stride 2	27 x 27 x 96
	LRN-1	27 x 27 x 96	local size 5, alpha 0.0001, beta 0.75	27 x 27 x 96
2	Conv-2	27 x 27 x 96	filters 256, filter size 5 x 5, pad 2	27 x 27 x 256
	ReLU-2	27 x 27 x 256	-	27 x 27 x 256
	Pool-2	27 x 27 x 256	pool type MAX, window size 3 x 3, stride 2	13 x 13 x 256
	LRN-2	13 x 13 x 256	local size 5, alpha 0.0001, beta 0.75	13 x 13 x 256
3	Conv-3	13 x 13 x 256	filters 384, filter size 3 x 3, pad 1	13 x 13 x 384
	ReLU-3	13 x 13 x 384	-	13 x 13 x 384
4	Conv-4	13 x 13 x 384	filters 384, filter size 3 x 3, pad 1	13 x 13 x 384
	ReLU-4	13 x 13 x 384	-	13 x 13 x 384
5	Conv-5	13 x 13 x 384	filters 256, filter size 3 x 3, pad 1	13 x 13 x 256
	ReLU-5	13 x 13 x 256	-	13 x 13 x 256
	Pool-5	13 x 13 x 256	pool type MAX, window size 3 x 3, stride 2	6 x 6 x 256
6	FC-6	9216 vector (6 x 6 x 256)	filters 4096	4096
	ReLU-6	4096	-	4096
	Dropout-6	4096	dropout ratio 0.5	4096
7	FC-7	4096	filters 4096	4096
	ReLU-7	4096	-	4096
	Dropout-7	4096	dropout ratio 0.5	4096
8	FC-8	4096	filters 1000 (total categories of ImageNet dataset), Softmax	1000

Table 3.1 Parameter details of Alexnet architecture

The number of pixels to be added on each side of input is specified by padding parameter. The added pixels have the values zero and hence it is zero-padding. The local size parameter of the LRN indicates the number of channels to sum over (for cross-channel LRN) or the side length of the square region to sum over (for within channel LRN). Alpha is the scaling parameter and beta is the exponent in the formula for the LRN.

To reduce overfitting in the Alexnet architecture, authors adopted two techniques: Data augmentation and Dropout. As a part of data augmentation, ten patches were extracted from the input image by cropping in the form of four corner patches and one center patch along with their horizontal reflections. The first two fully connected layers were followed by dropout layer in which some neurons are ‘dropped out’ reducing the complex co-adaptations of neurons.

Our method uses the Alexnet architecture pre-trained on the ImageNet dataset as the baseline model for comparison of results with the proposed technique. The Alexnet model was the first model giving outstanding performance on the enormous and difficult ImageNet dataset. The record breaking results of the architecture secured the first position in the ILSVRC-2012 competition, and since then the Alexnet model has been adopted by many researchers as baseline.

3.1.2 Importance of Pooling

Pooling summarizes the sub-region input feature map from convolutional layer to reduce the spatial size of the representation. The gathering of multiple features from neighborhood helps in achieving positional and translational invariance for image classification. The different types of pooling are max-pooling, average pooling, L2-norm pooling, of which max-pooling is used often because of its better performance than the other two. Given a window size for the input feature map of pooling layer, max-pooling selects the element with maximum value from that particular window. The max-pooling operation is shown in Figure 3.6. Max pooling provides robustness to position by reducing the dimensions of the intermediate representations. The reduction in dimensions ultimately leads to decrease in parameters of the network and hence lesser computation time.

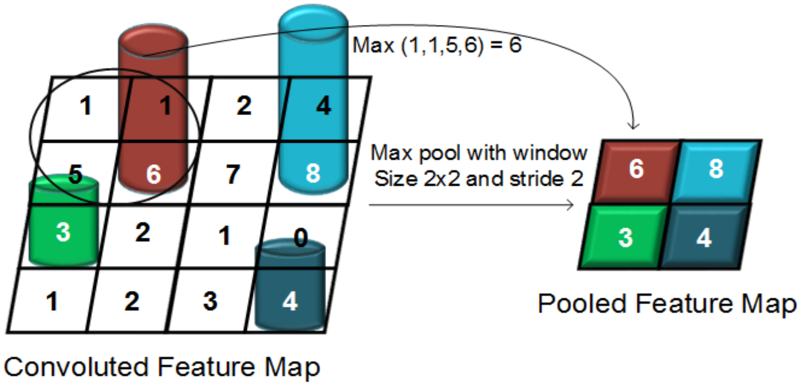
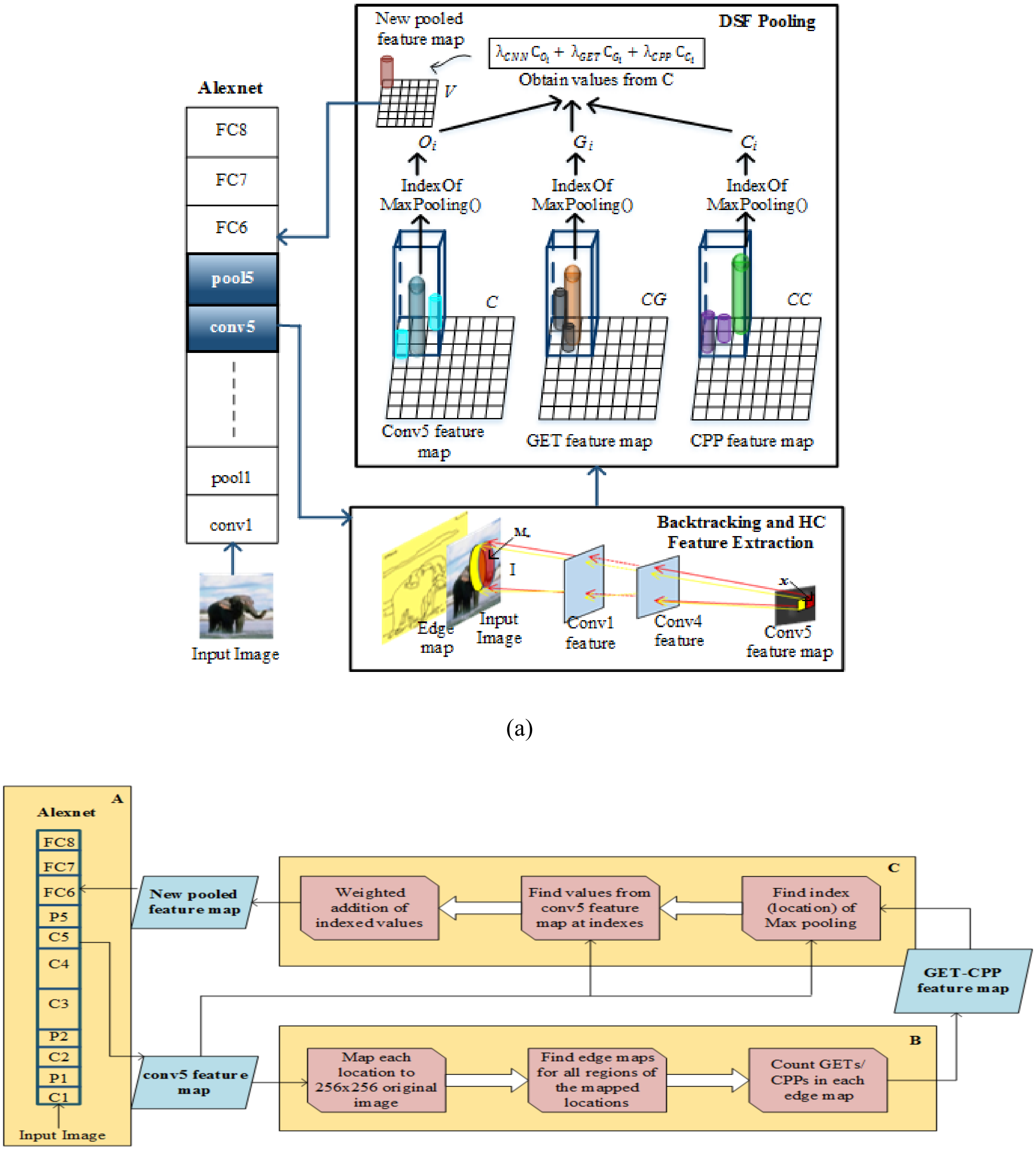


Figure 3.6 Max pooling operation.

3.2 System Architecture and Algorithm

An overview of the proposed system architecture is shown in Figure 3.7 (a) and the corresponding workflow is shown in Figure 3.7 (b). The combination of two different feature representations is done in 3 major parts - Deep learned feature extraction, Backtracking and Handcrafted feature extraction and finally Discriminative Shape Feature Pooling (DSFP).



(a)

(b)

Figure 3.7 (a) An overview of proposed system architecture. (b) Workflow of proposed method. A – DL feature extraction. B – Backtracking and HC feature extraction. C – DSFP.

The convolutional layers, pooling layers and fully connected layers are represented by C, P and FC respectively in Figure 3.7 (b) and the suffix shows the layer order number of the CNN. FC8 is the Classifier.

3.2.1 Deep Learned Feature Extraction

The feature maps from convolutional and pooling layer of the 5th layer (conv5 and pool5) are extracted from Alexnet model which is pre-trained on ImageNet dataset. The 5th layer of CNN represents high-level complex features of the image, which contain important object parts or the entire object itself. Hence, modifying these features with handcrafted features can produce more effective representation.

3.2.2 Backtracking and Handcrafted Feature Extraction

The steps for backtracking of conv5 feature map to the original image and corresponding GET/CPP feature extraction are shown in Algorithm 1.

```
Algorithm 1: Backtracking and GET-CPP Feature Extraction
Input: Input image $I$, conv5 feature map $C$
1. For each location $x \in C$,
  1.1   $M_x \leftarrow x \Theta I$, $M_x$ is the mapped region
  1.2   $C_{GET}(x) \leftarrow$ #GET($M_x$)
        $C_{CPP}(x) \leftarrow$ #CPP($M_x$)
Output: GET-CPP feature maps $CG$ and $CC$
```

The original input image I, undergoes convolutional and pooling operations in the CNN to produce reduced size feature maps based on the hyper-parameters of the network. Each location x

of the conv5 feature map C is mapped to its corresponding region M_x in I by backtracking. ʘ is the mapping operation which is based on hyper-parameters of the network- window size and stride for pooling layer and filter size, padding, number of filters for the convolutional layer.

Handcrafted features (in our case, edge map of GET-CPP) are extracted for each mapped region. The mapped regions obtained through backtracking and GET-CPP extraction is shown in Figure 3.8. We use the edge tracker proposed by [82] to extract the edge maps. The number of GET and CPP pixels are counted for each region to form GET and CPP feature maps CG and CC respectively. CG and CC are of same size as that of C.

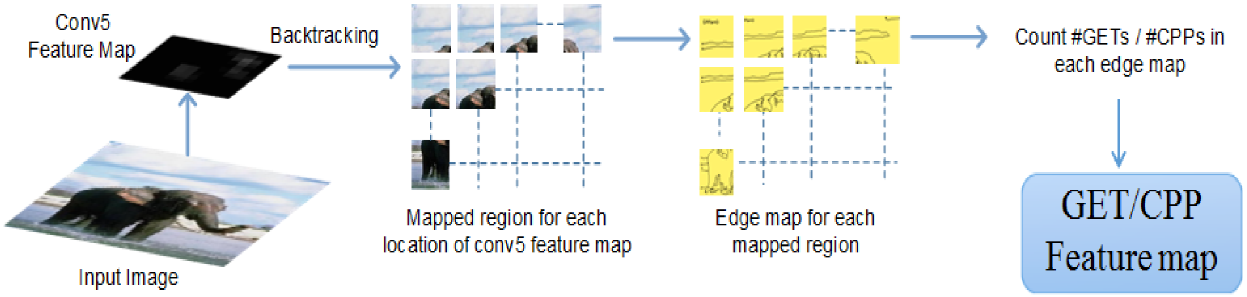


Figure 3.8 Backtracking and GET-CPP feature extraction.

3.2.3 Discriminative Shape Feature Pooling (DSFP)

The proposed DSFP method to obtain the new feature map based on the deep learned features and GET-CPP features is described in Algorithm 2.

For each pooling window of conv5 feature map C, and the handcrafted feature map CG and CC, the index (location) of max pooling is obtained.

Algorithm 2: Discriminative Shape Feature Pooling

```
Input: conv5 feature map C, GET-CPP feature maps CG and CC
1. For each pooling window $w$ in C, CG and CC,
  1.1  $O_i$ ← IndexOfMaxPooling(C, $w_{sz}$, $w_{st}$)
       $G_i$ ← IndexOfMaxPooling(CG, $w_{sz}$, $w_{st}$)
       $C_i$ ← IndexOfMaxPooling(CC, $w_{sz}$, $w_{st}$)
     where, $w_{sz}$ – pooling window size, $w_{st}$ – pooling stride
  1.2  $V_k$ ← $\lambda_{CNN}\, C_{O_i} + \lambda_{GET}\, C_{G_i} + \lambda_{CPP}\, C_{C_i}$
       where, k is the index of V, size(V) = [(size(C) - $w_{sz}$)/ $w_{st}$ + 1]; $\lambda_{CNN}, \lambda_{GET}, \lambda_{CPP} \in (0,1)$; $\lambda_{CNN} + \lambda_{GET} + \lambda_{CPP} = 1$.
Output: New pooled feature map V
```

The pooling parameters (window size and stride) are same for all the pooling windows. The function *IndexOfMaxPooling*(X, sz, st) gives the index of maximum element in window size sz with stride st from a feature map X, where X could be C, CG or CC. These indices tell us the locations of the visual salience in the image, which should be kept in the pooling process. Thus, values from conv5 feature map C of those indices are taken and weighted to form a new pooled feature map. The result obtained from indices of conv5 feature map C is the original pool5 feature map of the CNN. λ_{CNN}, λ_{GET} and λ_{CPP} are the weights for 3 locations derived from C, CG and CC respectively, the values of which lie between 0 and 1 and sum of the weights must equal to 1. The normalized new pooling results are still influenced by the original pooling results (by enforcing the final value in V to be equal or less than the maximum value in original pool5 feature map), meanwhile, balance the visual semantic cues from handcrafted features which were totally ignored in the conventional deep networks.

3.3 Softmax Classification

The feature representation of image is fed to a classifier for the task of image classification. In practice, many classification algorithms exist. These classifiers are used to train a model based on the input data and their corresponding labels. The trained model then predicts the class of an image when any unseen data is provided to the classifier.

A popular choice of classifier in terms of the deep CNN networks, is the Softmax classifier. The baseline Alexnet model uses the Softmax classifier, and hence we use the same classifier for classification. The fact that Softmax classifier gives the normalized probabilities for each class label is the reason for its popularity; and this property is also useful for top-k prediction, meaning that if the true label is present in the top k predicted labels or not. The Softmax classifier is a generalization of the binary form of logistic regression [83]. The mapping function *f* (Eq. 1) is defined such that given the input data set *x*, the function performs a linear dot product of the data *x* and the weight matrix **W** and maps the input to the output class labels**.**

$$f(x_i, \mathbf{W}) = \mathbf{W}x_i \qquad \text{(Eq. 1)}$$

The Softmax function takes a K-dimensional vector z of arbitrary real values and squashes it to a vector σ(z), which is a K-dimensional vector of real values ranging between 0 and 1 and sums up to 1 [84].

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \text{ for } j = 1, 2, \ldots, \mathrm{K} \qquad \text{(Eq. 2)}$$

The function attempts to minimize the cross entropy between the estimated class probabilities and the 'true' distribution. The classifier uses the cross-entropy loss function to interpret the un-normalized log probabilities for each class label. The cross-entropy loss function is of the form:

$$\boldsymbol{L_i} = -\log(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}) \qquad \text{(Eq. 3)}$$

In Eq. 3, $\boldsymbol{f_j}$ corresponds to the *j*-th element of vector of class scores $\boldsymbol{f}$. The complete loss for the dataset is the mean of $\boldsymbol{L_i}$ over all the training examples together with a regularization term. The purpose of selecting Softmax classifier over other classifiers is that it computes probabilities for all the class labels which allows to interpret its confidence in each class.

3.4 Implementation Details

Several deep learning software frameworks are available today, namely Caffe, Torch, Theano, Neon, TensorFlow, DeepLearning4J etc. [85]. The implementation is done using the open source framework Caffe [86] for deep learning on Ubuntu 14.04. Caffe is maintained by the Berkeley Vision and Learning Center (BLVC) and was initially designed for vision and later improved for other tasks such as speech recognition, neuroscience etc. The implementation of Caffe is done in C++, along with CUDA for computation using GPU; it also has an interface for MATLAB and Python. It is relatively easy to deploy and train CNN models using Caffe. The proposed shape feature pooling was implemented in MATLAB. The experiments were run on NVIDIA GeForce GTX 760 GPU. The GET-CPP features of the images are obtained from PCPG model [82].

We utilize the implementation of the Alexnet architecture available in Caffe, along with the Stochastic Gradient Descent (SGD) [87] optimization method to minimize the loss and update the parameters of the network through forward and backward pass. The idea of parameter settings has been adopted from Caffe fine-tuning tutorial [88]. We run our network for 9 epochs with an initial learning rate set to 0.001, which reduced by a factor of 10 after the completion of one-third of the maximum iterations chosen for learning. The momentum and weight decay was 0.9 and 0.0005 respectively and the maximum number of iterations and the test interval were decided based on the batch size used in the network and the number of images in the training set and the validation set. As a part of regularization, the fully connected layers of Alexnet are flowed by dropout layer to reduce overfitting.

The experimental results with comparison and evaluation on different datasets are discussed in Chapter 4.

Chapter 4

Experiments and Evaluation

The proposed method has been applied to various datasets containing a large variety of classes. The results on different datasets prove the success of research idea. This chapter discusses the details of the datasets used for evaluation and results obtained for image classification. The measurements used for classification and softmax classifier is described in the sub-sections. Finally, the experiment results have been compared with the results from baseline Alexnet and approaches by other researchers; and some analytical comments have been made based on the results obtained to justify the research contributions.

4.1 Datasets

The experiments based on proposed idea were carried out on 4 different datasets. We used the Caltech-256 dataset, which is a larger dataset of 257 categories. To narrow down the results of the investigation, we further tested on the Pascal VOC 2007 dataset which has only 20 categories. While the classes of Caltech-256 dataset and Pascal VOC 2007 dataset belong to diverse domains internally, we used the Oxford flowers dataset and KTH animals dataset to deduce the performance on specific domains. A summary of the datasets is shown in Table 4.1.

Dataset	#Classes	#Images	Training set		#Test images	Image Categories
			#Train	#Val		
Caltech-256	257	30,607	45/class	15/class	15,187	Man-made objects, natural scenery, living-beings
Pascal VOC 2007	20	9,963	2,501	2,510	4,952	Birds. Animals. Persons, vehicles, indoor objects
Oxford-102	102	8,189	10/class	10/class	6,149	Flowers
KTH-Animals	19	1,742	60%/class	20%/class	20%/class	Animals

Table 4.1 Summary of the datasets

4.1.1 Caltech-256 dataset

The Caltech-256 dataset [89] is a challenging dataset consisting of a wide variety of 256 categories of images along with an additional clutter category. The dataset is an extension of Caltech-101 dataset [90] and has considerably more categories and images than Caltech-101. The overall images in Caltech-256 is 30,607 with each class ranging from a minimum of 80 images to a maximum of about 827 images. The size of images varies largely and though the categories are independent to each other, some of them are closely related; which makes the classification of images difficult.

The image categories vary from manmade objects to natural scenery and living beings as well. The categories include animals such as bear, greyhound, dolphin, elephant, frog, goat etc., birds and insects such as owl, penguins, grasshoppers, butterfly etc., trees, plants and flowers such as cactus, bonsai, hibiscus, palm-tree and also, faces of people and other objects like soda-can, coffee-mug, computer-monitor, airplane etc. Some of the sample images from different classes of the Caltech-256 dataset are shown in Figure 4.1. For our experiments, we selected 60 images per class for training, out of which 15 were considered for validation; and the remaining images for testing.

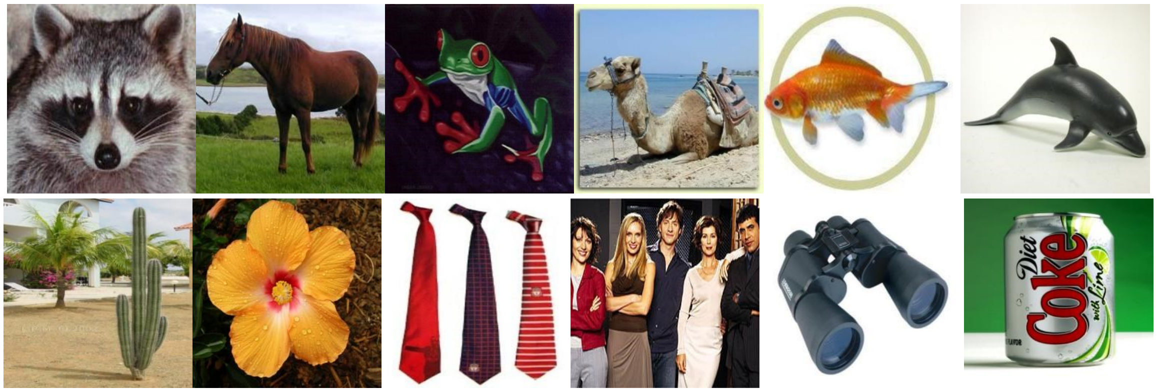


Figure 4.1 Sample images from Caltech-256 dataset.

4.1.2 Pascal VOC 2007 dataset

The Pascal Visual Object Classes (VOC) Challenge [91] began in the year 2005 and was being held every year until 2012; every year a modified and extended version of the Pascal VOC dataset was made available for experiments. The tasks of challenge include image classification, detection, segmentation, action classification and person layout. Due to the significant variation in images in terms of size, orientation, illumination, position and occlusion, Pascal VOC dataset has attracted researchers to evaluate their recognition methods. The very first dataset provided in 2005 consisted of only 4 classes, which increased to 10 classes in the year 2006; and later 20 classes in 2007. Since 2007, the number of classes remained 20, the only changes in the datasets were addition or deletion of images from the previous year's dataset and information added or modified for the segmentation, action classification and person layout challenges.

For our experiments, we used the Pascal VOC 2007 dataset [58] consisting of 20 classes having a total of 9,963 images split into 5,011 training images and 4,952 testing images. 2,510 images of the training set were used as validation set. Overall, the images contain approximately 24,500 objects. The classes include bird, animals like cat, dog, cow, horse, sheep; vehicles like aeroplane, bicycle, boat, bus, motorbike, car, train; indoor objects like bottle, chair, dining table, potted plant, sofa, tv/monitor and a class containing persons. Figure 4.2 shows the sample images from the Pascal VOC 2007 dataset.

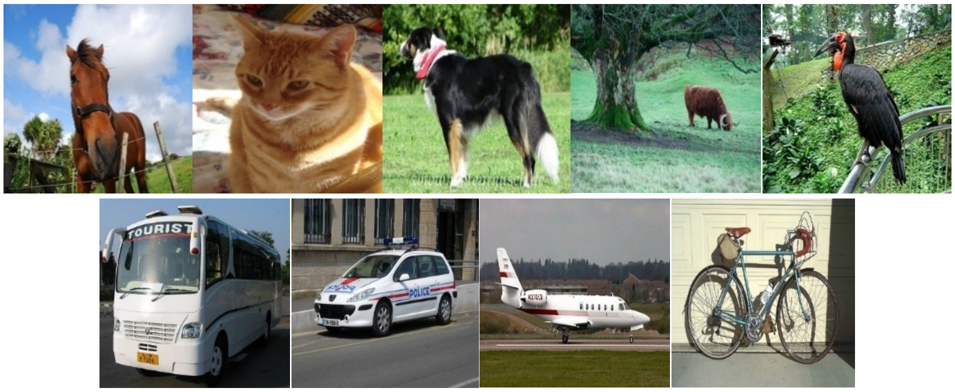


Figure 4.2 Sample images from Pascal VOC 2007 dataset.

4.1.3 Oxford-102 Flowers dataset

The Oxford-102 flowers dataset [92] consists of 102 flower categories, with each category having between 40 to 258 images. The dataset has a total of 8,189 images, of which 20 images per class are chosen for training and remaining as testing. 10 images per class are used for validation from the training set. The huge similarity between the classes and relatively smaller similarity in the images of same class makes the prediction of 102 categories of flowers difficult. The images vary in terms of scale, pose and illumination. To test the success of our method whether it can overcome such variations, we chose the flowers dataset for our experiments. Some sample images of the dataset are shown in Figure 4.3. The different categories of flowers include artichoke, hibiscus, daffodil, marigold, sunflower etc.

Figure 4.3 Sample images from Oxford-102 dataset.

4.1.4 KTH Animals Dataset

The KTH animals dataset [93] consists of outdoor images of 19 categories of different animals. Each class of the dataset consists an average of about 80-85 images. The 19 image classes are bear, goat, tiger, cow, giraffe, kangaroo, gorilla, panda, penguin, coyote, sheep, skunk, zebra, deer, elephant, lion, leopard, cougar and horse. We selected 80% of the images from each class for training and the remaining 20% for testing. 20% of images per class from training set were selected as validation set. Figure 4.4 shows some sample images from the KTH animals dataset.

Figure 4.4 Sample images from KTH animals dataset.

4.2 Data Augmentation and Pre-processing

As a part of data augmentation, ten patches of size 227 x 227 were extracted from the 256 x 256 images by cropping in the form of four corner patches and one center patch along with their horizontal reflections. In this way, the training data size was increased by a factor of 10. The result of data augmentation on the 256 x 256 input image in form of 10 different 227 x 227 patches is shown in Figure 4.5. The CNN model for all the datasets has been pre-trained on the ImageNet dataset and all the extracted patches were subtracted from the ImageNet mean as a part of pre-processing.

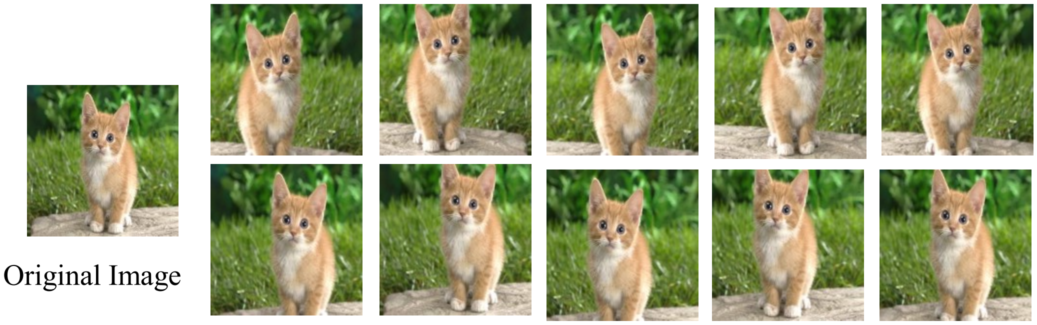


Figure 4.5 Data augmentation.

4.3 Evaluation metrics for Classification performance

Parameter tuning in the network plays an important role in building appropriate trained model which can be helpful for better prediction. A validation strategy known as cross-validation [94] is used to validate the performance of our method. The goal of cross-validation is to divide the dataset into 3 different sets: training set, validation set and test set; and train the model in the training phase based on the training set and validation set. The use of validation set in the training phase is to avoid overfitting in the network; because if the network is tested on the same data (i.e. training data), the model will fail for the unseen test data. Hence, the use of validation set is helpful for better tuning of parameters which ultimately builds good model for prediction. Cross-validation is considered to improve the predictive power of the trained model [95].

Some of the datasets that we have chosen for our experiments already provide the partition into training set, validation set and test set, while for the others we manually partition into the three sets. For instance, the Pascal VOC 2007 dataset and Oxford-102 Flowers dataset comes along with splits.

The performance of experiments has been evaluated based on the overall classification accuracy or mean average precision which are the evaluation metrics for image classification. The performance of classification can be visualized with a specific table layout, known as the confusion matrix or error matrix [96]. The scenario of the confusion matrix is shown in Table 4.2.

		Predicted	
		Positive	**Negative**
Actual	**Positive**	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

Table 4.2 Confusion matrix for evaluation of classification

The terms of the confusion matrix are calculated with respect to each class of the dataset. To explain the terms of the matrix, let us consider a classification problem with a dataset of two classes: cats and dogs. The confusion matrix for classification of the cat class denotes true positive to be the actual number of cats that were correctly classified as cats, and the false positive indicates the number of dogs that were incorrectly classified as cats. False Negative is the number of cats that were misclassified as dogs and true negative is the number of dogs that were correctly classified as non-cats.

The overall classification accuracy of the dataset can be classified based on the confusion matrix, which is the proportion of the total number of predictions that were correct. We measure the classification accuracy for Caltech-256, Oxford flowers and KTH animals dataset.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \qquad \text{(Eq. 4)}$$

The performance of Pascal VOC 2007 dataset is evaluated by the mean average precision (mAP), an evaluation metric suggested by the Pascal VOC Challenge [58]. Precision is the proportion of predicted positive cases that were actually correct, whereas the ratio of true prediction of the class to total number of instances in that class is called the Recall.

$$Precision = \frac{TP}{TP+FP} \quad \text{(Eq. 5)}$$

$$Recall = \frac{TP}{TP+FN} \quad \text{(Eq. 6)}$$

Since the images in Pascal VOC 2007 dataset contain instances from multiple classes, results are evaluated based on binary classification tasks, for each of the 20 classes of the dataset. For each of the tasks and class, a precision/recall curve is plotted and average precision (AP) is calculated, which is the mean precision at a set of eleven equally spaced recall levels [0, 0.1, …, 1].

$$Average\ Precision\ (AP) = \frac{1}{11}\sum_{r\ \epsilon\ \{0,0.1,\dots,1\}} p_{interp}(r) \quad \text{(Eq. 7)}$$

The precision at each recall level *r* is interpolated by taking the maximum precision for which the corresponding recall exceeds *r*.

$$p_{interp}(r) = \max_{r':r' \geq r} p(r') \quad \text{(Eq. 8)}$$

where, *p(r')* is the measured precision at recall *r'*. The mean average precision (mAP) for the entire dataset is then calculated by taking the mean of the average precisions over all the 20 classes.

$$mean\ Average\ Precision\ (mAP) = \frac{1}{n}\sum_{i=1}^{n} AP_i \quad \text{(Eq. 8)}$$

where, *n* is the number of classes, i.e. 20 and AP_i denotes the average precision of class *i*.

4.4 Experiment Results and Comparison

Apart from the GET-CPP features, we also conducted experiments on SURF feature for our method. The performance of baseline Alexnet model, SURF-based pooling and DSFP for all the 4 datasets is shown in Table 4. The results show that in all the cases, DSFP outperforms the baseline model, and DSFP performs better than SURF-based pooling also. The DSFP results reported in Table 4.3 are for the weight values $\lambda_{CNN} = 0.5$, $\lambda_{GET} = 0.25$ and $\lambda_{CPP} = 0.25$ and $\lambda_{CNN} = 0.75$ and $\lambda_{SURF} = 0.25$ for SURF-based pooling.

	Caltech-256 (%)	**Pascal VOC 2007 (mAP)**	**Oxford-102 (%)**	**KTH Animals (%)**
Baseline	74.47	80.21	80.45	93.01
DSFP	76.13	81.45	83.61	95.69
SURF-based pooling	71.31	80.03	81.15	95.12

Table 4.3 Experiment results baseline Alexnet, SURF-based pooling and DSFP

We investigated the classification performance of our method based on the assignment of different combination of weights for perceptual shape feature pooling. As discussed in Section 3.2.3, the weights are λ_{CNN} (weight for pool5 feature map from Alexnet), λ_{GET} (weight for GET feature map) and λ_{CPP} (weight for CPP feature map), which are normalized from 0 to 1. The results on different datasets for some specific combination of weights is shown in Figure 4.6.

The different combinations in Figure 21 include entire GET and CPP feature map ($\lambda_{GET} = 1$, $\lambda_{CPP} = 1$), baseline Alexnet model ($\lambda_{CNN} = 1$), combination of GET feature map and original pool5 feature map ($\lambda_{GET} = \lambda_{CNN} = 0.5$), combination of CPP and original pool5 feature map ($\lambda_{CPP} = \lambda_{CNN} = 0.5$) and combination of all three ($\lambda_{CPP} = \lambda_{GET} = 0.25$, $\lambda_{CNN} = 0.5$).

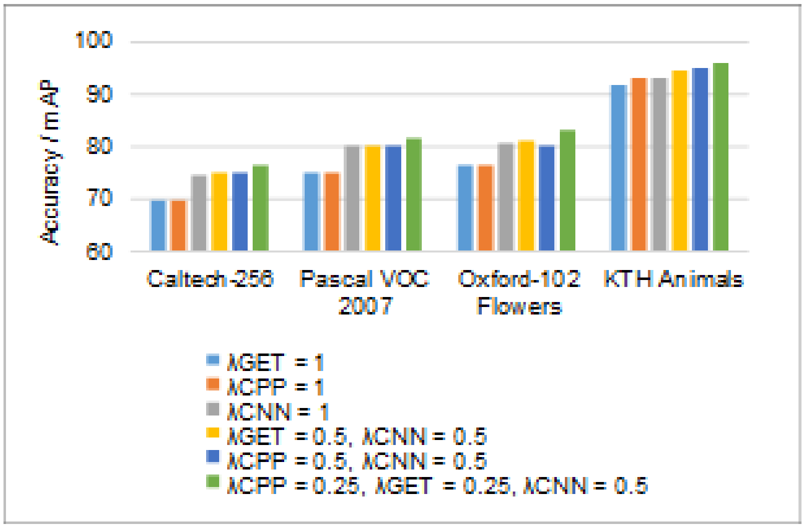


Figure 4.6 Result comparison of proposed method for different combination of weights of the pool5, GET and CPP feature map.

It can be inferred that the performance of pooling with reference to only the perceptual features is worse than the baseline CNN model. However, the combination of the CNN model along with the perceptual features gives better results compared to the perceptual features or CNN model alone. Moreover, better results have been obtained when $\lambda_{CNN} >= (\lambda_{GET} + \lambda_{CPP})$; this shows that the original pooled features from the baseline CNN still play importance roles, although the injection of GET and CPP features also helps in good performance.

Methods	Accuracy (%)
Nilsback and Zisserman – Color, HOG, SIFT [92]	72.8
Chai et al. – Superpixel segmentation [97]	80.0
Baseline Alexnet Model	80.45
DSFP in CNN (ours)	**83.61**

Table 4.4 Comparison of classification accuracy on Oxford-102

Apart from this, we also compared the results of our proposed method with other approaches that involve the use of handcrafted features or deep learning models alone. The comparison of results on Oxford-102 Flowers, Caltech-256, Pascal VOC 2007 and dataset in Table 4.4, Table 4.5 and Table 4.6 respectively, state that the methods using handcrafted features alone perform worse than the deep learning techniques. However, our method shows better performance in comparison to handcrafted features and several other CNN models.

Methods	mAP
Huang et al. – SIFT, Improved Fisher Kernel [36]	58.05
Sande et al. – SIFT, C-SIFT, OpponentSIFT, RGB-SIFT, rg-SIFT [98]	60.05
Razavian et al. – Overfeat [15]	77.2
Oquab et al. – Transfer of mid-level CNN representation [99]	77.7
He et al. – SPP-net [19]	80.1
Chatfield et al. – CNN-S [100]	82.4
Baseline Alexnet Model	80.21
DSFP in CNN (ours)	**81.45**

Table 4.5 Comparison of mAP on Pascal VOC 2007

Methods	Accuracy (%)
Sohn et al. – Convolutional RBMs, SIFT [101]	47.94
Huang et al. – SIFT, Improved Fisher Kernel [36]	52.0
Bo et al. – Multipath HMP [102]	55.2
Zeiler and Fergus – ZF-net [66]	74.2
Chatfield et al. – CNN-S [100]	77.6
Gao et al. – DEFEATnet [75]	48.52
Baseline Alexnet model	74.47
DSFP in CNN (ours)	**76.13**

Table 4.6 Comparison of classification accuracy on Caltech-256

The CNN-S network by K. Chatfield et al. [100] gives a little higher or similar results compared to DSFP, and is composed of 8 layers just like the Alexnet, but uses more filters for several layers compared to Alexnet; hence the model needs to tune more parameters leading to increase in computation time. The comparison of parameters of Alexnet and CNN-S is shown in Table 4.7 which indicates that along with increased parameters, CNN-S requires more memory to process one image compared to Alexnet. In this sense, our method produces similar results compared to the CNN network that have more parameters than Alexnet.

Layers	1	2	3	4	5	6	7	Total Parameters	Memory
Alexnet	96	256	384	384	256	4096	4096	60 million	~ 22 MB / image
CNN-S [99]	96	256	512	512	512	4096	4096	79 million	~ 27 MB / image

Table 4.7 Parameter comparison of Alexnet and CNN-S [99]

A detailed study of class-wise results on the Caltech-256 and Pascal VOC datasets revealed that the method provides better performance for categories of living beings, that include animals, birds, insects, persons and natural scenes like flowers, plants; compared to the results of the baseline Alexnet model. Also, our method is statistically significant with respect to these categories of the datasets. Figure 4.7 shows the results on living beings and natural scenes categories of the two datasets.

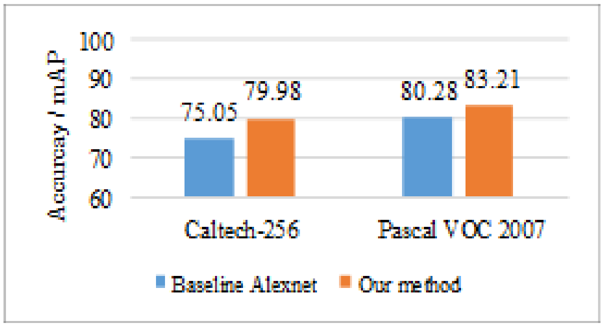


Figure 4.7 Result comparison on Caltech-256 and Pascal VOC 2007 dataset on categories of living-beings and natural scenes.

	mAP	**aero**	**bike**	**bird**	**boat**	**bottle**	**bus**	**car**	**cat**	**chair**	**cow**	**table**	**dog**	**horse**	**mbike**	**person**	**plant**	**sheep**	**sofa**	**train**	**tv**
Alexnet	**80.21**	90.69	88.64	76.34	76.46	71.65	75.94	86.68	85.60	74.10	62.45	73.64	81.12	82.05	82.40	90.32	85.26	79.10	66.30	88.85	86.61
DSFP	**81.45**	88.81	**91.07**	**83.67**	72.34	**75.52**	74.58	**86.90**	**88.52**	71.85	**68.16**	**76.32**	**86.47**	**85.34**	81.11	**91.11**	**90.33**	78.81	63.67	87.96	86.46

Table 4.8 Pascal VOC 2007 classification results

	butterfly	**cormorant**	**elephant**	**gorilla**	**ostrich**	**owl**	**penguin**	**hibiscus**	**hawksbill**	**Christ**	**swan**
Alexnet	67.65	78.57	77.36	76.87	87.10	69.05	70.42	90.11	90.91	66.47	78.38
DSFP	**82.35**	**92.86**	**83.02**	**77.61**	87.10	64.29	**74.65**	90.11	**93.45**	**68.14**	**81.08**

Table 4.9 Classification accuracy % of Caltech-256 (selected classes)

The classification results on test set of 20 classes in Pascal VOC 2007 dataset are shown in Table 4.8; which reveals that the overall performance of all classes is consistent and better. Table 4.9 shows classification accuracy of some selected classes from several categories of living beings and natural scenes of Caltech-256 dataset.

By examining the learning curve for the baseline Alexnet model and our method, we observed that our method converged quicker compared to the baseline Alexnet model. The guidance of handcrafted features helped the CNN model to quickly reduce the loss, which indicates the efficiency of our method. The learning curve of all the 4 datasets for Alexnet and our method is shown in Figure 4.8.

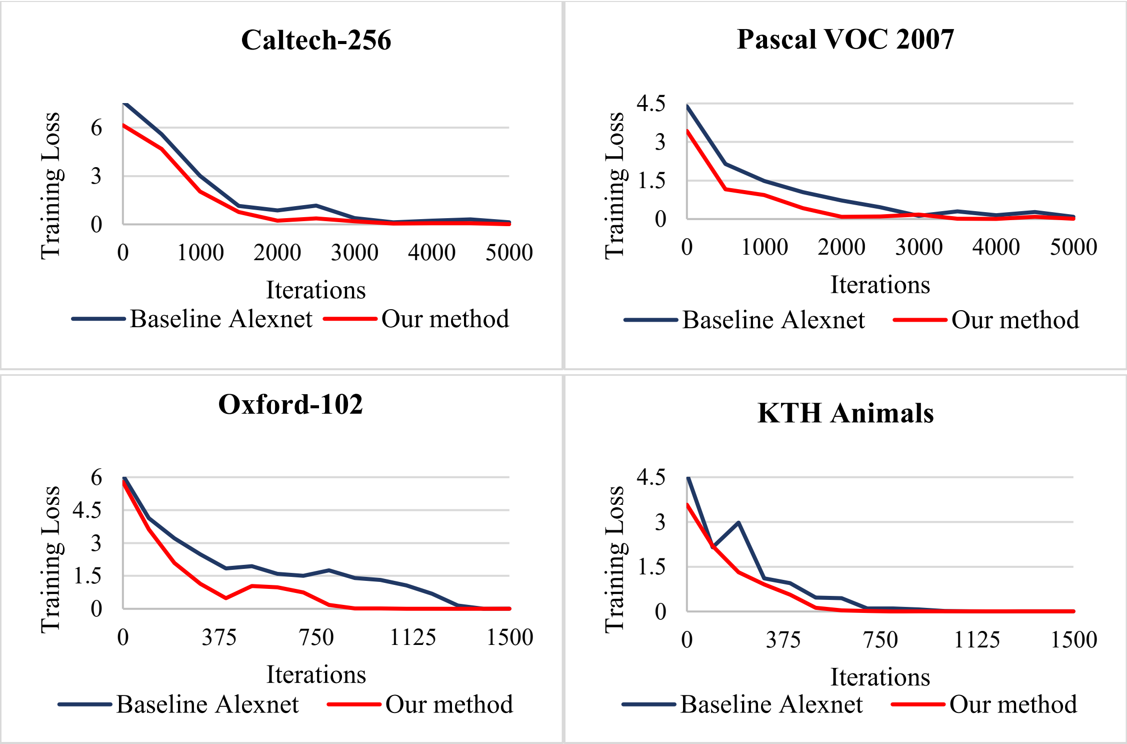


Figure 4.8 Learning curve (Training Loss versus Iterations) for baseline Alexnet and proposed method.

Additionally, we examined various classes of datasets to explore the reason behind obtained results. For detailed analysis, we compared the edge maps of images. A few sample images and their corresponding edge maps from some classes of the datasets are shown in Figure 4.9.

The baseline CNN model misclassified binoculars as microscope and goose as ibis, whereas our method worked well for those classes. The difference in edge maps of the images can be seen in Figure 4.9 (a), (b), (c) and (d), because of which the structure of images was identified correctly for our method resulting in better performance. However, our method did not perform very well for some very similar type of classes like coffee mug and beer mug, and mussels and snail. The reason behind failure can be deduced from the edge maps in Figure 4.9 (e), (f), (g) and (h) which shows nearly similar structures for the network to get confused easily.

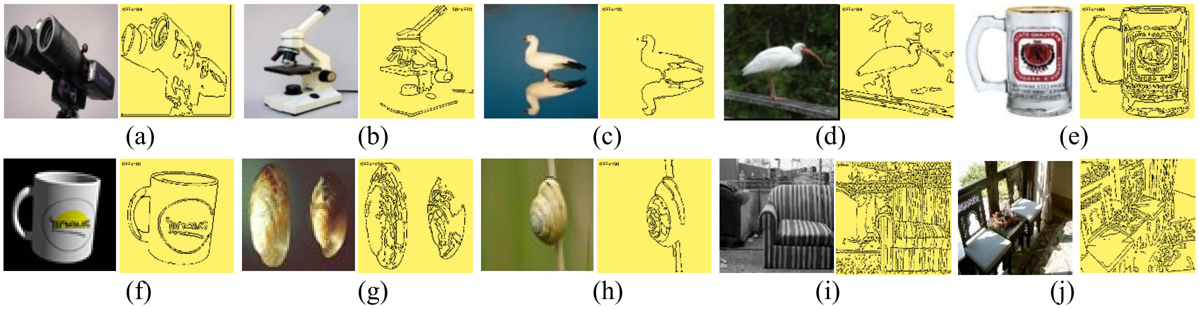


Figure 4.9 Sample images and corresponding edge maps of (a) Binoculars (b) Microscope (c) Goose (d) Ibis (e) Beer mug (f) Coffee mug (g) Mussels (h) Snail (i) Sofa (j) Chair.

Besides such similarity in the classes, background of images played a vital role in performance. Images with noisy backgrounds showed poor performance for our method. Based on our observation, the edge map of such noisy images was unclear and the objects in the image could not be identified correctly. Hence, the injection of these features into CNN could not produce an effective representation which led to wrong prediction results. Images with noisy background and their edge maps that did not perform well for our method can be seen in Figure 4.9 (i) and (j).

A possible solution to improve the performance of similar looking classes could be to use a combination of various other handcrafted features like color features, texture features or local image features like SIFT along with the perceptual shape features. On the other hand, to eliminate the problem of noisy images, some pre-processing can be done to remove the noise from images, or the objects in the image can be separated from the background via segmentation to achieve improved prediction results on such images.

Figure 4.10 shows the visualization of 2 types of learned pooled layer 5 feature maps. The maps on the second row are from the baseline model, and the third row shows maps from proposed DSFP model. It can be seen that our pooling scheme can discover and catch more details of each target images. For example, for the binocular image, our feature map can tell the difference for the left and right parts; for the flower, our output shows the different patterns for the petals stamen. The visualization partially reveals the reasons of performance difference.

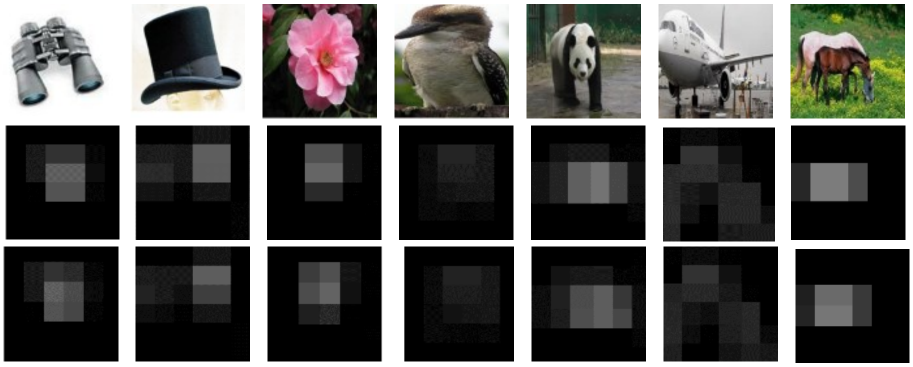


Figure 4.10 Visualization of learned feature maps with and without DSFP approach.

Chapter 5

Conclusion and Future Work

The idea of modifying the pooling layer of learned deep network with the help of perceptual shape feature representation works in this research, the results on a variety of datasets has consistently proved that. Today, when researchers are trying to go deeper into the deep networks, we have adopted a novel way to combine the handcrafted features with deep network of 8 layers which has also reduced the computation cost. Apart from this, the method shows faster convergence rate compared to baseline. The modified visual representation of feature map contains more detailed information of objects, so that the performance of proposed method is better. The consistent performance of the method on different datasets contribute as a very good potential for practical solutions of many application domains.

Although the proposed method provides better classification results, there are still some images wrongly classified. To improve the performance, we plan to inject the other handcrafted features like SIFT, SURF, color or texture features along with the perceptual shape features into deep network. We believe that the combination of local and global features can prove to be beneficial by considering different aspects of image and provide better visual representation. Apart from this, we also intend to find a way of learning the weight parameters to make them dynamic, instead of using static values through hard-coding. We further plan to consider parallel architectures to speed up the network training.

Bibliography

[1] Z. Lan, S.-I. Yu, B. Raj and A. G. Hauptmann, "Local Handcrafted Features are Convoltuional Neural Networks," *arXiv preprint arXiv:1511.05045,* 2015.

[2] D. Lisin, M. Mattar, M. Blaschko, E. Learned-Miller and M. Benfield, "Combining Local and Global Image Features for Object Class Recognition," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops,* vol. 3, p. 47, 2005.

[3] D. Lowe, "Object Recognition from Local Scale-Invariant Features," *Proceedings of 7th IEEE International Conference on Computer Vision (ICCV),* vol. 2, pp. 1150-1157, 1999.

[4] H. Bay, T. Tuytelaars and L. V. Gool, "SURF: Speeded Up Robust Features," *European Conference on Computer Vision (ECCV),* pp. 404-417, 7 May 2006.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* vol. 1, pp. 886-893, 25 June 2005.

[6] G. Csurka, C. Dance, L. Fan, J. Willamowski and C. Bray, "Visual categorization with bags of keypoints," *Workshop on Statistical Learning in Computer Vision, ECCV,* vol. 1, no. 1, pp. 1-22, 15 May 2004.

[7] D. P. Tian, "A Review on Image Feature Extraction and Representation Techniques," *International Journal of Multimedia and Ubiquitous Engineering,* vol. 8, no. 4, pp. 385-396, July 2013.

[8] S. A. Medjahed, "A Comparative Study of Feature extarction Methods in Image Classification," *International Journal of Image, Graphics and Signal Processing,* vol. 7, no. 3, pp. 16-23, 1 February 2015.

[9] L. Liu and P. Fieguth, "Texture Classification from Random Features," *IEEE Transcations on Pattern Analysis and Machine Intelligence,* vol. 34, no. 3, pp. 574-586, March 2012.

[10] R. Duda and P. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Communications of the ACM,* vol. 15, no. 1, pp. 11-15, 1972.

[11] A. Aggarwal and K. Singh, "Zernike moments-based retrieval of CT and MR images," in *Anual IEEE Indian Conference*, 2015.

[12] L. Jin, S. Gao, Z. Li and J. Tang, "Hand-Crafted Features or Machine Learnt Features? Together They Improve RGB-D Object Recognition," in *2014 IEEE International Symposium on Multimedia*, 2014.

[13] M. Dandan, "A Survey on Deep Learning: One Small Step toward AI," in *Department of Computer Science, University of New Mexico*, USA, 2012.

[14] K. Jarrett, K. Kavukcuoglu, M. Ranzato and Y. LeCun, "What is the best multi-stage architecture for object recognition?," in *2009 IEEE 12th International Conference on Computer Vision*, 2009.

[15] A. S. Razavian, H. Azizpour, J. Sullivan and S. Carlsson, "CNN Features off-the-shelf: An Astounding Baseline for Recognition," in *Proceedings of the IEEE Conference on CVPR Workshops*, 2014.

[16] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE,* vol. 86, no. 11, pp. 2278-2324, 1998.

[17] A. Krizhevsky, I. Sutskever and G. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems,* pp. 1097-1105, 2012.

[18] Z. Zheng, Z. Li, A. Nagar and K. Park, "Compact deep neural networks for device based image classification," in *2015 IEEE International Conference on Multimedia and Expo Workshops*, 2015.

[19] K. He, X. Zhang, S. Ren and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," in *13th European Conference on Computer Vision*, 2014.

[20] Q.-G. Gao and A. Wong, "Curve Detection based on Perceptual Organization," *Pattern Recognition,* vol. 26, no. 7, pp. 1039-1046, July 1993.

[21] R. Lan and Y. Zhou, "Quaternion-Michelson Descriptor for Color Image Classification," *IEEE Transactions on Image Processing,* vol. 25, no. 11, pp. 5281-5292, November 2016.

[22] T. Tuytelaars and K. Mikolajczyk, "Local Invariant Feature Detectors: A Survey," *Foundations and Trends in Computer Graphics and Vision,* vol. 3, no. 3, pp. 177-280, January 2008.

[23] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision,* vol. 60, no. 2, pp. 91-110, November 2004.

[24] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 27, no. 10, pp. 1615-1630, 2005.

[25] C. Harris and M. Stephens, "A Combined Corner and Edge Detector," *Alvey Vision Conference,* vol. 15, pp. 147-151, 1988.

[26] H. I. Kim, S. Shin, W. Wang and S. I. Jeon, "SVM-based Harris corner detection for breast mammogram image normal/abnormal classification," in *Proceedings of the 2013 Research in Adaptive and Convergent Systems*, 2013.

[27] A. P. Witkin, "Scale-space Filtering," *Proceedings of 8th International Joint Conference on Artificial Intelligence,* pp. 1019-1022, 1983.

[28] Y. Ke and R. Sukthankar, "PCA-SIFT: A More Distinctive Representation for Local Image Descriptors," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004.

[29] V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, R. Grzeszczuk and B. Girod, "CHoG: Compressed Histogram of Gradiemts A Low Bit-rate Descriptor," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[30] E. Rosten and T. Drummond, "Machine Learning for High-Speed Corner Detection," *9th European Conference on Computer Vision,* pp. 430-443, 7 May 2006.

[31] M. Calonder, V. Lepetit, C. Strecha and P. Fua, "BRIEF: Biinary Robust Independent Elementary Features," in *European Cinference on Computer Vision*, 2010.

[32] S. Leutenegger, M. Chli and R. Y. Siegwart, "BRISK: Binary Robust invariant scalable keypoints," in *IEEE International Conference on Computer Vision*, 2011.

[33] E. Rublee, V. Rabaud, K. Konolige and G. Bradski, " ORB: An efficient alternative to SIFT or SURF," in *IEEE International Conference on Computer Vision*, 2011.

[34] A. Alahi, R. Ortiz and P. Vandergheynst, " FREAK: Fast Retina Keypoint," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[35] T. Joachim, "Text Categorization with Support Vector Machines: Learning with many Relevant Features," in *Proceedings of the 10th European Conference on Machine Learning*, 1998.

[36] Y. Huang, Z. Wu, L. Wang and T. Tan, "Feature Coding in Image Classification: A Comprehensive Study," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 36, no. 3, pp. 493-506, March 2013.

[37] L. Fei-Fei and P. Perona, "A Bayesian Hierarchical Model for Learning Natural Scene Categories," in *IEEE Computer Scoiety Conference on Computer Vision and Pattern Recognition*, 2005.

[38] J. A. Hartigan and A. M. Wong, "Algorithm AS 136: A K-Means Clustering Algorithm," *Journal of the Royal Statistical Society. Series C (Applied Statistics),* vol. 28, no. 1, pp. 100-108, 1979.

[39] K. Zhou, K. Yu, T. Zhang and T. S. Huang, "Image Classification using Super-Vector coding of Local Image Descriptors," in *Proceedings of 11th European Conference on Computer Vision*, 2010.

[40] Y.-L. Boureau, F. Bach, Y. LeCun and J. Ponce, "Learning Mid-Level Features for Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[41] M. T. Law, N. Thome and M. Cord, "Bag-of-Words Image Representation: Key Ideas and Further Insight," in *Fusion in Computer Vision*, Springer International Publishing, 2014, pp. 29-52.

[42] T. Deselaers, L. Pimenidis and H. Ney, "Bag-of-Visual-Words models for Adult Image Classification," in *19th International Conference on Pattern Recognition*, 2008.

[43] T. Li, T. Mei, I.-S. Kweon and X.-S. Hua, "Contextual Bag-of-Words for Visual Categorization," *IEEE Transactions on Circuits and systems for Video Technology,* vol. 21, no. 4, pp. 381-392, April 2011.

[44] S. Schwartz, Visual Perception: A Clinincal Orientation, McGraw-Hill Medical, 2009.

[45] S. Sergyan, "Color histogram features based image classification in content-based image retrieval systems," in *6th International Symposium on Applied Machine Intelligence and Informatics*, 2008.

[46] G.-H. Liu and J.-Y. Yang, "Content-based Image Retrieval using Color Difference Histogram," *Pattern Recognition,* vol. 46, no. 1, pp. 188-198, 2013.

[47] M. Yang and L. Zhang, "Gabor Feature Based Sparse Representation for Face Recognition with Gabor Occlusion Dictionary," in *11th European Conference on Computer Vision*, Greece, 2010.

[48] I. Fogel and D. Sagi, "Gabor Filters as Texture Discriminator," *Biological Cybernetics,* vol. 61, no. 2, pp. 103-113, 1989.

[49] D. Sudarvizhi, "Feature based image retrieval system using Zernike moments and Daubechies Wavelet Transform," in *2016 International Conference on Recent Trends in Information Technology*, 2016.

[50] V. Risojević and Z. Babić, "Fusion of Global and Local Descriptors for Remote Sensing Image Classification," *IEEE Geoscience and Remote Sensing Letters,* vol. 10, no. 4, pp. 836-840, 2012.

[51] A. E. Abdel-Hakim and A. A. Farag, "CSIFT: A SIFT Descriptor with Color Invariant Characteristics," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

[52] Y. Bengio, "Learning Deep Architectures for AI," *Foundations and Trends in Machine Learning,* vol. 2, no. 1, pp. 1-127, 2009.

[53] H. A. Song and S.-Y. Lee, "Hierarchical Representation Using NMF," in *Proceedings of 20th International Conference on Neural Information Processing*, Korea, 2013.

[54] Y. Bengio, A. Courville and P. Vincent, "Representation Learning: A Review and New Perspectives," *IEEE Transactions on Pattern analysis and Machine Intelligence,* vol. 35, no. 8, pp. 1798-1828, 2013.

[55] G. E. Hinton, S. Osindero and Y.-W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Computation,* vol. 18, no. 7, pp. 1527-1554, 2006.

[56] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg and L. Fei-Fei, "ImageNet Large Scale Visual

Recognition Challenge," *International Journal of Computer Vision,* vol. 115, no. 3, pp. 211-252, 2015.

[57] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *IEEE Conference on Computer Vision and PAttern Recognition*, 2009.

[58] M. Everingham, L. V. Gool, C. K. I. Williams, J. Winn and A. Zisserman, "The PASCAL Visual Object Classes (VOC) Challenge," *Interntational Journal of Computer Vision,* vol. 88, no. 2, pp. 303-338, 2010.

[59] Y. LeCun, F. J. Huang and L. Bottou, "Learning methods for generic object recognition with invariance to pose and lighting," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004.

[60] D. Ciresan, U. Meier, J. Masci, L. Gambardella and J. Schmidhuber, "Flexible, High Performance Convolutional Neural Networks for Image Classification," in *Proceedings of 22nd International Joint Conference on Artificial Intelligence*, 2011.

[61] R. Fergus, "Deep Learning Methods for Vision," *CVPR 2012 Tutorial,* 2012.

[62] M. Nielsen, Neural Networks and Deep Learning, Determination Press, 2015.

[63] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research,* vol. 15, no. 1, pp. 1929-1958, 2014.

[64] D. Erhan, P.-A. Manzagol, Y. Bengio, S. Bengio and P. Vincent, "The Difficulty of Training Deep Architectures and the Effect of Unsupervised Pre-Training," in *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, Clearwater (Florida), USA, 2009.

[65] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent and S. Bengio, "Why Does Unsupervised Pre-training Help Deep Learning?," *Journal of Machine Learning Research,* vol. 11, pp. 625-660, 2010.

[66] M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," in *Proceedings of 13th European Conference on Computer Vision*, Zurich, Switzerland, 2014.

[67] K. Simonyan and A. Zisserman, "Very Deep ConvolutionalNetworks for Large-Scale Image Recognition," *arXiv preprint arXiv:1409.1556,* 2014.

[68] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang and Y. Gong, "Locality-constrained Linear Coding for image classification," in *2010 IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[69] S. Lazebnik, C. Schmid and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," in *2006 IEEE Computer Society Conference on Computer Vision and Patetrn Recognition*, 2006.

[70] H. Wang, A. Cruz-Roa, A. Basavanhally, H. Gilmore, N. Shih, M. Feldman, J. Tomaszewski, F. Gonzalez and A. Madabhushi, "Mitosis detection in breast cancer pathology images by combining handcrafted and convolutional neural network features," *Journal of Medical Imaging,* vol. 1, no. 3, 2014.

[71] S. Wu, Y.-C. Chen, X. Li, A.-C. Wu, J.-J. You and W.-S. Zheng, "An Enhanced Deep Feature Representation for Person Re-identification," in *2016 IEEE Winter Conference on Applications of Computer Vision*, 2016.

[72] M. N. Kashif, S. E. A. Raza, K. Sirinukunwattana, M. Arif and Rajpoot Nasir, "Handcrafted Features with convoltuional neural networks for detection of tumor cells in histology images," in *2016 IEEE 13th International Symposium on Biomedical Imaging*, 2016.

[73] K. Sirinukunwattana, S. E. A. Raza, Y.-W. Tsang, D. Snead, I. Cree and N. Rajpoot, "A Spatially Constrained Deep Learning Framework for Detection of Epithelial Tumor Nuclei in Cancer Histology Images," in *Patch-Based Techniques in Medical Imaging*, Springer, 2015, pp. 154-162.

[74] S. Mallat, "Group invariant scattering," *Communications on Pure and Applied Mathematics,* vol. 65, no. 10, pp. 1331-1398, 2012.

[75] S. Gao, L. Duan and I. Tsang, "DEFEATnet – A Deep Conventional Image Representation for Image Classification," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 26, no. 3, pp. 494-505, March 2016.

[76] Y. Gong, L. Wang, R. Guo and S. Lazebnik, "Multi-scale Orderless Pooling of Deep Convolutional Activation Features," in *European Conference on Computer Vision*, 2014.

[77] K. Wolfgang, "Gestalt psychology: An introduction to new concepts in modern psychology," *WW Norton & Company,* 1970.

[78] G. Guy and G. Medioni, "Inferring Global Conceptual Contours from Local Features," in *1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1993.

[79] S. Kim and I. S. Kweon, "Biologically Motivated Perceptual Feature: Generalized Robust Invariant Feature," in *7th Asian Conference on Computer Vision*, Hyderabad, India, 2006.

[80] M. Jian, H. Guo and L. Liu, "Texture Image classification using Visual Perceptual Texture Features and Gabor Wavelet Features," in *Asia-Pacific Conference on Information Processing*, 2009.

[81] B. Long, T. Konkle, M. A. Cohen and G. A. Alvarez, "Mid-Level Perceptual Features Distinguish Objects of Different Real-World Sizes," *Journal of Wxperiment Psychology: General,* vol. 145, no. 1, pp. 95-109, 2016.

[82] G. Hu and Q. Gao, "A non-parametric statistics based method for generic curve partition and classification," in *2010 17th IEEE International Conference on Image Processing*, 2010.

[83] "CS231n Convolutional Neural Networks for Visual Recognition".*Stanford University.*

[84] "Softmax function, Wikipedia," [Online]. Available: https://en.wikipedia.org/wiki/Softmax_function.

[85] S. Bahrampour, N. Ramakrishnan, L. Schott and M. Shah, "Comparative Study of Deep Learning Software Frameworks," *arXiv preprint arXiv:1511.06435,* 2015.

[86] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama and T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding," in *Proceedings of the 22nd ACM International Conference on Multimedia*, New York, NY, USA, 2014.

[87] L. Bottou, "Large-Scale Machine Learning with Stochastic Gradient Descent," in *19th International Conference on Computational Statistics*, Paris, France, 2010.

[88] Y. Jia and E. Shelhamer, "Fine-tuning CaffeNet for Style Recognition on “Flickr Style” Data," [Online]. Available: http://caffe.berkeleyvision.org/gathered/examples/finetune_flickr_style.html.

[89] G. Griffin, A. Holub and P. Perona, "Caltech-256 Object Category Dataset," 2007.

[90] F.-F. Li, R. Fergus and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," in *IEEE CVPR Workshop of Generative Model Based Vision*, 2004.

[91] M. Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams, J. Winn and A. Zisserman, "The PASCAL Visual Object Classes Challenge: A Retrospective," *International Journal of Computer Vision,* vol. 111, no. 1, pp. 98-136, 2015.

[92] M.-E. Nilsback and A. Zisserman, "Automated Flower Classification over a Large Number of Classes," in *Sixth Indian Conference on Computer Vision, Graphics and Image Processing, 2008*, 2008.

[93] H. M. Afkham, A. T. Targhi, J.-o. Eklundh and A. Pronobis, "Joint Visual Vocabulary for Animal Classification," in *Proceedings of the International Conference on Pattern Recognition*, Tampa, FL, USA, 2008.

[94] "Cross-validation (statistics)," [Online]. Available: https://en.wikipedia.org/wiki/Cross-validation_(statistics).

[95] K. Ron, "A study of cross-validation and bootstrap for accuracy estimation and model selection," *International Joint Conference on Artifical Intelligence,* vol. 14, no. 2, pp. 1137-1145, 1995.

[96] "Confusion matrix," [Online]. Available: https://en.wikipedia.org/wiki/Confusion_matrix.

[97] Y. Chai, V. Lempitsky and A. Zisserman, "A Bi-level Co-Segmentation Method for Image Classification," in *International Conference on Computer Vision*, 2011.

[98] K. Van De Sande, T. Gevers and C. Snoek, "Evaluating Color Descriptors for Object and Scene Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 32, no. 9, pp. 1582-1596, 2010.

[99] M. Oquab, L. Bottou, I. Laptev and J. Sivic, "Learning and transferrring mid-level image representations using convolutional neural networks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[100] K. Chatfield, K. Simonyan and A. Zisserman, "Return of the Devil in the Details: Delving Deep into Convolutional Nets," *arXiv preprint arXiv:1405.3531,* 2014.

[101] K. Sohn, D. Jung, H. Lee and A. Hero, "Efficient Learning of sparse, distributed, convolutional feature representations for object recognition," in *IEEE International Conference on Computer Vision*, 2011.

[102] L. Bo, X. Ren and D. Fox, "Multipath sparse coding using hierarchical matching pursuit," in *IEEE Conference on computer Vision and Pattern Recognition*, 2013.

[103] T. Lindeberg, "Edge Detection and Ridge Detection with Automatic Scale Selection," *International Journal of Computer Vision,* vol. 30, no. 2, pp. 117-157, 1998.

[104] A. Wills and Y. Sui, "An algebraic model for fast Corner Detection," in *IEEE 12th International Conference on Computer Vision*, 2009.

[105] K. Simonyan, A. Vedaldi and A. Zisserman, "Deep Fisher Networks for Large Scale Image Classification," *Advances in Neural Information Processing Ssytems,* pp. 163-171, 2013.

[106] L.-C. Chiu, T.-S. Chang, J.-Y. Chen and N. Y.-C. Chang, "Fast SIFT Design for Real-Time Visual Feature Extarction," *IEEE Transactions on Image Processing,* vol. 22, no. 8, pp. 3158-3167, 2013.

[107] S. K. Pal and S. Mitra, "Multilayer perceptron, fuzzy sets, and classification," *IEEE Transactions on Neural Networks,* vol. 3, no. 5, pp. 683-697, 1992.

[108] Y.-L. Boureau, J. Ponce and Y. Lecun, "A Theoretical Analysis of Feature Pooling in Visual Recognition," in *27th International Conference on Machine Learning*, Haifa, Israel, 2010.

[109] D. M. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," 2011.

[110] A. Oliva and A. Torralba, "Modeling the Shape of the Scene: A Holistic Representation of the Spatial Envelope," *International Journal of Computer Vision,* vol. 42, no. 3, pp. 145-175, 2001.