
Preface

General paradigm in solving a computer vision problem is to represent a raw image using a more informative vector called feature vector and train a classifier on top of feature vectors collected from training set. From classification perspective, there are several off-the-shelf methods such as gradient boosting, random forest and support vector machines that are able to accurately model nonlinear decision boundaries. Hence, solving a computer vision problem mainly depends on the feature extraction algorithm.

Feature extraction methods such as scale invariant feature transform, histogram of oriented gradients, bank of Gabor filters, local binary pattern, bag of features and Fisher vectors are some of the methods that performed well compared with their predecessors. These methods mainly create the feature vector in several steps. For example, scale invariant feature transform and histogram of oriented gradients first compute gradient of the image. Then, they pool gradient magnitudes over different regions and concatenate them in order to create the final feature vector. Similarly, bag of feature and Fisher vectors start with extracting a feature vector such as histogram of oriented gradient on regions around bunch salient points on image. Then, these features are pooled again in order to create higher level feature vectors.

Despite the great efforts in computer vision community, the above hand-engineered features were not able to properly model large classes of natural objects. Advent of convolutional neural networks, large datasets and parallel computing hardware changed the course of computer vision. Instead of designing feature vectors by hand, convolutional neural networks learn a composite feature transformation function that makes classes of objects linearly separable in the feature space.

Recently, convolutional neural networks have surpassed human in different tasks such as classification of natural objects and classification of traffic signs. After their great success, convolutional neural networks have become the first choice for learning features from training data.

One of the fields that have been greatly influenced by convolutional neural networks is automotive industry. Tasks such as pedestrian detection, car detection, traffic sign recognition, traffic light recognition and road scene understanding are rarely done using hand-crafted features anymore.

Designing, implementing and evaluating are crucial steps in developing a successful computer vision-based method. In order to design a neural network, one must have the basic knowledge about the underlying process of neural network and training algorithms. Implementing a neural network requires a deep knowledge about libraries that can be used for this purpose. Moreover, neural network must be evaluated quantitatively and qualitatively before using them in practical applications.

Instead of going into details of mathematical concepts, this book tries to adequately explain fundamentals of neural network and show how to implement and assess them in practice. Specifically, Chap. 2 covers basic concepts related to classification and it derives the idea of feature learning using neural network starting from linear classifiers. Then, Chap. 3 shows how to derive convolutional neural networks from fully connected neural networks. It also reviews classical network architectures and mentions different techniques for evaluating neural networks.

Next, Chap. 4 thoroughly talks about a practical library for implementing convolutional neural networks. It also explains how to use Python interface of this library in order to create and evaluate neural networks. The next two chapters explain practical examples about detection and classification of traffic signs using convolutional neural networks. Finally, the last chapter introduces a few techniques for visualizing neural networks using Python interface.

Graduate/undergraduate students as well as machine vision practitioners can use the book to gain a hand-on knowledge in the field of convolutional neural networks. Exercises have been designed such that they will help readers to acquire deeper knowledge in the field. Last but not least, Python scripts have been provided so reader will be able to reproduce the results and practice the topics of this book easily.

Books Website

Most of codes explained in this book are available in <https://github.com/pcnn/>. The codes are written in Python 2.7 and they require *numpy* and *matplotlib* libraries. You can download and try the codes on your own.

Tarragona, Spain

Hamed Habibi Aghdam

<http://www.springer.com/978-3-319-57549-0>

Guide to Convolutional Neural Networks

A Practical Application to Traffic-Sign Detection and
Classification

Habibi Aghdam, H.; Jahani Heravi, E.

2017, XXIII, 282 p. 150 illus., 111 illus. in color.,

Hardcover

ISBN: 978-3-319-57549-0