ACTIVE CONTROL OF CAMERA PARAMETERS AND ALGORITHM SELECTION FOR OBJECT DETECTION

YULONG WU

A THESIS SUBMITTED TO
THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

GRADUATE PROGRAM IN
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO

January 2017

Abstract

In this thesis, we quantitatively investigate the effect of camera parameters, shutter speed and voltage gain, on the performance of several popular object detection algorithms, under various illumination conditions. Our experimental results indicate a significant difference in sensitivity of the evaluated algorithms to these camera parameters. Based on the experimental benchmark results, a novel active control of camera parameters method and an algorithm selection extension are proposed. In empirical evaluation, our active control approach outperforms the conventional auto-exposure method for most algorithms. Also, the proposed algorithm selection extension has demonstrated the capability of selecting a proper $\langle algorithm, shutter, gain \rangle$ tuple, in order to deal with varying light conditions.

Acknowledgements

I would like to thank my supervisor, John K. Tsotsos, for his support and guidance. He is a superb scientist and mentor who gave me invaluable a throughout the development and writing of this thesis. I am also grateful to my supervisory committee member, Richard P. Wildes, for his critical comments and precious advice, and to my labmate Mahdi Biparva who greatly helped with my understanding of neural networks. Finally, I would like to thank my parents and friends. Without their patience, understanding, and support, I could not have done this.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Background

1.1.1 Active Vision

In the 1980s, Bajcsy [4] introduced the concept of active perception, as "a problem of intelligent control strategies applied to the data acquisition process". This idea was later explored and termed "active vision", with an emphasis on visual perception, by Aloimonos et al. [5]. In their studies, it was shown that many vision problems, especially shape estimation and depth computation, could be solved in a much more efficient way by an active observer than a passive one, for which these problems are ill-posed. Active vision was later formalized as a special case of the attention problem by Tsotsos [6], which is observed in the human visual system.

Despite the advantages of being "active", most vision guided robotic systems are characterized by their passive perspectives. First, the datasets that they are trained on, e.g. [7] [8] [9] [10], are camera sensor biased [11]. The trained vision algorithms often fail in real-world applications, due to the variety of illumination conditions they may face. This leads to the question: how would object detection algorithms perform on poor exposed images, especially for extreme low or high illumination? Second, these robotic systems are relying on camera's built-in auto-exposure algorithms and show poor results in uncontrolled environments [12], to deal with varying light conditions.

There are a number of researchers that are working in the area of active vision to progress it further. Dickinson et al. [13] proposed to integrate attention and viewpoint control into an object recognition/detection framework. Lu et al. proposed a method of adjusting camera parameters based on image entropy [14]. Browatzki et al. [15] applied active object recognition to humanoids. Interested reader should also see [16].

1.1.2 The Image Formation Process

As the first component of a typical vision guided robotic system, image acquisition is very important for the success of specific vision-related tasks. In this section, we review the key components of a typical image formation pipeline, which is illustrated

in Figure 1.1, and related parameters that affect camera exposure.

Starting from light sources, photons reflect off the surface of objects, and then go through the optics (lens) and finally hit the image sensor. Typically, there are two types of sensors: charge-coupled device (CCD) and complementary metal-oxide on silicon (CMOS). In the image sensor, photon-induced charge is accumulated, which is later transferred to a *charge amplifier* and converted into a voltage. The voltage is further quantized into an integer by an analog-to-digital converter (ADC). For more information of this physical process, we refer the reader to [17] [18].

There are many factors that contribute to the intensity of each pixels in an image, from the physical direction of light sources, to the reflection coefficient and surface normals of object surfaces, and to the internal parameters of digital cameras. For camera parameters, four of them are discussed here. The first one is the aperture of an optical system, which determines the cone angle of a bundle of rays that come to a focus in the image plane. A smaller aperture lets fewer incoming rays go through the optics and results in darker images, while a larger aperture results in brighter images. The second one is shutter speed, or exposure time, which is the length of time when the digital sensor inside the camera is exposed to light. With a faster shutter speed, more photons are accumulated at the image sensor, which results in darker images. The third one is voltage gain, which determines the degree to which the electronic signal is amplified. The gain control is useful for adjusting

the intensity of acquired images, though it may magnify sensor noise [19].

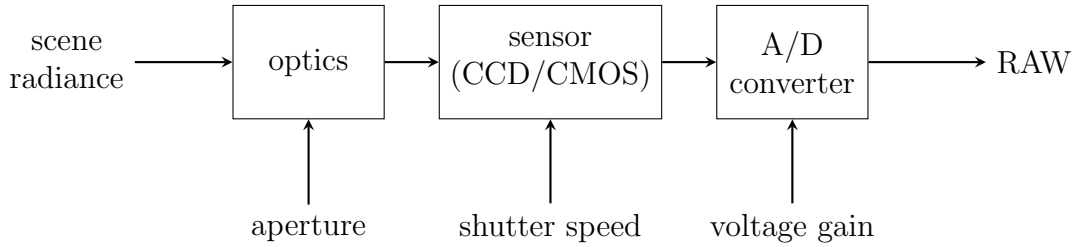


Figure 1.1: The image formation pipeline in a digital camera.

To make a camera adaptive to variant light conditions, the aforementioned camera parameters need to be set properly. Inappropriate configurations may result in images with blooming/saturation effects or low contrast. Unlike the human vision system, where a number of complex mechanisms [20] [21] are used to compensate for luminance changes, modern cameras are equipped with algorithmic sensor configuration modules. For example, shutter speed and voltage gain are typically controlled by the auto-exposure controllers, which are based on the mean brightness of region-of-interest (ROI) in the perceived image. While these methods could result in *good* images from the perspective of a human, it is not always the case for a robot [14].

1.1.3 Object Detection and Camera Parameters

One of the major concerns in this thesis is that the control of camera parameters is isolated from the choice of object detection algorithms. The effect of camera

parameters, i.e. shutter speed and voltage gain, on the performance of object detection algorithms has been largely overlooked in the literature. This is primarily due to the wide utilization of offline image datasets, where the intrinsic camera parameters are often unknown or non-uniform. On the contrary, by quantitatively evaluating various object detection algorithms, we found that these parameters play an important role in determining the performance of the algorithms. With this in mind, we propose an active control of camera parameters method, which is illustrated in Figure 1.2, to improve the robustness and adaptivity of vision algorithms under varying light conditions.

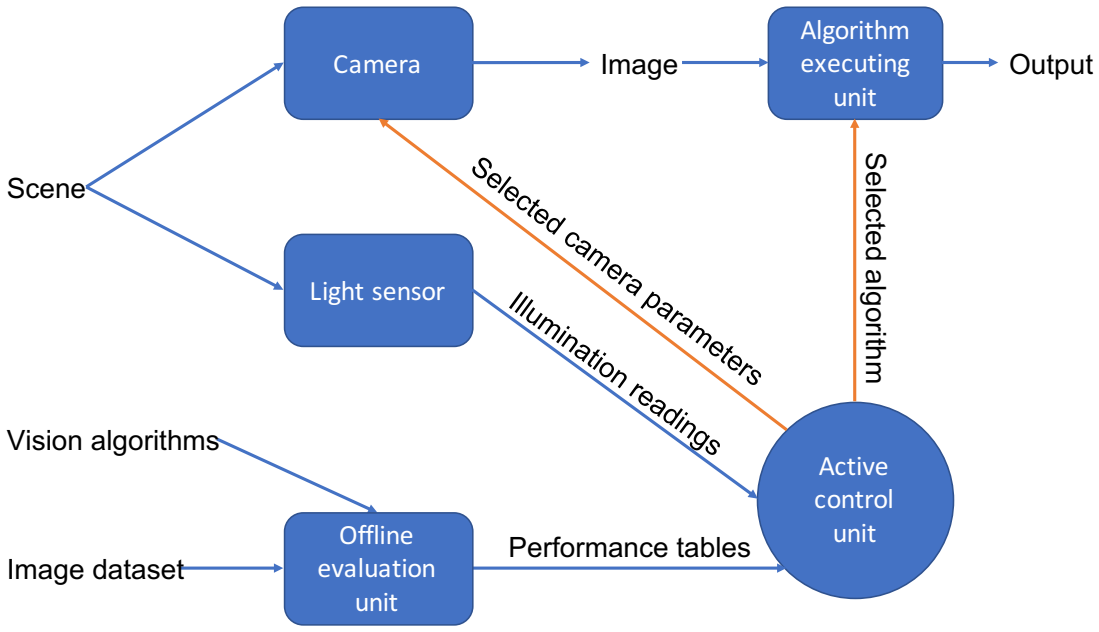


Figure 1.2: The proposed active control of camera parameters framework.

With increasing attention to the area of object detection over the years, the pool

of publicly available object detection algorithms has been expanding. Research on how to effectively utilize the abundance of available algorithms has become more popular. In the literature, there are two common approaches: one is fusing different detectors as reviewed in [22], the other is automatic selection based on predefined constraints, like [23]. In this thesis work, we focus on the second approach.

1.1.4 Related Work

This thesis follows the studies on comparing interest point detectors and saliency algorithms to uncover sensor bias by Andreopoulos et al. [11]. In their work, a new image dataset was created for four different scenes under variant light conditions, by uniformly sampling the shutter speed and voltage gain parameters. Five interest-point detectors and two saliency algorithms (the Harris-Affine and Hessian-Affine detectors [24], Kadir and Bradys detector [25], the MSER detector [26], SURFs detector [27], the Itti-Koch-Niebur saliency algorithm [28], and the AIM saliency algorithm [29]) were evaluated over the aforementioned dataset.

By quantitative analysis, we found that a generic camera parameters controlling strategy could not guarantee reliable results for all light conditions and vision algorithms. Purposive control of sensor parameters is required for a robust vision guided robotic system. Moreover, their experimental results indicated that:

> Offline datasets used to evaluate vision algorithms, typically suffer from a significant sensor specific bias which can make many of the exper-

imental methodologies used to evaluate vision algorithms unable to provide results that generalize in less controlled environments. Active and purposive control of the shutter speed and gain can lead to significantly more reliable feature detection under variant illumination and non-constant viewpoints.

Motivated by these observations and arguments, this thesis continues to investigate the effect of shutter speed and voltage gain on the performance of object detection algorithms and to demonstrate how active control could improve high-level vision algorithms, compared with conventional methods.

1.2 Scope of Study

In this thesis, we mainly investigate the sensitivity of object detection algorithms to camera parameters and possible approaches to work around this problem. It is intended for vision guided robotic systems, especially those that are designed for object detection tasks and need to adapt to different light conditions.

For object detection algorithms, we do not study how to improve the algorithms by designing new features, changing their structure, or retraining on other datasets. Also, the number of examined algorithms is limited to four.

Only two camera parameters, i.e. shutter speed and voltage gain, are studied. Other camera parameters and techniques are beyond the scope of this work, including but not limited to, focal length, aperture, brightness, contrast, hue, saturation,

gamma, white balance, pixel format/resolution, image format.

1.3 Contributions

There are three main contributions of this thesis. The first one is the quantitative analysis of the performance of four object detection algorithms with respect to variant ambient illumination, shutter speed and voltage gain, which is detailed in Chapter 3. A significant difference in sensitivity of these algorithms to illumination, shutter speed and voltage gain has been observed.

The second one is the proposed active control of camera parameters, which is detailed in Section 5.2. In empirical evaluation, our approach significantly outperforms the conventional camera's built-in auto-exposure algorithm for most evaluated algorithms.

The third one is the proposed algorithm selection extension. This approach has allowed us to select the best-performing algorithm and camera parameters under different light conditions.

1.4 Thesis Outline

This thesis is organized into six chapters:

- Chapter 1 introduces the topic of the thesis and overviews related research,

especially the active vision paradigm and image formation process. Also, the motivation and significance of the work is presented.

- Chapter 2 reviews four popular object detection algorithms.
- Chapter 3 presents the quantitative analysis of the performance of object detection algorithms with respect to variant illumination, shutter speed and voltage gain.
- Chapter 4 evaluates one of the illumination preprocessing techniques, i.e. the Laplacian-of-Guassian (LoG) enhancement technique.
- Chapter 5 describes the proposed active control of camera parameters approach and its algorithm selection extension.
- Chapter 6 details the empirical evaluation of the proposed approaches.
- Chapter 7 concludes this thesis and presents possible future work.

Chapter 2

Object Detection

2.1 What is Object Detection?

Object detection is a visual recognition task, which identifies and localizes instances of target objects in an image [30]. It answers not only what the objects are but also where they are located, with location often represented by bounding boxes. This task is difficult for many reasons and the major challenges include: 1) arbitrary background; 2) occlusion by other objects; 3) different viewpoints; 4) variant object scales, appearances and positions; 5) deformation and intra-class variation; and 6) variant light conditions.

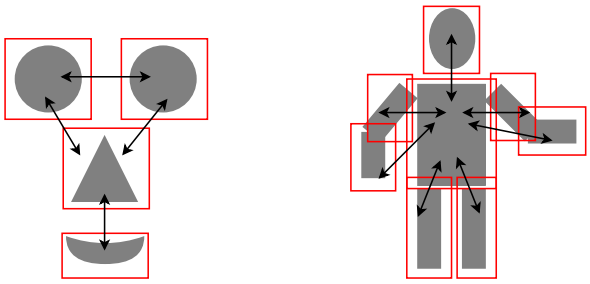
Despite the difficulties, consistent attention has been drawn to this topic during the past thirty years in the computer vision community, and a spectrum of algo-

rithms have been proposed. Based on their performance on the PASCAL VOC challenges [8], four algorithms – *deformable part models*, *bag-of-words with spatial pyramid matching*, *regions with convolutional neural networks*, and *spatial pyramid pooling in deep convolutional networks* – are reviewed in the following sections. For comprehensive surveys on object detection and recognition, we refer the reader to [31, 32, 33, 34, 35].

2.2 Object Detection Algorithms

2.2.1 Deformable Part Models

The deformable part models (DPM) [1] is based on the pictorial structures framework [36] [37], in which visual objects are represented by a set of parts arranged in a deformable configuration. Each part captures the local appearance properties of a part of an object, while deformable configurations encode the spatial relations between parts. Two examples of the model are presented in Figure 2.1.



(a) Human face (b) Pedestrian

Figure 2.1: Deformable part models for human face and pedestrian. Each rectangle represents a part while the lines denote relative spatial relations.

The deformable part models can also be interpreted as an undirected graph $G = (V, E)$, where the vertices $V = \{v_1, ..., v_n\}$ represent the n parts and the edges E represent the connections between parts. An instance of object is given by a configuration $L = (l_1, ..., l_n)$, where each l_i describes the spatial location of the part v_i relative to the center of an object. The goal of this model is to find the optimal configuration L^* that maximizes the similarity between instance parts and the corresponding model parts while minimizing the overall cost of deformation. It can be formulated as

$$L^* = \arg\min_L \Big(\sum_{n=1}^{n} m_i(l_i) + \sum_{d_{ij}(l_i,l_j) \in E} d_{ij}(l_i, l_j) \Big),$$

where m_i is the match cost function and d_{ij} is the deformation cost function. This minimization problem is NP-Hard [38], without limiting the structure of the graph

G. Felzenszwalb et al. [1] managed to compute the optimal match in polynomial time by restricting the parts to a tree structure, specifically a star graph where the root is at a coarser resolution.

In DPM, an object is represented by a coarse root filter that covers the entire object and four high resolution part filters that cover parts of the object. A filter is a rectangular template which applies to the feature map of an image (a feature map is a matrix of d-dimensional feature vectors computed at densely sampled locations of an image). The response, or score, of a filter F at a location (x, y) in a feature map G is defined as the dot product of the filter and a subwindow of the feature map with top-left corner at (x, y),

$$\sum_{x',y'} F[x', y'] \cdot G[x + x', y + y'].$$

Filters are trained using latent SVM with a stochastic gradient descent approach and data mining technique. See details in [1].

The matching process of DPM is demonstrated in Figure 2.2. Given an input image and trained models, the system first compute a histogram of oriented gradients (HoG) [39] feature pyramid (a feature pyramid is a set of feature maps at different scales). The response of root and part filters are then computed. Part filter responses are further transformed to allow for spatial uncertainty, which spreads high filter scores to nearby locations , with a penalty for the deformation cost. Finally, the root filter response and the transformed part filter responses are combined

to generate the probability of the objects existing.

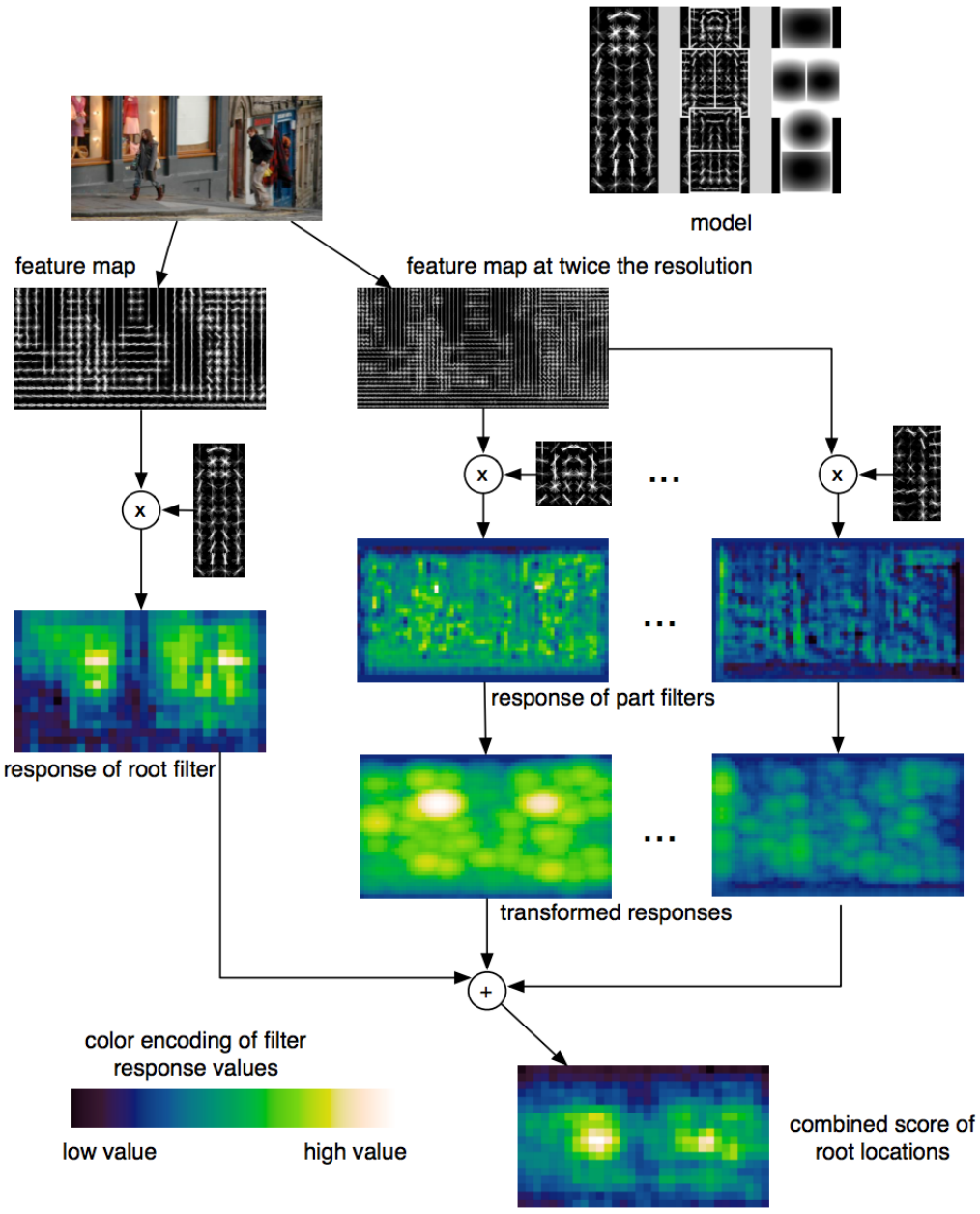


Figure 2.2: The matching process of the deformable part models [1].

This model is capable of representing highly variable objects, and is accurate in predicting bounding boxes. However, the DPM often suffers from difficulties in training, as the local parts are not labeled in the training dataset and need to be treated as latent variables during the training phase.

2.2.2 Bag-of-Words with Spatial Pyramid Matching

The bag-of-words (BoW) model was first introduced to natural language processing and later applied to computer vision tasks [40]. In this model, local features are treated as visual words, and an image is represented as an encoding of the visual words. To compute the BoW representation, typical steps include: (1) Detect interest points/regions or apply dense sampling; (2) Compute feature descriptors around the local patch of interest; (3) Build a visual vocabulary (k-means cluster centers in case of Vector of Locally Aggregated Descriptors (VLAD) [41] encoding, or Gaussian Mixture Models (GMM) for Fisher encoding); (4) Compute an encoding for each spatial region and the final representation is achieved by pooling or stacking the encodings. This pipeline is illustrated in Figure 2.3.

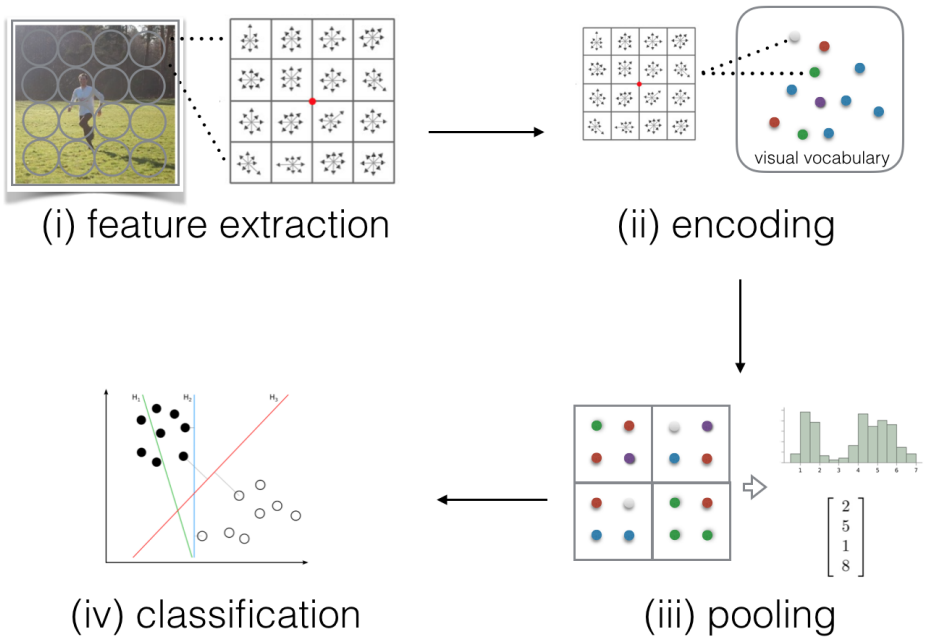


Figure 2.3: A typical pipeline of the bag-of-words model.

Interest Point Detection

The purpose of interest point detection is to find the interest points/regions that are scale/viewpoint invariant, which often reduce the computation required. Common approaches include the Harris corner detector, the Harris-Laplace region detector and the Difference-of-Gaussians region detector, which are reviewed in [42] [43].

In recent studies, it was shown that simple-minded dense sampling often outperforms the interest point based sampling for various object recognition algorithms [44]. In addition, the number of extracted interest points strongly affects the effi-

ciency of BoW representations.

Local Descriptors

The next step of BoW is to describe the local patch around the interest points/regions or densely-sampled points. The most widely used approach is the scale-invariant feature transform (SIFT) [45] descriptor.

The SIFT descriptor is obtained by dividing the 16x16 neighborhood around an interest point into 16 sub-blocks and computing a histogram of gradient for each sub-block. The dimensionality of SIFT descriptor is 128. Often, principal component analysis (PCA) is used to reduce the dimensionality [46].

Encoding

Once local descriptors are computed, the next step is to build a visual vocabulary and encode local features to generate fixed-length representations.

There are two common ways of building the visual vocabulary, k-means clustering and GMM clustering. K-means clustering partitions the local descriptor space into informative regions, and local descriptors are assigned to the closest (in Euclidean distance) center of cluster. In GMM clustering, a GMM is the probability

density on R^D given by

$$p(\mathbf{x}|\theta) = \sum_{k=1}^{K} p(\mathbf{x}|\mu_k, \mathbf{\Sigma}_k)\pi_k, \quad p(\mathbf{x}|\mathbf{u}_k, \mathbf{\Sigma}_k) = \frac{1}{\sqrt{(2\pi)^D det\mathbf{\Sigma}_k}} e^{-\frac{1}{2}(\mathbf{x}-\mu_k)^T \sum_k^{-1}(\mathbf{x}-\mu_k),}$$

where $\theta = (\pi_1, \mu_1, \mathbf{\Sigma}_1, ..., \pi_k, \mu_k, \mathbf{\Sigma}_k)$ is the vector of parameters, π_k is the prior probability values, μ_k is the mean values, $\mathbf{\Sigma}_k$ is the positive definite covariance matrices. Local descriptors are assigned to the multivariate normal components that maximize the component posterior probability given the data.

Common encoding methods include Histogram Encoding (VQ) [47], Kernel codebook encoding (KCB) [48, 49], Locality constrained linear coding (LLC) [50], Fisher encoding (FK) [51] and Supervector encoding (SV) [52]. For a comprehensive evaluation of these encoding methods, we refer the reader to [53].

Spatial Pooling

One of the successful extensions to the BoW model is spatial pyramid matching [54], which partitions an image into increasingly fine sub-regions and compute an encoding for each sub-region. The final BoW representation is the concatenation of these encodings. This approach gives the orderless BoW representation the ability to encode spatial information, and has been demonstrated useful despite its simplicity.

2.2.3 Regions with Convolutional Neural Networks

In 1990, LeCun et al. published the LeNet-5 convolutional neural network (CNN), which was able to classify handwritten digits and has motivated most modern CNN frameworks. In this model, there are multiple layers, which can be trained by the backpropagation algorithm [55]. However, due to the lack of training data and computing power, LeNet-5 did not perform well on complex problems, e.g., large-scale image classification.

Since then, significant progress has been made in machine learning and computer vision communities. Notably, Krizhevsky et al. proposed the AlexNet [2], which used deep neural networks and large-scale training data. To make training faster, they used non-saturating neurons and a very efficient GPU implementation. AlexNet demonstrated high accuracy in object recognition tasks on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [9]. In 2014, Girshick et al. [56] proposed the Regions with Convolutional Neural Networks (R-CNN) for object detection tasks.

In the R-CNN, there are three major components, which are illustrated in Figure 2.4. The first one is the computation of object-independent region proposals. Around 2000 region proposals are extracted from an input image by the selective search algorithm [57], which is based on a set of complementary and hierarchical

grouping strategies. In selective search, initial regions are the generated by the Felzenszwalb and Huttenlocher segmentation method [58]. Then, the similarity between regions and their neighbors are computed. The most similar regions are grouped. This process is repeated until the while image becomes a single region.

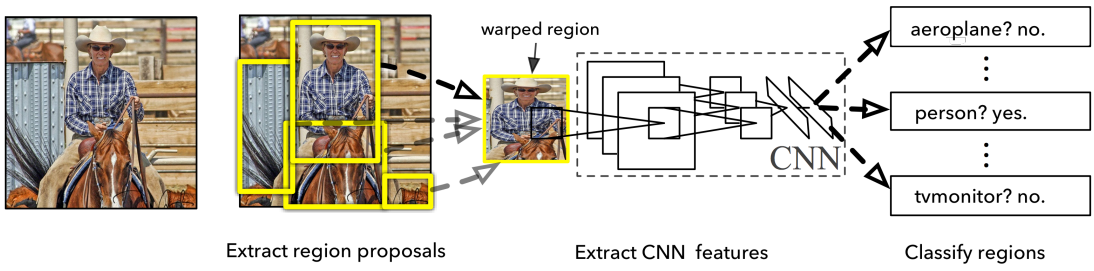


Figure 2.4: The processing pipeline of R-CNN [2].

The second component is to extract fixed-length features for each region from a large convolutional neural network. Typically, the dimension of input of a convolutional network is fixed. However, the size and aspect ratio of region proposals varies. To address this issue, image regions are wrapped to the required size (originally 227 x 227 pixel size). The CNN is pre-trained on the ILSVRC 2012 dataset and fine-tuned by the PASCAL VOC dataset. The pooling layer $pool_5$ features are extracted for classifying image regions.

The third component is a set of linear SVM classifiers , each of which is trained for one class. Given an input image, all classifiers are applied to each of the region proposals. After that, the results are combined using non-maximum suppression

(if a positive region has a large intersection-over-union (IoU) with another positive region that has a higher score, it will be removed).

2.2.4 Spatial Pyramid Pooling in Deep Convolutional Networks

As spatial pyramid matching boosts the performance of BoW models, as described in Section 2.2.2, this idea was later applied to deep convolutional neural networks in [3] (denoted as SPP-net).

Considering the popular seven-layer convolutional network architectures [2] [59], there are five convolutional layers and two fully-connected layers. For the convolutional layers, they can take an input of an arbitrary size, as the convolution filters are applied to the input in a sliding manner. However, in the fully-connected layers, the input size has to be fixed. In SPP-net, a spatial pyramid pooling layer is added between the last convolutional layer and the first fully-connected layer, as shown in Figure 2.5, which results in a CNN that accepts variable size of input.

Besides avoiding artificially warping an input image, SPP-net is also much faster than R-CNN. The feature maps are computed only once for the entire image, and are pooled over arbitrary regions-of-interest (region proposals in the case of object detection) to generate fixed-length representations.

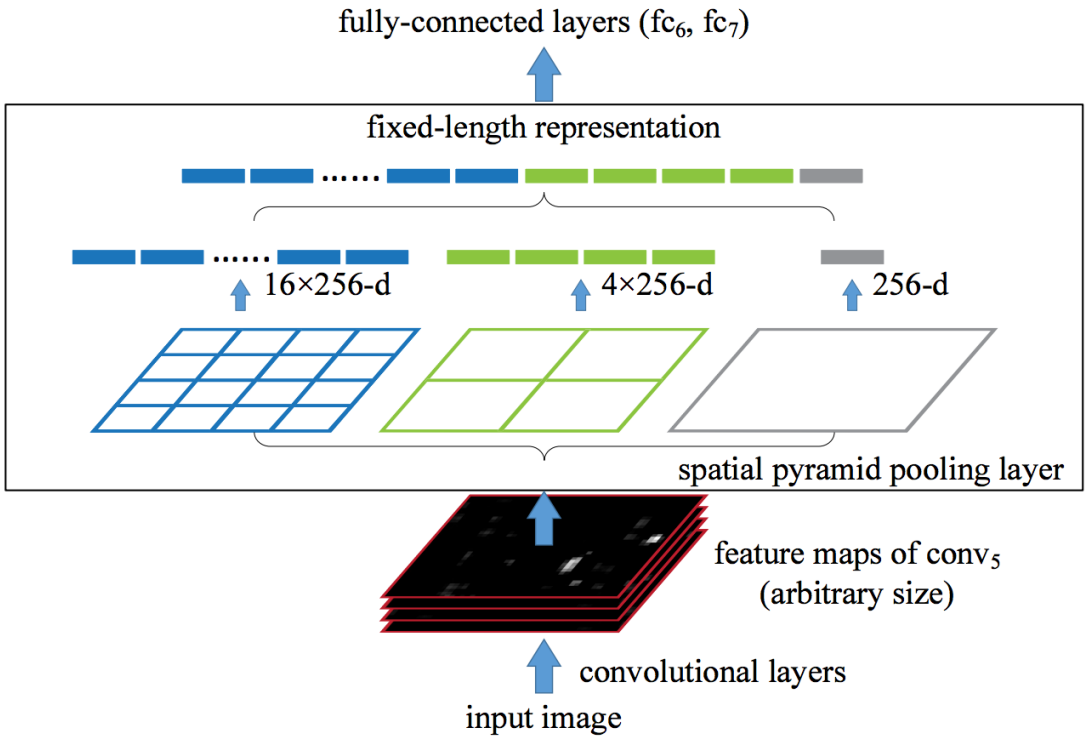


Figure 2.5: A neural network structure with a spatial pyramid pooling layer [3].

2.3 Compensating for Illumination Changes

A slight change in ambient illumination may cause an object to appear differently, which makes robustness to illumination variation a challenging task. To achieve this goal, constant efforts have been made in the computer vision and robotics communities. See [60, 61, 62, 63, 64, 65].

There are four common approaches to dealing with varying light conditions. The first one is based on relatively illumination-insensitive representations of an

image, such as edge maps [66], features in the frequency domain derived for a differentiated image [67], and inferred albedo and surface normal from neural networks [68]. Better illumination invariance could be achieved by using these representation instead of the original image.

The second approach is to use multiple instances-based models, where each instance corresponds to one lighting condition. Belhumeur [69] proved that the set of images of an object in fixed pose but with variant illumination, forms a convex cone, and the dimension of this illumination cone equals the number of distinct surface normals. However, algorithms based on this approach typically need large amount of training data and have high computational cost.

The third approach is camera sensor accommodation, which dates back to the 1970s [70]. It was proposed that sensor accommodation, automatic control by computer over the parameters of camera, should be an integral part of the recognition process. This idea was later applied on active fixation in the context of object recognition [71].

The last one is illumination preprocessing. Preprocessing has been a common procedure in object recognition pipelines, which aims to improve the reliability of a vision system. For face recognition particularly, a study [72] demonstrated that illumination preprocessing is helpful in handling illumination variations. In this thesis, we investigate one of the illumination preprocessing approaches, i.e. the

Laplacian-of-Gaussian (LoG) enhancement technique.

2.4 Summary

In this chapter, four object detection algorithms were reviewed. No modules were found in these algorithms to deal with varying illumination. In DPM, the low-level features are based on HoG, which could be contrast-normalized for better invariance to changes in illumination and shadowing [39]. In BoW, local features are represented by SIFT descriptors, which are weak with respect to illumination invariance [73]. R-CNN and SPP-net are both based on convolutional neural networks which strongly depend on the intensity of input images. Given the fact that they are trained on databases where the majority of images are under normal illumination, and are not trained/fined-tuned on images with uniform distribution of illumination, limited tolerance to changes in illumination is expected. Specific quantitative evaluation of these algorithms is presented in Chapter 3.

In addition, four common approaches to illumination compensation have been discussed, which provide clues about how to solve this problem. The evaluation of the LoG preprocessing method is described in Chapter 4.

Chapter 3

Quantitative Analysis

As introduced in Section 2.1, one of the challenges for the object detection task is the sensitivity to varying light conditions. However, few quantitative analysis of the effect of illumination and sensor configuration have been conducted in the literature. In this chapter, we present our experiments that examine to what extent these two factors affect the performance of object detection algorithms. First, we introduce a new image dataset that incorporates the ambient light conditions and sensor configurations in Section 3.1. Then, the detailed experimental protocol and setup are described in Section 3.2. Finally, the results and discussions are presented in Section 3.3.

3.1 Dataset

3.1.1 Overview

The created image dataset includes 2240 images in total, by viewing 5 different objects (*bicycle*, *bottle*, *chair*, *pottedplant* and *tvmonitor*), under 7 illumination conditions and with 64 camera configurations (8 shutter speeds $\times$ 8 voltage gains). Each image is in 8-bit/color RGB format and of 1280x1204 resolution. Every object instance in the dataset is annotated with a label and a bounding box. Samples of this dataset can be found in Table 3.1 and 3.2.

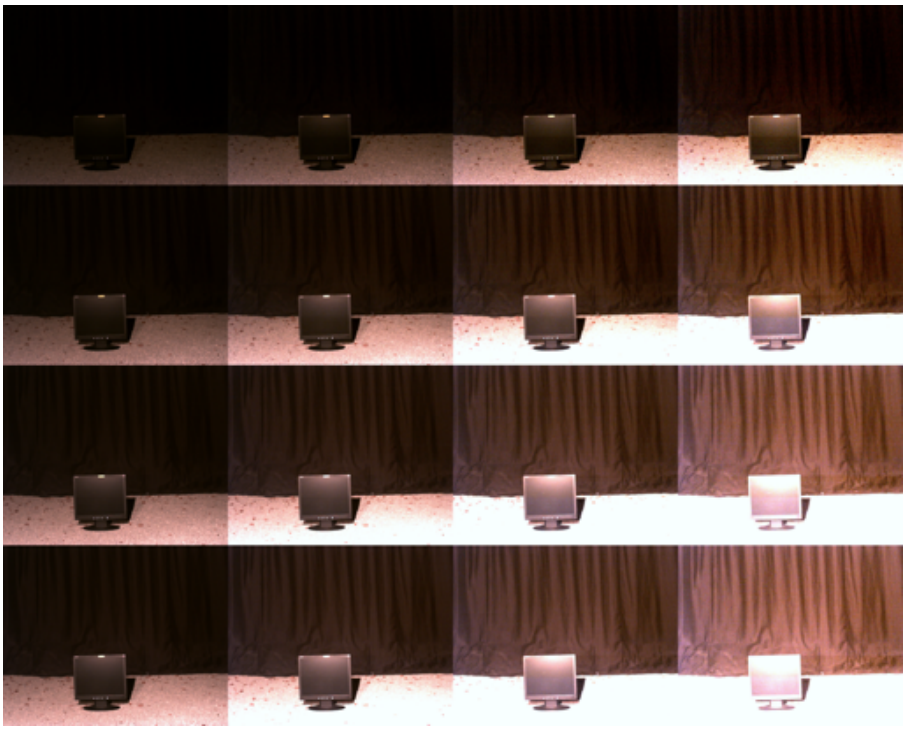


Table 3.1: Images of *tvmonitor* at 800lx illumination with different camera configurations. The shutter speed increases from top to bottom, and the voltage gain increases from left to right.

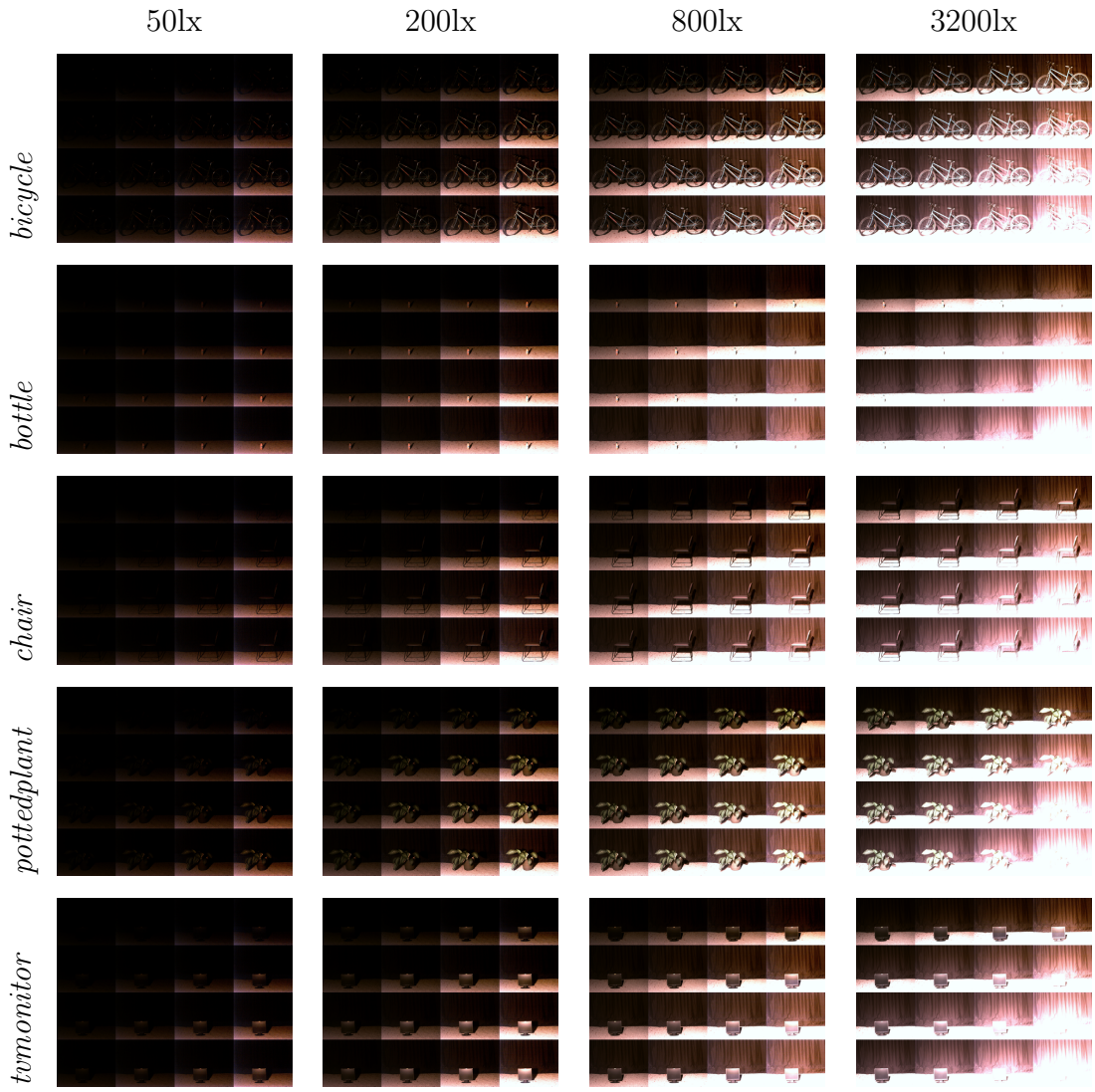


Table 3.2: Sample images from the dataset. There are five objects in total and each of them is pictured with seven (but only four are shown) different light conditions. Each table cell contains a matrix of images, resulting from different camera configurations, same as in Table 3.1. (Best viewed on high-resolution display.)

3.1.2 Data Acquisition

To properly control the ambient illumination of an object, our dataset was created in a lab environment completely enclosed by blackout curtains to eliminate stray ambient illumination. The major components included a digital camera, two light bulbs, a light sensor and other decorations, see Figure 3.1 for their relative positions. Also, there was a laptop connected to the camera for controlling the shutter speed and voltage gain of the camera.

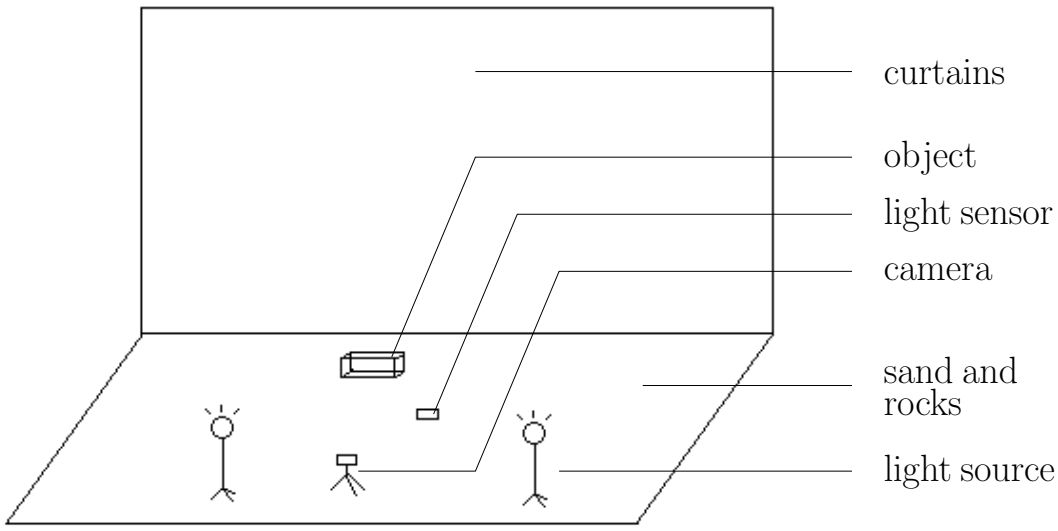


Figure 3.1: Stage setup for creating the dataset.

The camera we used was a Point Grey Flea3 camera(mode: FL3-U3-13E4C-C), which was equipped with a CMOS sensor and an API interface. The allowed shutter speed and voltage gain range were 0.016ms-24.973ms and 0dB-24.014dB

respectively. These permissible ranges were uniformly sampled into 8 distinct values in each dimension. The i^{th} sample from a range $[a, b]$ was set as $a + \frac{b-a}{8}(i-1)$, where $i \in \{1, ..., 8\}$, leading to 8×8 candidate settings for the shutter/gain parameters, under which images were acquired. The aperture was fixed at 4, and the red and blue white-balancing channels were set to 500 and 800 respectively. All other parameters were kept at default values. For the detailed camera specifications, we refer the reader to Appendix A.

Intensity-controllable light bulbs were used to control the illumination of scene with no additional light sources, and a Yoctopuce light sensor was used for measurements. The selected incident-light levels were 50lx, 200lx, 400lx, 800lx, 1600lx and 3200lx.

3.2 Experimental Setup

We evaluated the performance of four object detection algorithms (DPM, BoW, R-CNN and SPP-net), with respect to variant illumination, shutter speed and voltage gain configurations. The original implementations were used for all the algorithms except BoW.

3.2.1 Algorithm Setup

All the algorithms were required to detect the five objects in the dataset introduced in Section 3.1, for a given input image. To make their results comparable, the outputs were required to be a list of bounding boxes, associated with the labels and confidence scores, similar to the PASCAL VOC challenge. No optimization or transfer learning techniques were applied.

For the DPM, we used the Release 5 version implementation as published at `https://people.eecs.berkeley.edu/~rbg/latent/`. There were twenty class-specific detectors trained on the PASCAL VOC 2007 dataset. However, only five of them were used to detect the objects in our dataset. The outputs of each detector were combined using non-maximum suppression with a 0.5 overlap threshold.

For the BoW, we replicated the idea in [57] with our own implementation. To make it consistent with the other algorithms, it was trained only on the PASCAL VOC 2007 dataset [8]. Local features were sampled densely over the images and represented by SIFT and HoG descriptors. We used a visual book size of 1000 and a spatial pyramid with 3 levels using 1x1, 2x2, and 4x4. For the classifiers, we used linear SVMs with a chi-squared kernel. The training procedure is illustrated in Figure 3.2. The positive examples come from the ground-truth bounding boxes of the object of interest and the negative examples are from the ground-truth bounding

boxes of the other objects and also randomly selected regions which contain no objects (as suggested in [74]). Once the classifier has been trained, it is retrained in the hard-negative mining phase. The classifier is applied to the regions generated by the selective search algorithm. The false positives with highest score are added to the negative examples and the classifier is retrained. We repeated this retraining process for two iterations.

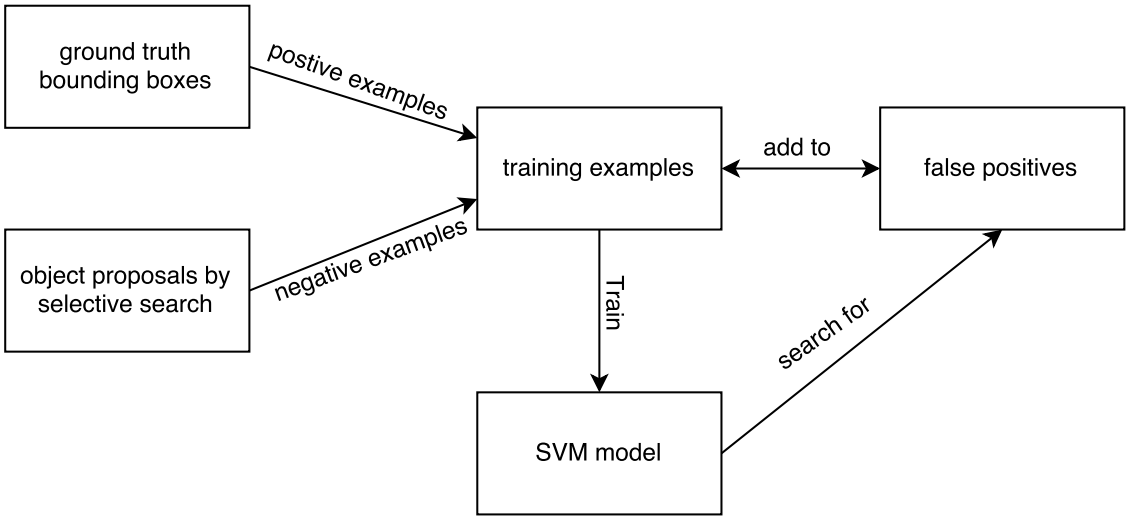


Figure 3.2: Demonstration of the training procedure.

we also experimented with the improved Fisher Vector (signed square-rooting followed by L^2 normalization). Due to computing and memory limit, we were only able to use a spatial pyramid with 2 layers, i.e. 1x1 and 2x2. The vocabulary size of the GMM was 128 (although [53] has suggested using 256 for the PASCAL dataset). The resulting dimension of image descriptors was 163840. Both the linear

and χ^2 kernels have been tested. However, the results are not promising. After the first training phase and right before the retraining process, the performance of the learned classifiers on the *testset* is 86.55% (the previous setup achieved 87.06%). Also, the computation of FV is time-consuming, making it less practical for object detection where we need to compute a FV for each region proposal. While the results could be possibly improved by optimizing parameters and introducing other feature descriptors, we stopped here.

For the R-CNN and SPP-net, the neural network was pre-trained on ImageNet and fine-tuned on the PASCAL VOC 2007 dataset. Twenty linear SVM classifiers were trained for the twenty classes of object in the PASCAL dataset. When evaluating on our dataset, only five classifiers (which correspond to the five objects in our dataset) were used, and the outputs were combined by non-maximum suppression with an overlap threshold of 0.5.

3.2.2 Evaluation Procedures

The output, given an input image, of each algorithm was required to be a list of predicted object instances, each represented by a bounding box, a level and a confidence score. A predicted instance is considered true if the label is correct and the bounding box overlaps no less than 50% with the ground-truth bounding box, otherwise false.

Following the methodology by Andreopoulos & Tsotsos [11], the evaluation procedures include:

1. Run the object detection algorithms on all the images that correspond to each $\langle illumination, shutter, gain \rangle$ combination;

2. Sort the outputs by their confidence scores and then evaluate them, using the aforementioned rule;

3. Compute the precision-recall curve from the above results;

4. Compute the average precision (AP) by sampling the precision-recall curve.

The final results are represented by performance tables. A performance table is a 8x8 matrix M, where M_{ij} is the AP of an algorithm on all the images that correspond to i^{th} sample of shutter speed and j^{th} sample of voltage gain, for a illumination. The range of AP is $[0, 1]$. Larger APs are represented in black color, and smaller are in white color.

3.3 Results and Discussion

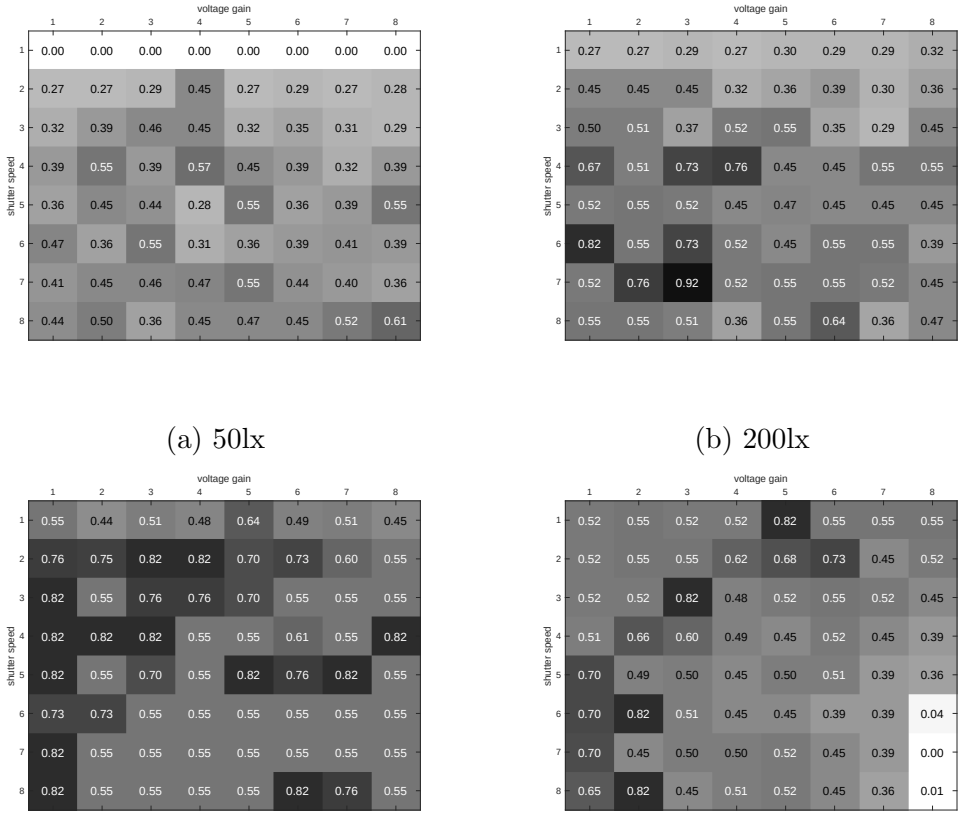
3.3.1 Overview

In this section, we present the results of the evaluated algorithms (DPM, BoW, R-CNN and SPP-net) on our dataset. The overall performance of each algorithm is

presented in Figure 3.3 - 3.6, where the numbers in each table represent the AP of an algorithm on all the images that are taken with the corresponding illumination, shutter speed and voltage gain.

The most obvious observation is that all algorithms only work with a subset of the $\langle shutter, gain \rangle$ pairs, for a specific illumination. The general pattern is that algorithms prefer faster shutter speed and/or smaller voltage gain when the scene is bright, and slower shutter speed and/or bigger voltage gain when the scene is dark. However, algorithms demonstrate different sensitivity to changes in shutter speed and voltage gain. The DPM accepts wider range of values in the shutter/gain space due to relative illumination robustness of the underlying HOG features, while the BoW, R-CNN and SPP-net work with narrower range of values.

As for the best-performing configurations, they vary among different algorithms and illuminations. Taking the 200lx illumination condition for example, the best-performing $\langle shutter, gain \rangle$ pairs are (7, 3) for the DPM, (8, 4) for the BoW, (5, 5) for the R-CNN, (5, 5), (6, 6) and more for the SPP-net.



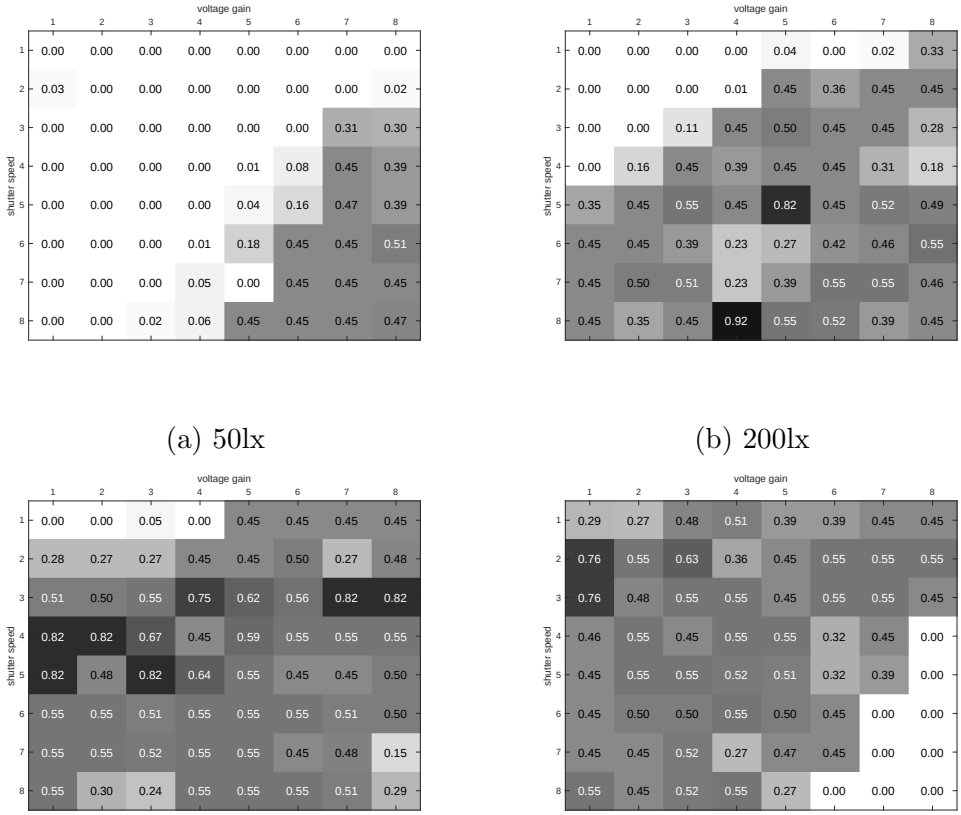
(a) 50lx

(b) 200lx

(c) 800lx

(d) 3200lx

Figure 3.3: The performance of DPM with respect to various illumination conditions. For each matrix, the shutter speed increases from top to bottom, and the voltage gain increases from left to right. The floating numbers are average precisions.



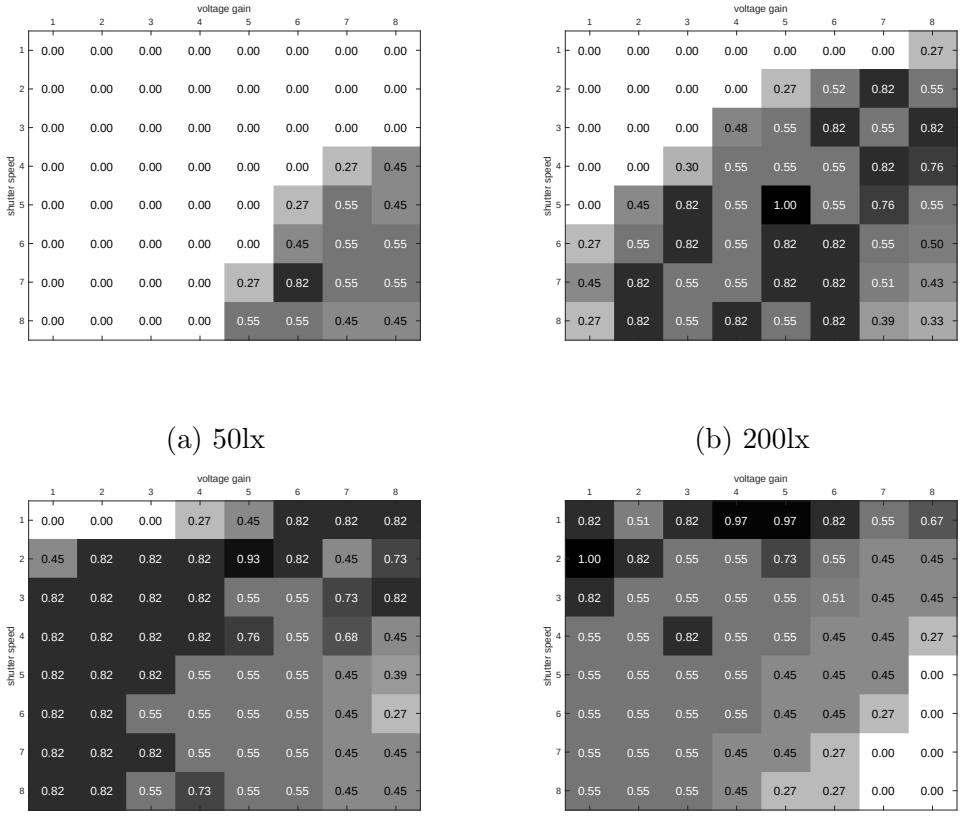
(a) 50lx

(b) 200lx

(c) 800lx

(d) 3200lx

Figure 3.4: The performance of BoW with respect to various illumination conditions. For each matrix, the shutter speed increases from top to bottom, and the voltage gain increases from left to right. The floating numbers are average precisions.



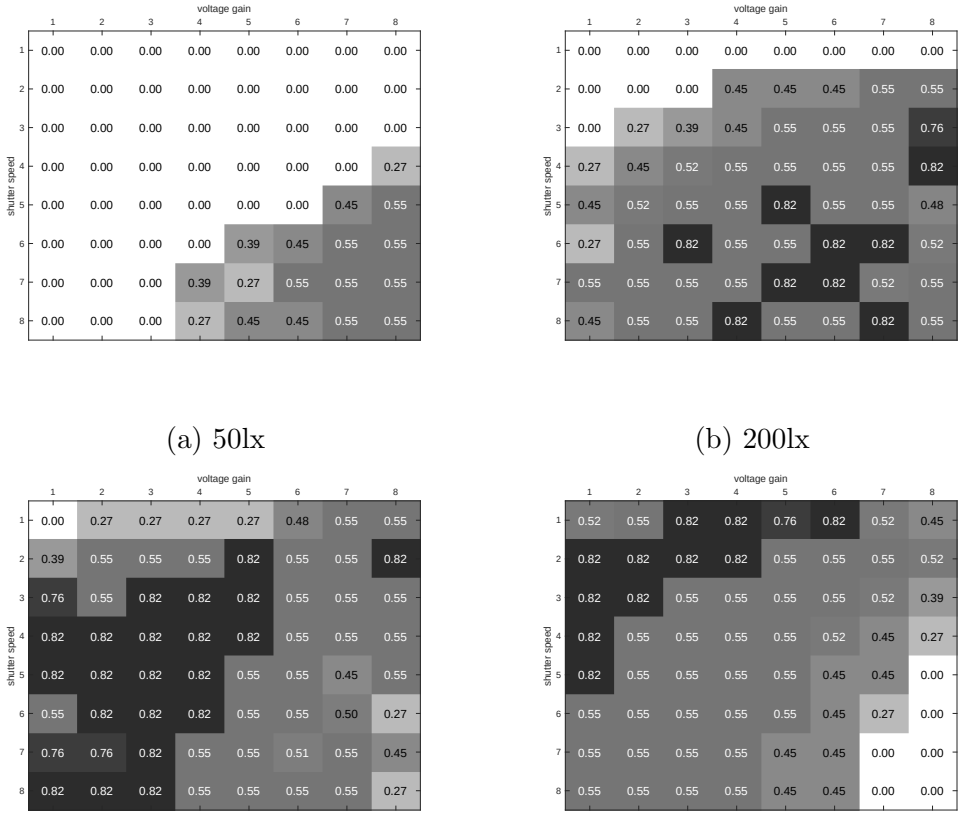
(a) 50lx

(b) 200lx

(c) 800lx

(d) 3200lx

Figure 3.5: The performance of R-CNN with respect to various illumination conditions. For each matrix, the shutter speed increases from top to bottom, and the voltage gain increases from left to right.



(a) 50lx

(b) 200lx

(c) 800lx

(d) 3200lx

Figure 3.6: The performance of SPP-net with respect to various illumination conditions. For each matrix, the shutter speed increases from top to bottom, and the voltage gain increases from left to right.

3.3.2 Independent Analysis

In this section, we independently analyze the performance of object detection algorithms with respect to three variables (illumination, shutter speed and voltage

gain). Results are represented by mean average precision (mAP) [75], which is the mean of a series of AP.

Illumination

Figure 3.7 summarizes the performance of the evaluated algorithms on various illumination conditions. The common trend is that the performance goes up, reaches the peak and falls as the ambient illumination goes from low to high. One possible reason is that all these algorithms are trained and tested on office datasets where the distribution of illumination is biased. It is also noteworthy that, for low illumination conditions, the DPM algorithm outperforms the other algorithms significantly due to relative illumination robustness of the underlying HOG features.

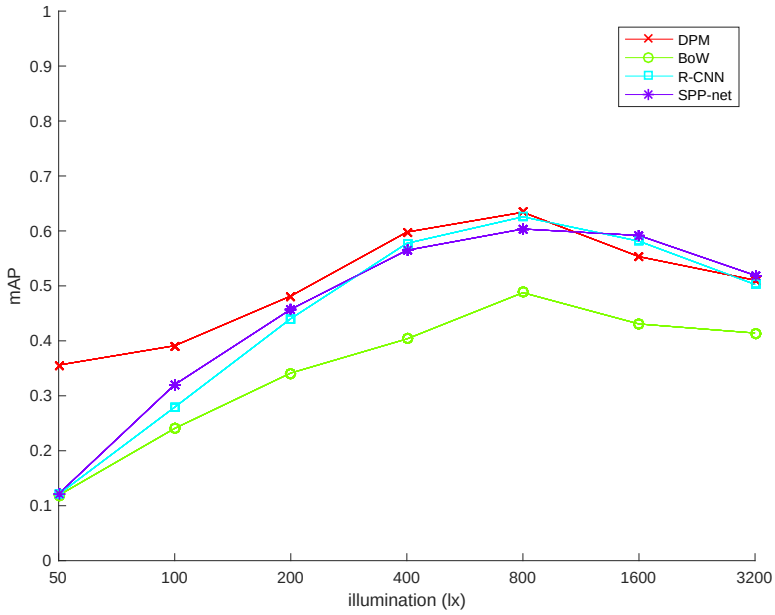


Figure 3.7: The mAP of four object detection algorithms with respect to various illumination conditions.

Shutter Speed

Figure 3.8 summarizes the performance of the evaluated algorithms on images taken with different shutter speeds. A larger shutter speed can improve the overall intensity of an image, from which all algorithms benefit. However, this effect is more obvious when the shutter speed is relatively small.

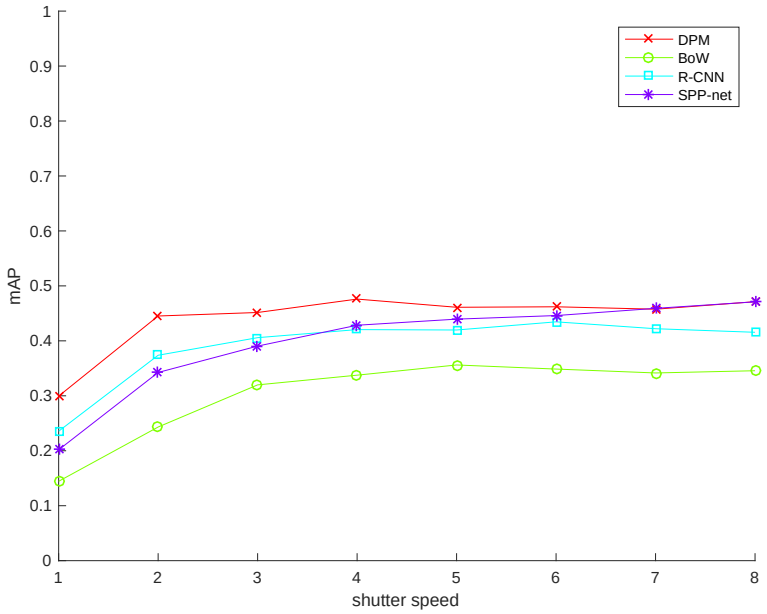


Figure 3.8: The mAP of four object detection algorithms with respect to various shutter speeds.

Voltage Gain

Figure 3.9 summarizes the performance of the evaluated algorithms on images taken with different voltage gains. As discussed in Section 1.1.2, the voltage gain increases the intensity but may also magnify the sensor noise. For DPM, constant performance loss was observed as the voltage gain increases, because the gradient computation becomes unreliable as the introduced noise increases. On the other hand, BoW, R-CNN and SPP-net benefit from increasing voltage gain when the

values are small.

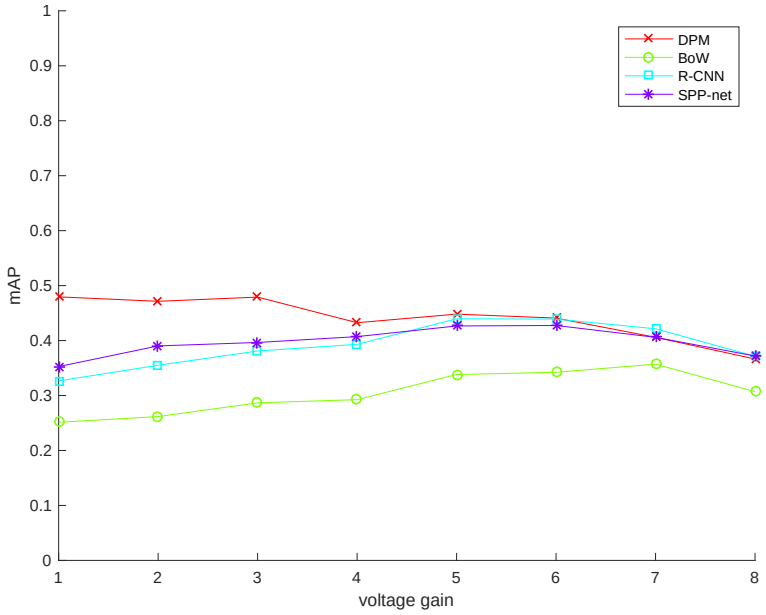


Figure 3.9: The mAP of four object detection algorithms with respect to various voltage gains.

Discussion

It was observed that the performance of object detection algorithms was highly dependent on the variables, i.e. illumination, shutter speed and voltage gain. These performance transients could be due to non-uniform sample representation at the given camera parameters in the original training set, and thus our method is able to uncover statistical irregularities in the training ensemble.

3.3.3 Auto-exposure

As discussed in Section 1.1.2, most modern digital cameras are equipped with a built-in auto-exposure (auto-shutter and/or auto-gain) algorithm. When creating our dataset, we also took a picture of each scene with the camera set in auto-exposure mode and recorded corresponding shutter/gain values. In this section, we present an empirical analysis of these recorded parameters.

Figure 3.10 shows the shutter speed and voltage gain values chosen by the auto-exposure algorithm of the Flea3 camera. For each illumination, the sensor parameters were set differently, and all recorded values are plotted in this figure. When the illumination is low, the values used are closer to each other. However, when the illumination is high, for example 1600px, there is more uncertainty between whether shutter speed or voltage gain should be reduced. The auto-exposure method chooses slower shutter speed and smaller voltage gain in some cases, while in other cases, faster shutter speed and larger voltage gain.

The $\langle shutter, gain \rangle$ pairs suggested by the auto-exposure algorithm are not the best-performing ones in the performance tables of object detection algorithms. Taking the 200lx illumination for example, the proposed (7, 8) and (7, 7) do not yield good results. Figure 3.3 and 3.4 illustrate the performance tables of the DPM on the images taken with auto-exposure.

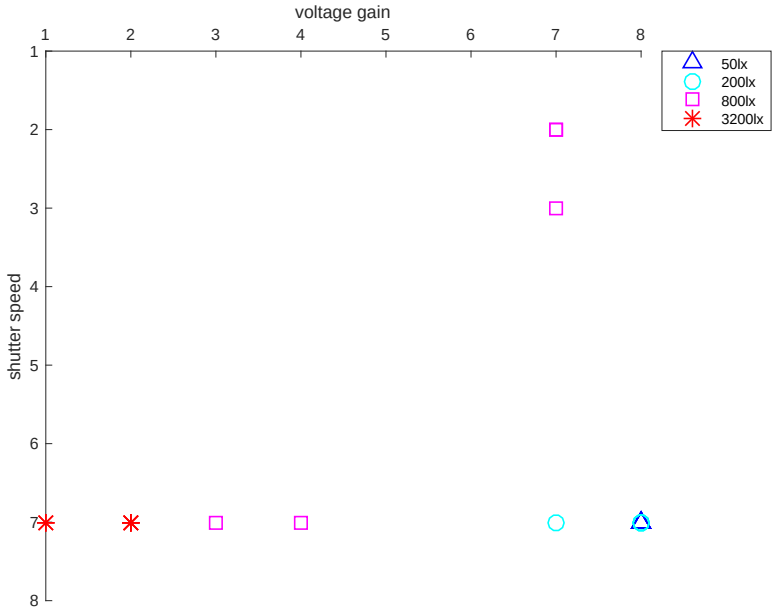


Figure 3.10: The $\langle shutter, gain \rangle$ pairs set by auto-exposure for various light conditions. (Original values have been mapped into discrete integers following the same procedures described in Section 3.1.2.)

Table 3.3: The images acquired with auto-exposure under various illumination conditions. The top 3 outputs of the DPM algorithm are overlaid on the input images.

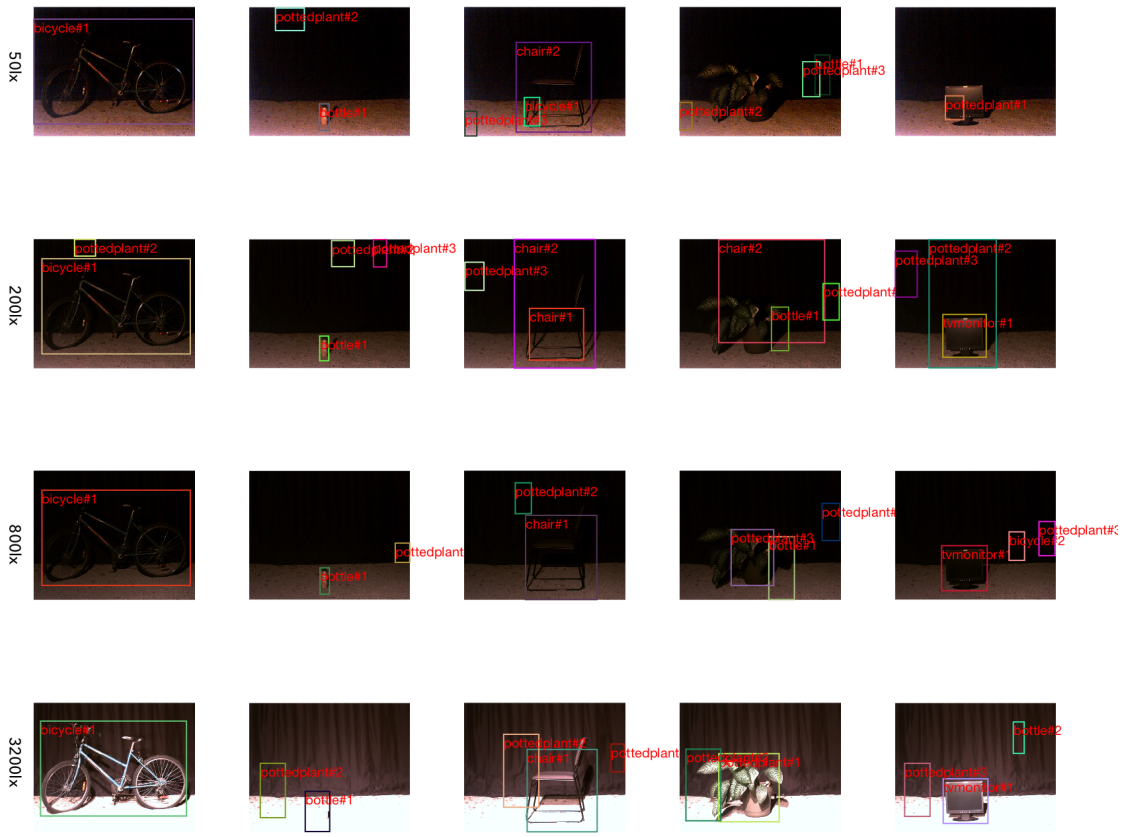


Table 3.4: The images acquired with best-performing camera parameters, based on Table 3.3, under various illumination conditions. The top 3 outputs of the DPM algorithm are overlaid on the input images.

3.4 Summary

In this chapter, we quantitatively evaluated the performance of four object detection algorithms with respect to their illumination and camera sensor bias. It was observed that the values of shutter speed and voltage gain parameters need to be chosen properly for the algorithms to work. Also, a generic (without task-directed knowledge) camera parameters controlling method is insufficient to yield reliable results for all the algorithms. Instead, the setting of shutter speed and voltage gain need to be dependent on the ambient illumination and current vision algorithm, which agrees with Andreopoulos's conclusions [11].

Chapter 4

Illumination Preprocessing

As discussed in Section 2.3, one of the common approaches to illumination compensation is image preprocessing. In this chapter, we evaluate how this approach could possibly improve the performance of an object detection algorithm. Specifically, the preprocessing here is the Laplacian-of-Gaussian (LoG), which has been previously used in [76, 77, 78, 79].

4.1 The Laplacian-of-Gaussian

The Laplacian is a 2-D isotropic measure of the second spatial derivative of an image, which filters regions with rapid intensity changes. Given an image $I(x, y)$,

the Laplacian $L(x, y)$ is given by

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2},$$

which can be computed using a convolution filter. Given that the Laplacian is based on a second derivative measurement on the image, it is very sensitive to noise. Therefore, an image is often Guassian smoothed, which can also be implemented by convolution, before applying a Laplacian filter. This step reduces high frequency noise components before the differentiation step. The Laplacian-of-Gaussian has the form:

$$LoG(x, y) = -\frac{1}{\pi\sigma^4}\Big[1 - \frac{x^2 + y^2}{2\sigma^2}\Big]e^{-\frac{x^2+y^2}{2\sigma^2}}.$$

Since convolution operations are associative, the LoG can be computed by convolution using a single kernel. Figure 4.1 demonstrates the LoG function (for a Guassian $\sigma = 1$).

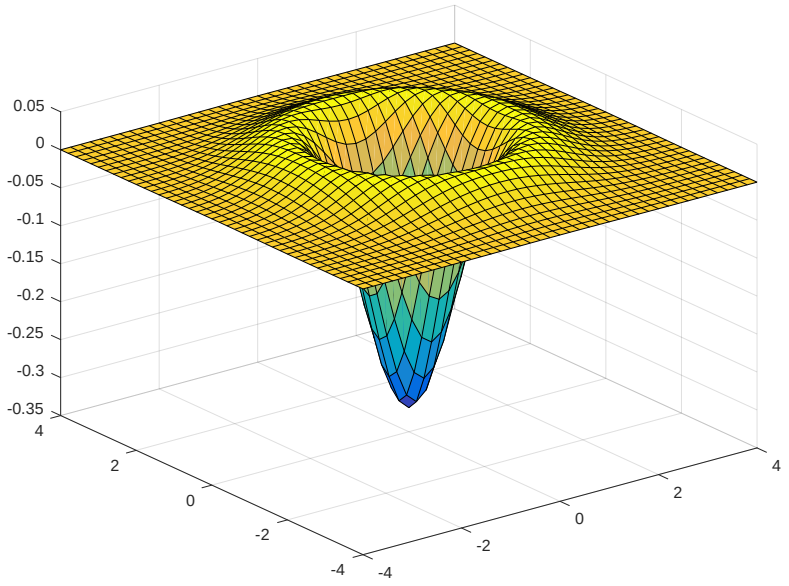


Figure 4.1: The 2-D Laplacian-of-Gaussian function (with Gaussian $\sigma = 1$).

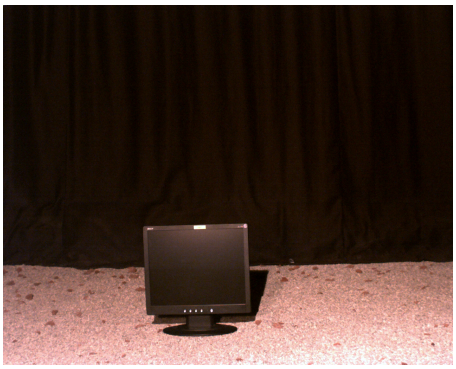
4.2 Experimental Setup

In this section, we describe how the LoG is used to enhance an image for illumination compensation and how the four object detection algorithms are evaluated on the preprocessed images.

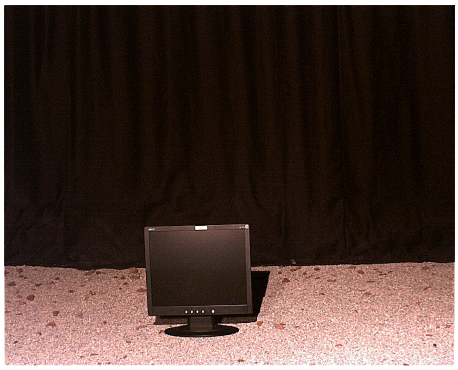
First of all, the LoG filter has a parameter σ to configure, which controls to what extent the image should be smoothed before the Laplacian step. To find a proper σ, we preprocess our dataset using three different σ (i.e. $\sigma = 0.5, 1, 2, 4$). Once the σ value is chosen, we run the LoG filter on the original image. Then,

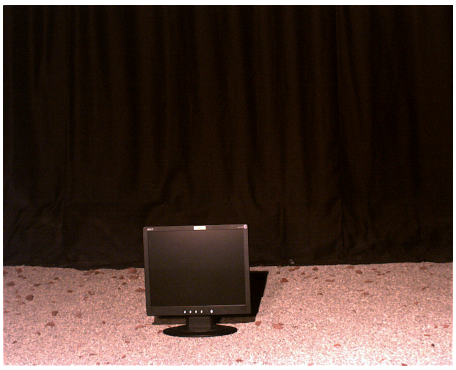
the filtered image and the original image are combined via pixel subtraction (if the value is within [0, 255], it is rounded to the nearest integer, if less than 0 it is set to 0, and if greater than 255 it is set to 255, as in [80]).The filtered image is not scaled before combining. Examples are shown in Figure 4.2. Notably, running object detection algorithms directly on the output of the LoG filter was not feasible in the scope of this work, as the algorithms are trained on original images or hand-crafted representations.

Once the dataset is preprocessed, we run the four algorithms (DPM, BoW, R-CNN, SPP-net) on the preprocessed images. The algorithms are set up using the same configuration described in Section 3.2.1. The outputs of algorithms are evaluated using the same procedures described in Section 3.2.2.

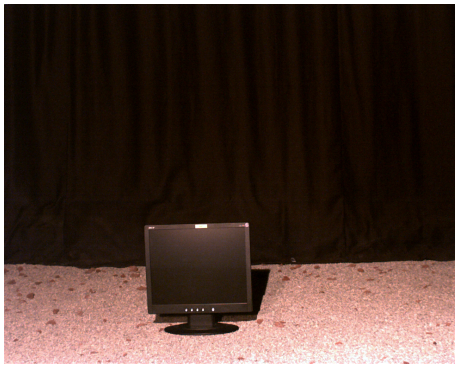


(a)





(b) (c)



(d) (e)

Figure 4.2: The images before and after the LoG preprocessing. (a) is the original image; (b)-(e) are the images after processing using $\sigma = 0.5, 1, 2, 4$ respectively.

4.3 Results and Discussion

The LoG preprocessing makes edges sharper and increases the contrast. However, this process also magnifies the effect of noise. Table 4.1 summarizes the performance of object detection algorithms on the LoG enhanced dataset. For the BoW, R-CNN and SPP-net, only minor improvements were observed after the LoG preprocessing when the σ is set properly. From this point, we conclude that the LoG preprocessing is helpful for illumination compensation but the improvements are not sufficient.

	Original dataset	LoG enhanced dataset			
		$\sigma = 0.5$	$\sigma = 1$	$\sigma = 2$	$\sigma = 4$
DPM	0.50	0.45	0.49	**0.51**	0.50
BoW	0.35	**0.36**	0.35	0.35	0.35
R-CNN	0.45	0.44	**0.47**	0.45	0.44
SPP-net	0.45	**0.47**	**0.47**	0.46	0.46

Table 4.1: The mAP of four object detection algorithms on the original dataset and the LoG enhanced dataset.

Chapter 5

Active Control of Camera Parameters and Algorithm Selection

5.1 Overview

From the quantitative analysis in Chapter 3, it is observed that the camera's intrinsic parameters have a significant impact on the performance of the object detection algorithms, and the optimal shutter speed and voltage gain configurations are algorithm and ambient illumination-specific. In this chapter, we propose a novel active control of camera parameters method based on the experimental benchmark

statistics.

First, we discuss the case where there is only one single object detection algorithm. We describe how this active control of camera parameters method could be used to select the optimal camera parameters in Section 5.2. Then, we describe the proposed algorithm selection extension, which automatically selects the optimal $\langle algorithm, shutter, gain \rangle$ combination based on ambient illumination in Section 5.3.

5.2 Active Control of Camera Parameters

In this section, we describe the motivation of our active control of camera parameters system, discuss the possible challenges and present the proposed implementation.

5.2.1 Motivation

The motivation of this active control of camera parameters approach is derived mainly from the analysis of the behaviors of object detection algorithms. It has been observed that vision algorithms behave differently with respect to variant illumination, shutter speed and voltage gain, as described in Chapter 3. The goal in this work is to find a way to systematically encode these behaviors and utilize them, to improve the stability and robustness of a vision guided robotic system.

An intuitive way would be benchmarking each vision algorithm, with respect to all possible $\langle shutter, gain \rangle$ pairs, for all illumination conditions. In this proposed system, the robot first measures the ambient illumination, then it looks up the corresponding performance table, and finally it selects the best-performing parameters to control the camera before taking a picture. This whole process is illustrated in Figure 1.2 and the lookup step is depicted in Figure 5.1.

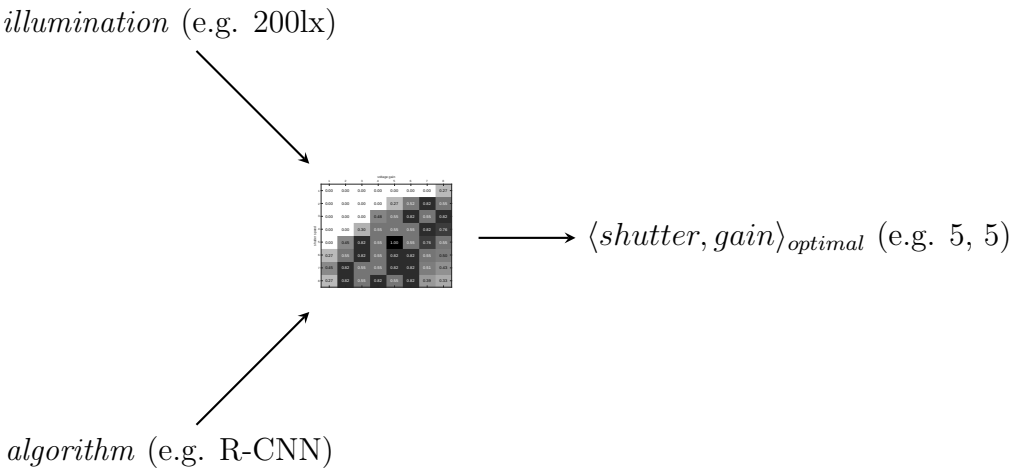


Figure 5.1: Demonstration of the active control of camera parameters.

5.2.2 Challenges

Despite the simplicity of the idea, there are also challenging problems. The first one is the reliability of the noisy performance tables, and the second one is that there are multiple optimal choices.

Figure 5.2 demonstrates the original performance table of DPM on images taken

with various camera configurations, under the illumination 800lx. In this case, the optimal $\langle shutter, gain \rangle$ pairs are $(2,3)$, $(2,4)$, $(3,1)$, $(4,1)$, $(4,2)$, $(4,3)$, $(4,8)$, $(5,1)$, $(5,5)$, $(5,7)$, $(7,1)$, $(8,1)$ and $(8,6)$, which all yield the best result 0.82. In this situation, it is unclear which one should be selected. By visual inspection, we can see that the majority of optimal choices are in the top-left quarter of the performance table and only a few outliers go beyond that area.

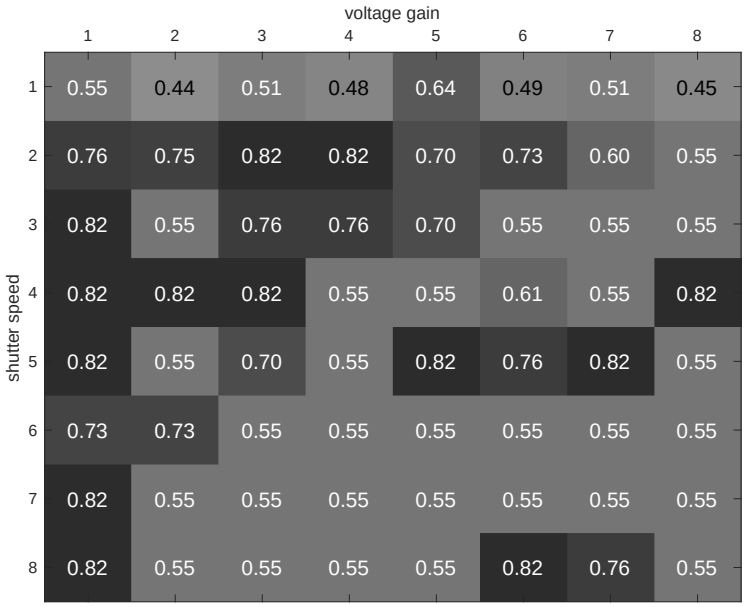


Figure 5.2: The performance table of DPM for illumination 800lx.

A practical active control of camera parameters system needs to balance between each local individual measurement and the global pattern, and minimize the possibility of multiple-choice situations.

5.2.3 Implementation

In this section, we describe our implementation of the active control of camera parameters method. The idea is to smooth the performance tables.

There are four major components in this system: (1) Create an image dataset of objects, by sampling the illumination, shutter speed, and voltage gain spaces; (2) Benchmark all available vision algorithms on the dataset and compute the performance table (see Figure 5.2 for example); (3) Smooth the performance tables using a Gaussian filter; (4) Select the optimal camera parameters based on the ambient illumination measured by light sensors.

The reason for smoothing is to remove outliers and reduce the possibility of multiple-maxima. The Guassian filter is used, due to its simplicity to trade off between each individual value and the local averages via the σ parameter. In our implementation, the kernel size is 3 x 3 and the σ value is set to 0.5, 1 or 2. For the values at boundaries, there isn't enough data to do a full smoothing operation. In such cases, we crop the Gaussian filter accordingly (zero-padding could be an alternative for the border effects).

For the purpose of this thesis, the values of ambient illumination, shutter speed and voltage gain are discontinuous. However, the actual readings of light sensor and the internal parameters of camera are continuous. To make the proposed system

accept continuous illumination and output continuous camera parameters, linear interpolation is applied to both the input and output.

5.3 Algorithm Selection

With the increasing availability of object detection algorithms, research on how to select an algorithm becomes more important. Previous efforts have been found in [81] and [23]. In this section, we present a new way of selecting algorithms based on their performance for various illumination conditions. This work extends the active control of camera parameters system proposed in the previous section.

The proposed extension is illustrated in Figure 5.3. Given an ambient illumination, it looks up the performance table for each of the available algorithms. An optimal $\langle shutter, gain \rangle$ pair is then selected following the procedures as described in 5.2.3 (the only difference is that the Gaussian smoothing operation is over all the three dimensions: illumination, shutter speed and voltage gain). Also, a confidence score (the smoothed AP) is assigned to each optimal configuration. After that, the confidence scores are compared, and the $\langle algorithm, shutter, gain \rangle$ combination with the highest score, is selected. See Appendix B for a Matlab implementation.

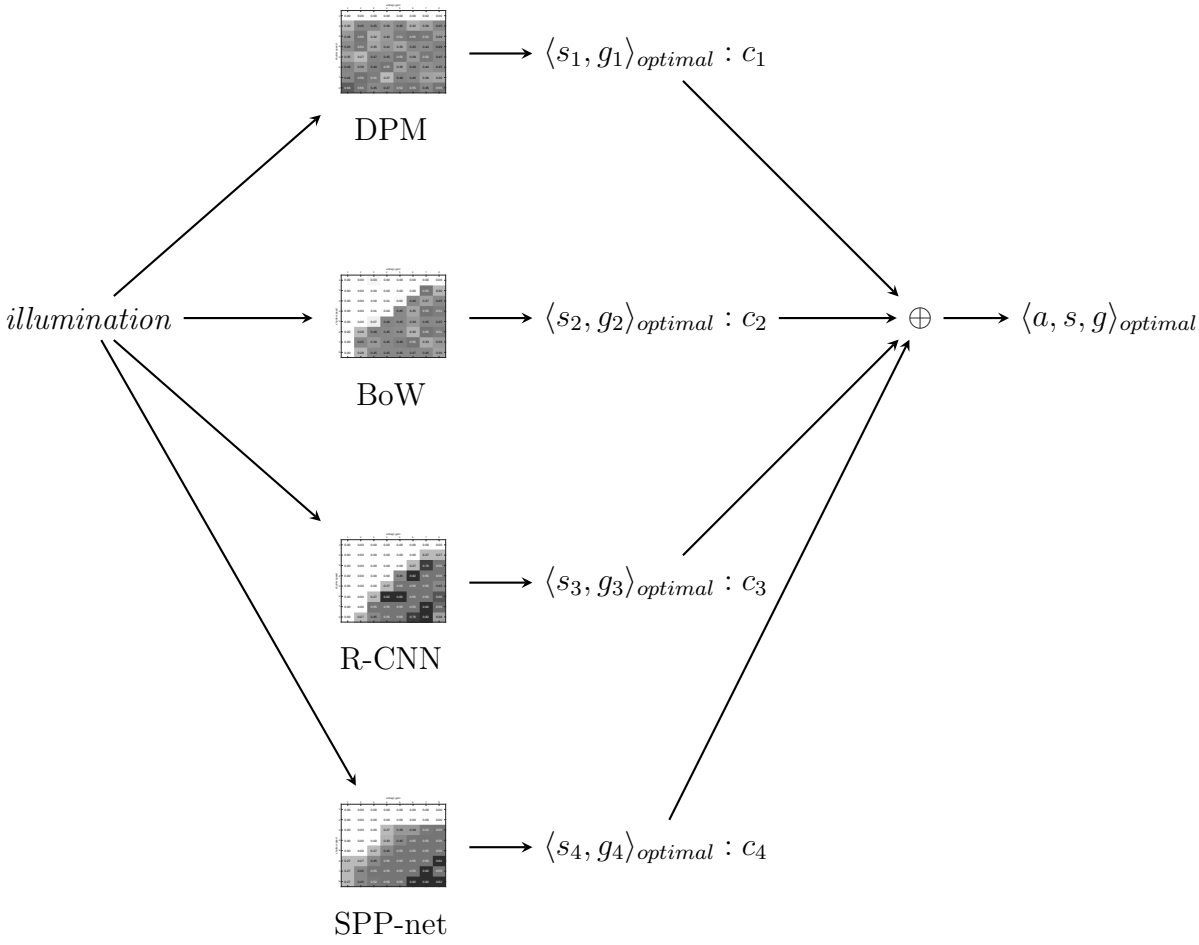


Figure 5.3: Demonstration of the algorithm selection extension.

5.4 Summary

In this chapter, we proposed the active control of camera parameters method based on experimental benchmark results and presented its algorithm selection extension to select the optimal $\langle algorithm, shutter, gain \rangle$ combinations when there are mul-

tiple vision algorithms available. We also discussed the underlying challenges and described how data should be processed properly (i.e. the Gaussian smoothing) to work around these issues. The empirical evaluation of the proposed system is to be presented in Chapter 6.

Chapter 6

Empirical Evaluation

6.1 Experimental Overview

To evaluate how the proposed active control of camera parameters method, and algorithm selection extension work, two experiments are conducted. For the active control experiment, the setup is described in Section 6.2.1 and the results are presented in Section 6.2.2. For the algorithm selection experiment, the setup is described in Section 6.3.1 and the results are presented in Section 6.3.2.

6.2 Experiment I: Active Control of Camera Parameters

The active control of camera parameters experiment is designed to be a proof-of-principle task which would demonstrate what performance gain could be achieved

by active control of shutter speed and voltage gain, for object detection algorithms. Conventionally, these two parameters are set by cameras built-in auto-exposure algorithms [82, 83], which set camera exposure by evaluating the mean brightness of an image.

6.2.1 Experimental Setup

This experiment is conducted on the dataset introduced in Section 3.1. We repeatedly split this dataset into two groups, one is used for the active control of camera parameters system to build the performance tables, and the other is used for evaluation (average values are presented with no explicit clarification). The evaluation procedures are as follows:

1. For each object and for each illumination, run the proposed method described in Section 5.2 on the *training* set and get the proposed $\langle shutter, gain \rangle$ pair.

2. Run each object detection algorithm on the image that corresponds to the proposed camera parameters, and on the image that are taken with auto-exposure.

3. Evaluate and compare the results of object detection algorithms on the two images (A predicted bounding box is considered correct if it overlaps no less than 50% with the ground-truth bounding box, otherwise false).

The camera was a Point Grey Flea3 camera (see specifications at Appendix A), which had an auto-exposure algorithm that is based on the average image intensity of the region-of-interest (ROI). We used the default optimal brightness level and ROI. Also, the auto shutter speed and voltage gain ranges were both kept at default values.

It's noteworthy that the shutter speed and voltage gain values set by the auto-exposure algorithm are continuous, while our dataset and proposed framework use discrete values. To make them comparable, both the shutter speed and voltage gain ranges were uniformly mapped into eight discrete values (from 1 to 8) following the same schema described in Section 3.1.2. Without explicit clarification, these two camera parameters are represented in relative values throughout this chapter.

6.2.2 Results and Discussions

Overview

Table 6.1 and Figure 6.1 summarize the results of each object detection algorithm with auto-exposure and active control. The active control of camera parameters method outperforms the conventional auto-exposure algorithm for three (DPM, Bow and R-CNN) out of four object detection algorithms.

Also, the results are dependent on the parameter σ of the Gaussian smoothing

operator, as noises in the performance tables of different algorithms vary. No single σ value, that constantly outperforms the others, has been found. However, $\sigma = 1$ gives an overall decent results in our experiment.

	auto-exposure	active control		
		$\sigma = 0.5$	$\sigma = 1$	$\sigma = 2$
DPM	0.48	0.51	0.57	**0.60**
BoW	0.38	0.49	**0.51**	0.49
R-CNN	**0.61**	0.56	0.60	0.60
SPP-net	0.63	0.66	**0.69**	0.66

Table 6.1: The mAP of four object detection algorithms with auto-exposure and active control. The best performance is highlighted for each algorithm.

Proposed Camera Parameters

In this section, we discuss the proposed $\langle shutter, gain \rangle$ pairs by the active control of camera parameter method. Figure 6.2 summarizes the results.

The global patterns of the selected camera parameters by the active control method are similar to values set by auto-exposure (See Figure 3.10). For low illumination conditions, the proposed method selects slower shutter speed and/or larger voltage gain, and faster shutter speed and/or smaller voltage gain for high

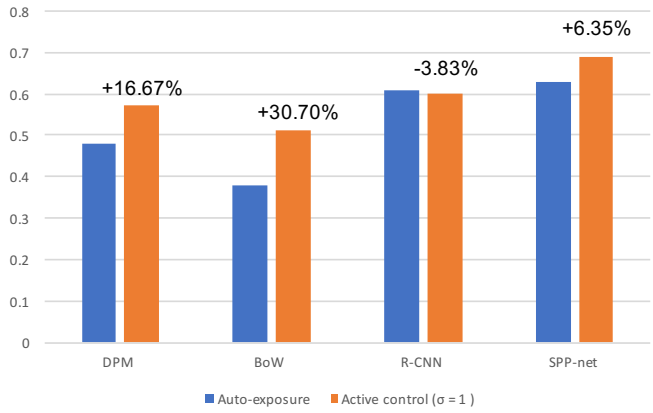
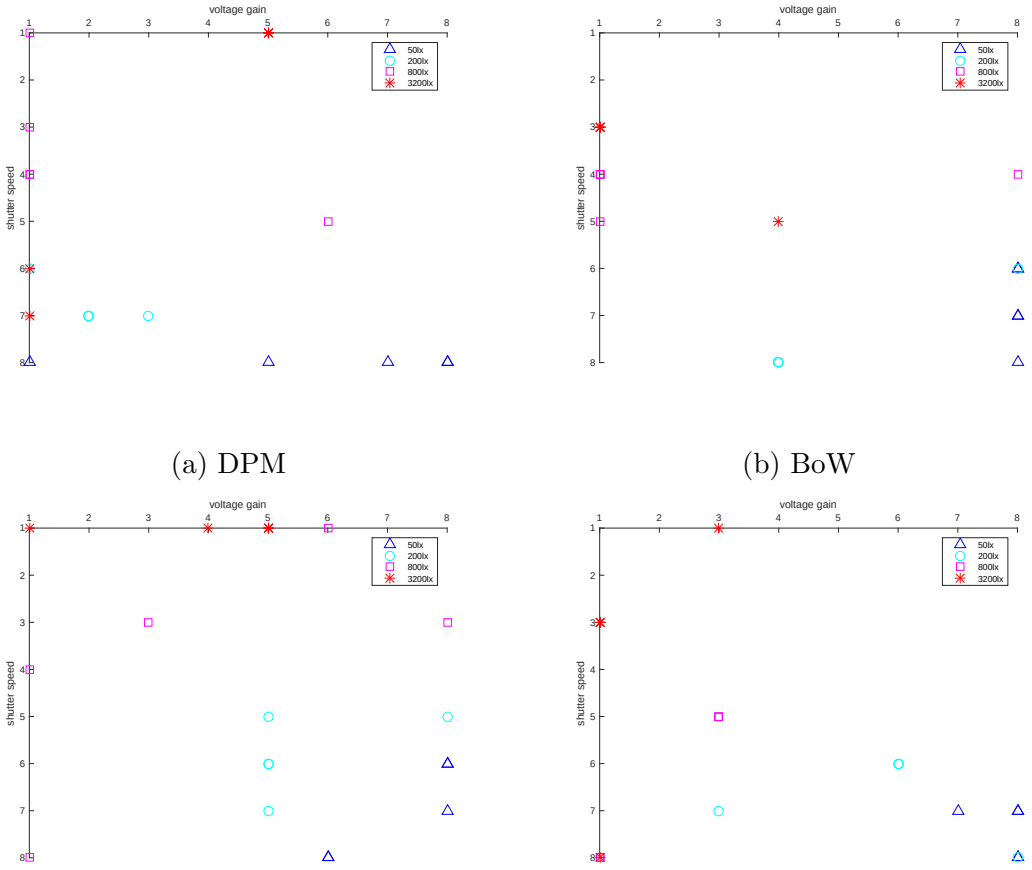


Figure 6.1: The comparison of auto-exposure and active control by the performance of four object detection algorithms.

illumination conditions.

Also, the selected camera parameters by the active control method vary among the object detection algorithms. Taking the 50lx illumination condition for example, the selected $\langle shutter, gain \rangle$ pairs for the DPM are $(1, 8)$, $(5, 8)$, $(7, 8)$ and $(8, 8)$, while the selected $\langle shutter, gain \rangle$ pairs for the BoW are $(8, 6)$, $(8, 7)$ and $(8, 8)$.



(a) DPM

(b) BoW

(c) R-CNN

(d) SPP-net

Figure 6.2: The proposed $\langle shutter, gain \rangle$ pairs by the active control method for four object detection algorithms.

6.3 Experiment II: Algorithm Selection

The algorithm selection experiment is to evaluate the efficiency of the proposed algorithm selection extension. In Experiment I, we have observed great improve-

ments over the conventional auto-exposure for three out of four object detection algorithms. In this experiment, we continue to investigate if the results could be further improved by dynamically selecting an algorithm instead of using a static one.

6.3.1 Experimental Setup

This experiment is also conducted on the dataset introduced in Section 3.1. This dataset is split into two sets, *training* and *testing*. The *training* set is used for building the performance tables for each vision algorithm, and the *testing* set is used for evaluation. This process is repeated by using different combination of *training* and *testing* sets, i.e. 5-fold cross-validation. The procedures are as follows:

1. Run the proposed system described in Section 5.3 on the *training* set, to compute performance tables;

2. For each object and for each illumination in the *testing* set, get the selected $\langle algorithm, shutter, gain \rangle$ tuple. Run the selected algorithm on the image that corresponds to the selected camera parameters;

3. Run a static algorithm on the image taken with the shutter speed and voltage gain suggested by the active control of camera parameters, as described in Experiment I.

4. Evaluate and compare the results (A predicted bounding box is considered correct if it overlaps no less than 50% with the ground-truth bounding box, otherwise false).

6.3.2 Results and Discussions

The results of this experiment is summarized in Table 6.2. The results of the proposed algorithm selection extension are decent, compared with the other approaches. SPP-net with active control of camera parameters yields the best results, with a mAP of 0.69. One possible reason is that the situation where different algorithms specialize in different light conditions is not found in this experiment.

algorithm	mAP
DPM with active control	0.57
BoW with active control	0.51
R-CNN with active control	0.60
SPP-net with active control	0.69
Algorithm selection ($\sigma = 1$)	0.60

Table 6.2: The results of the algorithm selection extension.

Also, the proposed algorithm selection extension demonstrates the capability of selecting algorithm properly based on the ambient illumination. Figure 6.3 sum-

marizes the number of times that each algorithm has been selected. The most frequently selected algorithms are R-CNN and SPP-net. DPM is also selected for low illumination conditions.

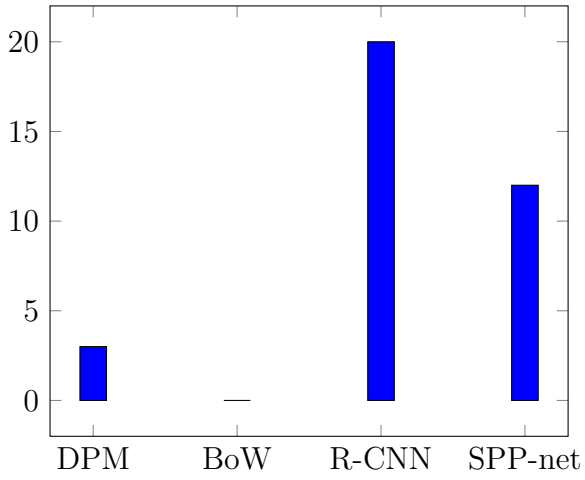


Figure 6.3: The number of times each algorithm has been selected ($\sigma = 1$).

Chapter 7

Conclusions

In this chapter, we conclude this thesis by summarizing the work and discussing directions for future study.

7.1 Summary

This thesis focuses on a novel framework for controlling camera parameters and selecting vision algorithms, to improve the robustness and adaptivity of vision guided robotic systems, under varying light conditions. Four algorithms were reviewed in Chapter 2 and quantitatively analyzed in Chapter 3. Based on the experimental results, a novel active control of camera parameters method was proposed in Section 5.2, and an algorithm selection extension was presented in Section 5.3.

Following the research of Andreopoulos et al. [11] on comparing various interest

point and saliency algorithms, we quantitatively analyzed the performance of four object detection algorithms, DPM, BoW, R-CNN and SPP-net, with respect to variant illumination, shutter speed and voltage gain configurations. A new dataset was also introduced for benchmarking. We found that the object detection algorithms demonstrated different sensitivity to the camera parameters, which agrees with Andreopoulos's results. In order to make them work properly, an algorithm and illumination-specific strategy for setting camera parameters is required.

Based on the observations in the quantitative analysis, a novel active control of camera parameters method was proposed. This method analyzes the characteristics of an object detection algorithm with respect to illumination, shutter speed and voltage gain, and selects the best-performing combination of camera parameters for a given ambient illumination. In the empirical evaluation, this proposed method has significantly outperformed the conventional auto-exposure approach, for three out of four algorithms.

Finally, an algorithm selection extension was proposed, which selects the optimal $\langle algorithm, shutter, gain \rangle$ tuple for various illumination conditions. This proposed extension has demonstrated the capability of selecting a proper algorithm based on the ambient illumination.

7.2 Future Work

Although the proposed methods are effective in improving the robustness of vision algorithms to illumination variations, they can be further improved in following directions.

The first direction is to investigate object dependencies. In this work, object dependencies have not been incorporated into the proposed framework. However, the performance of object detection algorithms could be dependent on specific objects. To verify whether this is the case, further investigation is required.

The second direction is to investigate other vision algorithms and camera parameters, using the same framework in this work. It can be summarized as: (1) create a dataset by sampling the camera parameter spaces; (2) evaluate the vision algorithms of interest on this dataset, and build performance tables; (3) optimize camera parameters using the performance tables.

The third direction is to investigate the effects of linear interpolation. Currently, the ambient illumination, shutter speed and voltage gain are sampled at discrete points. Linear interpolation is used to make the system accept continuous illumination inputs. It is worthy to investigate whether linear interpolation affects the performance of the proposed framework.

Bibliography

[1] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[4] R. Bajcsy, "Active perception vs. passive perception," in *IEEE Workshop on Computer Vision Representation and Control*, (Bellaire, Michigan), 1985.

[5] J. Aloimonos, I. Weiss, and A. Bandyopadhyay, "Active vision," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 333–356, 1988.

[6] J. K. Tsotsos, "On the relative complexity of active vs. passive visual search," *International Journal of Computer Vision*, vol. 7, no. 2, pp. 127–141, 1992.

[7] S. Bileschi, "CBCL streetscenes challenge framework (2007)." http://cbcl.mit.edu/software-datasets/streetscenes/.

[8] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[9] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," pp. 740–755, Springer, 2014.

[11] A. Andreopoulos and J. K. Tsotsos, "On sensor bias in experimental methods for comparing interest-point, saliency, and recognition algorithms," *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 1, pp. 110–126, 2012.

[12] I. Shim, J.-Y. Lee, and I. S. Kweon, "Auto-adjusting camera exposure for outdoor robotics using gradient information," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1011–1017, IEEE, 2014.

[13] S. J. Dickinson, H. I. Christensen, J. K. Tsotsos, and G. Olofsson, "Active object recognition integrating attention and viewpoint control," *Computer Vision and Image Understanding*, vol. 67, no. 3, pp. 239–260, 1997.

[14] H. Lu, H. Zhang, S. Yang, and Z. Zheng, "Camera parameters auto-adjusting technique for robust robot vision," in *IEEE International Conference on Robotics and Automation*, pp. 1518–1523, IEEE, 2010.

[15] B. Browatzki, V. Tikhanoff, G. Metta, H. H. Bülthoff, and C. Wallraven, "Active object recognition on a humanoid robot," in *IEEE Conference on Robotics and Automation*, pp. 2021–2028, IEEE, 2012.

[16] R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos, "Revisiting active perception," *arXiv preprint arXiv:1603.02729*, 2016.

[17] D. Litwiller, "CCD vs. CMOS," *Photonics Spectra*, vol. 35, no. 1, pp. 154–158, 2001.

[18] R. Ramanath, W. E. Snyder, Y. Yoo, and M. S. Drew, "Color image processing pipeline," *IEEE Signal Processing Magazine*, vol. 22, no. 1, pp. 34–43, 2005.

[19] G. E. Healey and R. Kondepudy, "Radiometric CCD camera calibration and noise estimation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 3, pp. 267–276, 1994.

[20] J. Valeton and D. van Norren, "Light adaptation of primate cones: an analysis based on extracellular data," *Vision Research*, vol. 23, no. 12, pp. 1539–1547, 1983.

[21] K. Purpura, E. Kaplan, and R. Shapley, "Background light and the contrast gain of primate p and m retinal ganglion cells," *Proceedings of the National Academy of Sciences*, vol. 85, no. 12, pp. 4534–4537, 1988.

[22] U. Knauer and U. Seiffert, "A comparison of late fusion methods for object detection," in *IEEE International Conference on Image Processing*, pp. 3297–3301, 2013.

[23] R. A. Bianchi, A. Ramisa, and R. L. De Mántaras, "Automatic selection of object recognition methods using reinforcement learning," in *Advances in Machine Learning I*, pp. 421–439, Springer, 2010.

[24] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point de-

tectors," *International journal of computer vision*, vol. 60, no. 1, pp. 63–86, 2004.

[25] T. Kadir and M. Brady, "Saliency, scale and image description," *International Journal of Computer Vision*, vol. 45, no. 2, pp. 83–105, 2001.

[26] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing*, vol. 22, no. 10, pp. 761–767, 2004.

[27] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*, pp. 404–417, Springer, 2006.

[28] L. Itti, C. Koch, E. Niebur, *et al.*, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 20, no. 11, pp. 1254–1259, 1998.

[29] N. D. Bruce and J. K. Tsotsos, "Saliency, attention, and visual search: An information theoretic approach," *Journal of vision*, vol. 9, no. 3, pp. 5–5, 2009.

[30] P. Perona, "Visual recognition circa 2008," in *Object categorization: computer and human vision perspectives* (S. J. Dickinson, A. Leonardis, B. Schiele, and T. M. J, eds.), pp. 55–68, Cambridge/New York: Cambridge University Press, 2009.

[31] M.-H. Yang, D. J. Kriegman, and N. Ahuja, "Detecting faces in images: A survey," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 1, pp. 34–58, 2002.

[32] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[33] A. Andreopoulos and J. K. Tsotsos, "50 years of object recognition: Directions forward," *Computer Vision and Image Understanding*, vol. 117, no. 8, pp. 827–891, 2013.

[34] S. Zafeiriou, C. Zhang, and Z. Zhang, "A survey on face detection in the wild: past, present and future," *Computer Vision and Image Understanding*, vol. 138, pp. 1–24, 2015.

[35] R. Verschae and J. Ruiz-del Solar, "Object detection: current and future directions," *Frontiers in Robotics and AI*, vol. 2, p. 29, 2015.

[36] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," *IEEE Transactions on Computers*, no. 1, pp. 67–92, 1973.

[37] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object

recognition," *International Journal of Computer Vision*, vol. 61, no. 1, pp. 55–79, 2005.

[38] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[39] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 1, pp. 886–893, IEEE, 2005.

[40] A. Bosch, X. Muñoz, and R. Martí, "Which is the best way to organize/classify images by content?," *Image and vision computing*, vol. 25, no. 6, pp. 778–791, 2007.

[41] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3304–3311, IEEE, 2010.

[42] K. Mikolajczyk, B. Leibe, and B. Schiele, "Local features for object class recognition," in *IEEE International Conference on Computer Vision*, vol. 2, pp. 1792–1799, IEEE, 2005.

[43] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: a sur-

vey," *Foundations and Trends in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177–280, 2008.

[44] E. Nowak, F. Jurie, and B. Triggs, "Sampling strategies for bag-of-features image classification," in *European Conference on Computer Vision*, pp. 490–503, Springer, 2006.

[45] D. G. Lowe, "Object recognition from local scale-invariant features," in *IEEE International Conference on Computer vision*, vol. 2, pp. 1150–1157, IEEE, 1999.

[46] Y. Ke and R. Sukthankar, "Pca-sift: A more distinctive representation for local image descriptors," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. II–506, IEEE, 2004.

[47] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *IEEE International Conference on Computer Vision*, pp. 1470–1477, IEEE, 2003.

[48] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8, IEEE, 2008.

[49] J. C. Van Gemert, J.-M. Geusebroek, C. J. Veenman, and A. W. Smeulders, "Kernel codebooks for scene categorization," in *European conference on computer vision*, pp. 696–709, Springer, 2008.

[50] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3360–3367, IEEE, 2010.

[51] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *European Conference on Computer Vision*, pp. 143–156, Springer, 2010.

[52] X. Zhou, K. Yu, T. Zhang, and T. S. Huang, "Image classification using super-vector coding of local image descriptors," in *European conference on computer vision*, pp. 141–154, Springer, 2010.

[53] K. Chatfield, V. S. Lempitsky, A. Vedaldi, and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods.," in *British Machine Vision Conference*, vol. 2, p. 8, 2011.

[54] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 2169–2178, IEEE, 2006.

[55] R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *International Joint Conference on Neural Networks*, pp. 593–605, IEEE, 1989.

[56] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 580–587, IEEE, 2014.

[57] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[58] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[59] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European Conference on Computer Vision*, pp. 818–833, Springer, 2014.

[60] Y. Adini, Y. Moses, and S. Ullman, "Face recognition: The problem of compensating for changes in illumination direction," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 19, no. 7, pp. 721–732, 1997.

[61] M. Osadchy and D. Keren, "Image detection under varying illumination and

pose," in *IEEE Conference on Computer Vision*, vol. 2, pp. 668–673, IEEE, 2001.

[62] M. Osadchy and D. Keren, "Efficient detection under varying illumination conditions and image plane rotations," *Computer Vision and Image Understanding*, vol. 93, no. 3, pp. 245–259, 2004.

[63] T. Tanaka, A. Shimada, D. Arita, and R.-i. Taniguchi, "Object detection under varying illumination based on adaptive background modeling considering spatial locality," in *Pacific-Rim Symposium on Image and Video Technology*, pp. 645–656, Springer, 2009.

[64] W. Maier, M. Eschey, and E. Steinbach, "Image-based object detection under varying illumination in environments with specular surfaces," in *IEEE International Conference on Image Processing*, pp. 1389–1392, IEEE, 2011.

[65] M. Linderoth, A. Robertsson, and R. Johansson, "Color-based detection robust to varying illumination spectrum," in *IEEE Workshop on Robot Vision*, pp. 120–125, IEEE, 2013.

[66] S.-D. Wei and S.-H. Lai, "Robust face recognition under lighting variations," in *International Conference on Pattern Recognition*, vol. 1, pp. 354–357, IEEE, 2004.

[67] E. G. Llano, H. M. Vazquez, J. Kittler, and K. Messer, "An illumination insensitive representation for face verification in the frequency domain," in *International Conference on Pattern Recognition*, vol. 1, pp. 215–218, IEEE, 2006.

[68] Y. Tang, R. Salakhutdinov, and G. Hinton, "Deep lambertian networks," in *International Conference on Machine Learning*, 2012.

[69] P. N. Belhumeur and D. J. Kriegman, "What is the set of images of an object under all possible illumination conditions?," *International Journal of Computer Vision*, vol. 28, no. 3, pp. 245–260, 1998.

[70] J. M. Tenenbaum, "Accommodation in computer vision.," tech. rep., DTIC Document, 1970.

[71] K. Brunnström, J.-O. Eklundh, and T. Uhlin, "Active fixation for scene exploration," *International Journal of Computer Vision*, vol. 17, no. 2, pp. 137–162, 1996.

[72] H. Han, S. Shan, X. Chen, and W. Gao, "A comparative study on illumination preprocessing in face recognition," *Pattern Recognition*, vol. 46, no. 6, pp. 1691–1699, 2013.

[73] R. J. Alitappeh, K. J. Saravi, and F. Mahmoudi, "A new illumination invariant

feature based on sift descriptor in color space," *Procedia Engineering*, vol. 41, pp. 305–311, 2012.

[74] A. Kanezaki, S. Inaba, Y. Ushiku, Y. Yamashita, H. Muraoka, Y. Kuniyoshi, and T. Harada, "Hard negative classes for multiple object detection," in *IEEE International Conference on Robotics and Automation*, pp. 3066–3073, IEEE, 2014.

[75] G. Salton and M. J. McGill, "Introduction to modern information retrieval," 1986.

[76] M. Sharif, S. Mohsin, M. Y. Javed, and M. A. Ali, "Single image face recognition using laplacian of gaussian and discrete cosine transforms.," *International Arab Journal of Information Technology*, vol. 9, no. 6, pp. 562–570, 2012.

[77] E. F. Badran, E. G. Mahmoud, and N. Hamdy, "An algorithm for detecting brain tumors in mri images," in *International Conference on Computer Engineering and Systems*, pp. 368–373, IEEE, 2010.

[78] C. Rathgeb and A. Uhl, "Secure iris recognition based on local intensity variations," in *International Conference Image Analysis and Recognition*, pp. 266–275, Springer, 2010.

[79] M. M. Rahman and S. Ishikawa, "Applying image pre-processing techniques for

appearance-based human posture recognition: an experimental analysis," in *Australasian Joint Conference on Artificial Intelligence*, pp. 152–159, Springer, 2004.

[80] F. Neycenssac, "Contrast enhancement using the laplacian-of-a-gaussian filter," *CVGIP: Graphical Models and Image Processing*, vol. 55, no. 6, pp. 447–463, 1993.

[81] M. Gabryel, M. Korytkowski, R. Scherer, and L. Rutkowski, "Object detection by simple fuzzy classifiers generated by boosting," in *Artificial Intelligence and Soft Computing*, pp. 540–547, Springer, 2013.

[82] B. K. Johnson, "Photographic exposure control system and method," Jan. 3 1984. US Patent 4,423,936.

[83] N. Sampat, S. Venkataraman, T. Yeh, and R. L. Kremens, "System implications of implementing auto-exposure on consumer digital cameras," in *Electronic Imaging'99*, pp. 100–107, International Society for Optics and Photonics, 1999.

Appendix A

Camera Specifications

In this appendix, the specifications of PointGrey Flea3 camera are presented in Table A.1 and the lens specifications are presented in Table A.2 . This camera comes with an API interface, which enables us to programmatically configure the internal parameters and transfer the perceived image to an computer instantly.

Resolution	1280 x 1024
Frame Rate	60 FPS
Megapixels	1.3 MP
Chroma	Color
Sensor Name	e2v EV76C560
Sensor Type	CMOS
Readout Method	Global shutter
Sensor Format	1/1.8”
Pixel Size	5.3 m
Lens Mount	C-mount
ADC	10-bit

Quantum Efficiency Blue (% at 470 nm)	47
Quantum Efficiency Green (% at 525 nm)	48
Quantum Efficiency Red (% at 640 nm)	41
Temporal Dark Noise (e-)	26.24
Saturation Capacity (e-)	5726
Dynamic Range (dB)	46.61
Gain Range	0 dB to 18 dB
Exposure Range	0.016 ms to 1 second
Trigger Modes	Standard, multi-shot
Partial Image Modes	Pixel binning, ROI
Image Processing	Gamma, lookup table, hue, saturation, and sharpness
Image Buffer	32 MB
User Sets	2 memory channels for custom camera settings
Flash Memory	1 MB non-volatile memory
Opto-isolated I/O Ports	1 input, 1 output
Non-isolated I/O Ports	2 bi-directional
Serial Port	1 (over non-isolated I/O)
Auxiliary Output	3.3 V, 150 mA maximum
Interface	USB 3.0
Power Requirements	5-24 V via GPIO or 5 V via USB 3.0
Power Consumption (Maximum)	<3 W
Dimensions	29 mm x 29 mm x 30 mm
Mass	41 g
Machine Vision Standard	USB3 Vision v1.0
Compliance	CE, FCC, KCC, RoHS
Temperature (Operating)	0 to 45C
Temperature (Storage)	-30 to 60C

Humidity (Operating)	20 to 80% (no condensation)
Humidity (Storage)	20 to 95% (no condensation)
Warranty	3 years

Table A.1: Specifications of the Flea3 Camera

Manufacturer Part Number	Fujinon HF12.5HA-1B
Focal Length	12.5mm
Optical Format	2/3”
Lens Mount	C Mount

Table A.2: Lens Specifications

Appendix B

Matlab Implementation

In this appendix, the Matlab implementation of the active control of camera parameters algorithm is presented.

```
function [alg, shutter, gain] = accp(ambient_illu)
%
% active control of camera parameters (with algorithm selection)
%
algorithms = {'DPM', 'BoW', 'R-CNN', 'SPP-net'};
illus = {'50', '100', '200', '400', '800', '1600', '3200'};

optm_shutter = zeros(length(algorithms), 1);
optm_gain = zeros(length(algorithms), 1);
optm_score = zeros(length(algorithms), 1);

kernel_size = 3;
sigma = 1.0;

for i = 1:length(algorithms)

    ap = zeros(8, 8, length(illus));

```

```
    for j = 1:length(illus)
        for s = 1:8
            for g = 1:8
                % compute the AP of ith algorithm on the images taken
                % with illumination j, shutter s and voltage gain g.
                ap(s, g, j) = get_ap(algorithms{i}, illus{j}, s, g);
            end
        end
    end

    % compute the optimal <shutter, gain> pair and associated score
    % for an algorithm.
    ap2 = smooth_3d(ap, kernel_size, sigma);
    [mx, idxes] = max(ap2(ambient_illu, :, :));
    [s, g] = ind2sub([8 8], idxes);
    optm_shutter(i) = s;
    optm_gain(i) = g;
    optm_score(i) = mx;
end

% return results
[~, idx] = max(optm_score);
alg = algorithms(idx);
shutter = optm_shutter(idx);
gain = optm_gain(idx);
end

function ap2 = smooth_3d(ap, sz, sigma)
%
%  smooth the AP matrix
%
h = gaussian_3d(sz, sigma);
hz = ceil(sz / 2);

ap2 = zeros(size(ap));

for i = 1:size(ap, 1)
    for s = 1:size(ap, 2)
       for g = 1:size(ap, 3)
           sm = 0;
```

```
            wt = 0;
            for x = 1:sz
                for y = 1:sz
                    for z = 1:sz
                        if i+x-hz >= 1 && i+x-hz <= size(ap, 1) ...
                          && s+y-hz >= 1 && s+y-hz <= size(ap, 2) ...
                          && g+z-hz >= 1 && g+z-hz <= size(ap, 3)
                            sm = sm + ap(i+x-hz, s+y-hz, g+z-hz) ...
                                * h(x, y, z);
                            wt = wt + h(x, y, z);
                        end
                    end
                end
            end
            ap2(i, s, g) = sm / wt;
        end
    end
end
end


function h = gaussian_3d(sz, sigma)
%
% Generate 3D gaussian kernel
%
h = zeros(sz, sz, sz);
hsz = ceil(sz / 2);

for x = 1:sz
    for y = 1:sz
        for z = 1:sz
            r2 = (x-hsz)^2 + (y-hsz)^2 + (z-hsz)^2;
            h(x, y, z) = exp(-r2/(2 * sigma^2));
        end
    end
end

h = h ./ sum(h(:)); % normalize
end
```