Towards Developing Computer Vision Algorithms and Architectures for Real-world Applications

by

Jun Cao

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved November 2018 by the
Graduate Supervisory Committee:

Baoxin Li, Chair
Huan Liu
Junshan Zhang
Yu Zhang

ARIZONA STATE UNIVERSITY

December 2018

ABSTRACT

Computer vision technology automatically extracts high level, meaningful information from visual data such as images or videos, and the object recognition and detection algorithms are essential in most computer vision applications. In this dissertation, we focus on developing algorithms used for real life computer vision applications, presenting innovative algorithms for object segmentation and feature extraction for objects and actions recognition in video data, and sparse feature selection algorithms for medical image analysis, as well as automated feature extraction using convolutional neural network for blood cancer grading.

To detect and classify objects in video, the objects have to be separated from the background, and then the discriminant features are extracted from the region of interest before feeding to a classifier. Effective object segmentation and feature extraction are often application specific, and posing major challenges for object detection and classification tasks. In this dissertation, we address effective object flow based ROI generation algorithm for segmenting moving objects in video data, which can be applied in surveillance and self driving vehicle areas. Optical flow can also be used as features in human action recognition algorithm, and we present using optical flow feature in pre-trained convolutional neural network to improve performance of human action recognition algorithms. Both algorithms outperform the state-of-the-arts at their time.

Medical images and videos pose unique challenges for image understanding mainly due to the fact that the tissues and cells are often irregularly shaped, colored, and textured, and hand selecting most discriminant features is often difficult, thus an automated feature selection method is desired. Sparse learning is a technique to extract the most discriminant and representative features from raw visual data. However, sparse learning with *L1* regularization only takes the sparsity in feature dimension into consideration; we improve the algorithm so it selects the type of features as well; less important or noisy feature types are entirely removed from the feature set. We demonstrate this algorithm to analyze the endoscopy images to detect unhealthy abnormalities in esophagus and stomach, such as ulcer and cancer. Besides sparsity constraint, other application specific constraints and prior knowledge may also need to be incorporated in the loss function in sparse learning to obtain the desired results. We demonstrate how to incorporate similar-inhibition constraint, gaze and attention prior in sparse dictionary selection for gastroscopic video summarization that enable intelligent key frame extraction from gastroscopic video data. With recent advancement in multi-layer neural networks, the automatic end-to-end feature learning becomes feasible. Convolutional neural network mimics the mammal visual cortex and can extract most discriminant features automatically from training samples. We present using convolutinal neural network with hierarchical classifier to grade the severity of Follicular Lymphoma, a type of blood cancer, and it reaches 91% accuracy, on par with analysis by expert pathologists.

Developing real world computer vision applications is more than just developing core vision algorithms to extract and understand information from visual data; it is also subject to many practical requirements and constraints, such as hardware and computing infrastructure, cost, robustness to lighting changes and deformation, ease of use and deployment, etc.The general processing pipeline and system architecture for the computer vision based applications share many similar design principles and architecture. We developed common processing components and a generic framework for computer vision application, and a versatile scale adaptive template matching algorithm for object detection. We demonstrate the design principle and best practices by developing and deploying a complete computer vision application in real life, building a multi-channel water level monitoring system, where the techniques and design methodology can be generalized to other real life applications. The general software engineering principles, such as modularity, abstraction, robust to requirement change, generality, etc., are all demonstrated in this research.

I dedicate this dissertation to my parents, my wife, and our three lovely children, Natalie, Vincent, and Melody.

ACKNOWLEDGMENTS

First and foremost, I want to thank my advisor, Dr. Baoxin Li. From the beginning of my PhD study when I took his computer vision and machine learning classes, to all the research advises that he gave to me, I have benefited greatly from his technical expertise and am deeply moved and motivated by his enthusiasm towards teaching and research. He took tremendous amount of time from his busy schedule advising on my research; without him, this PhD research would not be possible.

I also would like to thank my graduate committee members, Dr. Huan Liu, Dr. Yu Zhang, and Dr. Junshan Zhang. They gave invaluable advises on conducting research and writing better papers and dissertation; the discussion with them helped me on both the depth and the breadth of my research, and I am very grateful for them spending their precious time serving on my PhD dissertation committee.

I want to give special thanks to my long time friend, Dr. Hai Tao; he helped me write my first computer graphics program on a SGI workstation in early 1990s in our early years of graduate study, and it was the visit to his computer vision lab in University of California at Santa Cruz that inspired me to start my research in computer vision. He shared many of his insights to computer vision technology and the industry practices, which greatly enhanced my technical expertise and industrial know-how.

I have learned a lot from the collaboration with my collaborators, Dr. Yiling Wang, Dr. Ji Liu, Dr. Yang Cong, Dr. Peng Zhang, Dr. Zhigang Tu, Dr. Qiongjie Tian, among others. Many hours of discussion and coding with them inspired many new ideas, polished my technical skills, and opened up my eyes to many new technologies, and helps me stay updated with the latest development and technology trend in computer vision research and advances in artificial intelligence in general.

Dr. Kai Fu from University of Nebraska Medical Center and Dr. Qinglong Hu from Tuscon Pathology Associates provided most of the labeled medical image data, and taught

me invaluable knowledge in cell physiology and cancer pathology that can only be learned from expert pathologists; I am deeply impressed with their medical expertise and their caring to the patients' well being. What I learned from them goes far beyond the research project itself, and I cannot thank them more.

I also want to thank my colleagues and the management of my long time employer, Intel Corporation, for their support, and for helping me identify areas where computer vision can be applied, in and beyond semiconductor fabrication area. Intel Senior Fellow Dr. Karl Kempf mentored me for two years during my PhD study on applying machine learning techniques for marketing research; though this research is not included in the dissertation, the knowledge and research methodology I learned from Dr. Kempf greatly improved my research skill and enhanced my career.

I want to take this special opportunity to thank my advisor for my Master Degree in Physics, Dr. Heinz-Bernd Schuttler, and advisor for my Master Degree in Computer Science, Dr. Jeffrey Smith, at University of Georgia; not only they have taught me knowledges in science and technology, they have profoundly changed my view to life, and I feel that I owe them a PhD degree that is long overdue.

Last but not least, I would want to thank my parents, my wife Dr. Qing Zhou, and our three lovely children, Natalie, Vincent, and Melody, for their love, understanding, and patience during the long years.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

Computer vision (CV) generally refers to the capability of computer software to extract meaningful high level information from images or videos, and it attempts to enable computers to perform or mimic tasks that human visual systems can do. It takes raw digital visual data in the form of matrix of pixel intensity or a sequence of matrices of pixel intensity in the case of video as the input, and outputs discriminant (categorical) or generative (continuous) results. In the last two decades, the proliferation of digital cameras, the Internet, smart phones, surveillance camera networks, etc. have made acquiring and transmitting digital images and videos much easier; along with the advances in computing hardware and new computer vision algorithms, the widespread adaption of computer vision technology became possible, and it has profoundly changed the way we work and live. However, developing computer vision applications involves multiple processing stages and algorithms that are considered to be complicated by many industrial practitioners, and this technology is still under-utilized in many areas of the industry and the society where it should have played a bigger role; and computer vision algorithm development is still *ad-hoc* for new applications, lacking systematic approach. In this PhD research, we present the challenges in applying computer vision technology in several important areas for real life scenarios, and demonstrate how new algorithms are developed to address these challenges, and present the methodology of developing and deploying computer vision applications for real world use.

There are two major approaches for visually classifying objects, rule based (knowledge driven) approach and machine learning (data driven) approach. Rule based approach, also called appearance based approach, explicitly specifies what the object's appearance should be, and exploits these properties for tasks such as detection and recognition. The data driven approach, also called machine learning based approach, on the other hand, employs generic statistic models such as mixed Gaussian model or support vector machines, and trains the models with labeled training samples to determine the model parameters. Data driven approach learns the discriminant characteristics of the object by extract these properties embeded in the training samples; because of its ability to address many visual characteristic that are difficult to define in explicit way, e.g., the visual difference between a dog and a wolf, and the model itself does not need to contain explicit domain knowledge of the objects, are gaining popularity, and its advancement is largely the force driving the currently on-going and popular term A.I. (Artificial Intelligent) revolution as it known to the general public.

1.1 Challenges in Computer Vision Application Development

Computer vision algorithms transform the raw matrix of pixels into a higher level representation of the data, and provide understanding of the data for a specific problem(s). There are unique challenges for computer vision algorithm development in each problem domain, either for the requirement for higher accuracy, faster execution time, or broader usage scenarios. In this research, we focus on several algorithm challenges in two important areas of applications, intelligent monitoring system and medical image analysis, including efficient segmentation of moving objects, and selection of the most discriminant features. In addition, the software engineering principle for developing real world computer vision application is exemplified with the development and deployment of computer vision based water level monitoring system.

1.1.1 Effective Object Segmentation and Human Action Recognition Using Optical Flow

For object detection, raw sliding window is very inefficient in de [cite] In intelligent monitoring application, the background is not always static due to moving or shaking camera, static noise, moving background objects, etc. Segmenting the foreground objects is usually the first step of detecting the object; however, the simplest way for segmentation, which is using sliding window to segment out all possible sub-windows of the image, often does not work in real time due to the enormous number of sub-windows; to enable real time object detection, the algorithm that segments the foreground objects have to be optimized to greatly reduce the number of candidate objects.

For intelligent surveillance and related area, the data input form is usually video, which is a sequence of images, and this provides additional information on how object moves and behaves. One importation feature that can be derived from video data is optical flow (OF). Optical flow is often used to characterize the movement of the objects, and in this research, we develop algorithm to utilize optical flow information for real time object segmentation for moving background. Optical flow value characterizes local movement of pixels as well, thus it can be used as features to characterize actions in human action recognition. Optical flow values from a video is fed into pre-trained convolutional neural network to extract discriminant features for classifying action; by combining multiple RGB and optical features, we achieved higher accuracy than the current state-of-the-arts for human action recognition.

1.1.2 Selecting Most Discriminant Features for Medical Image Analysis

Computer vision technology has been trying to get its way into medical field for decades, however, their usage has not been widespread in the mainstream clinic practices, and pathologists still spend most of their time studying tissue and cell samples under microscopes, or hand analyzing other types of medical images, such as Computed Tomography (CT), Positron Emission Tomography (PET), Magnetic Resonant Imaging (MRI), ultrasound, X-ray, etc. Automatic classification for objects in medical images requires trained classifiers, and highly accurate classification require the most representative and discriminant features for model training and sample inferring. However, identifying which features are the most discriminant is a difficult task, and it traditionally relies on the trial and error with medical domain knowledge. In this research, automatic feature selection algorithm based on group sparsity sparse learning is used to detect anomalies in patient's stomaches from endoscopic images, and key frames are extracted from gastroscopic video streams by enforcing the self-inhibition rule in sparse dictionary learning.

Traditional computer vision algorithms in medical image analysis applications rely on hand-engineered features, which is time consuming to develop, requiring extensive domain knowledge, and may not yield optimal results. Also, for tissue level classification, the number of available labeled images may not be sufficient to train a deep neural network. Another challenge for tissue level classification is the image size; due to the large number of training images needed to be processed concurrently, the image size is limited by the memory size, CPU, and graphic card capabilities, among other hardware constraints. In this research, we developed a methodology to augment the data samples, reducing the image size by dividing the raw image into patches so that it can be trained on a deep neural network small enough to run on modestly configured computer hardware, and use a voting algorithm to determine the tissue level classification result from the results of classifying patches.

1.1.3 Using Software Engineering Principles for Real-world Computer Vision Application Development and Deployment

For developing and deploying real life applications, additional requirements and constraints have to be taken into consideration, such as usability, time to market (TTM), development cycle, deployment environment, cost constraint, infrastructure constraints, etc. Also, computer vision applications share many common processing stages, which can be modularized and encapsulated to help the development effort.

In this dissertation, a core algorithm, scale adaptive detection and measurement under affine template matching, is developed. And an architecture framework to enable rapid and incremental application development is proposed. Then we demonstrate how to develop and deploy a complete real world computer vision system for automated water level monitoring with this framwork. To develop real life computer vision applications, software engineering methodology needs to be followed. We exemplify this process with one application, computer vision based water level monitoring system, and it can be generalized in many scenarios in manufacturing and environment monitoring.

1.2 Dissertation Content and Major Contributions

This dissertation focuses on addressing computer vision algorithm challenges in several real life computer vision application scenarios, and applying these state-of-the-art computer vision algorithms in vision application development to solve practical problems, optimizing them for specific use cases, and deploying them for practical use. We focus on algorithms in two specific domains of computer vision applications; one is intelligent monitoring system, where real time object detection algorithm using optical flow is proposed; and optical flow is again used as discriminant features for human action recognition, which is also important for intelligent surveillance; the other one is medical image analysis system, focusing on sparse feature selection using $L1$ norm regularization and automatic feature extraction using deep learning. This dissertation also presents general software engineering principles and frameworks for real life computer vision application development, along with the firsthand experiences and lessons learned in how to apply this technology to build vision based systems from ground up. Though there are many well-known supervised and unsupervised machine learning algorithms than can be used in computer vision applications, for a specific use case, the algorithms often need to be customized, or new algorithms need to be developed to address the requirements and constraints of the problem. This dissertation focuses on these innovations and improvement in these algorithms as well.

There are three major parts in this dissertation:

Part 1 focuses on developing computer vision algorithms that can be used in intelligent surveillance systems and advanced driving assistance systems, including detecting pedestrians and vehicles from a moving platform, either on a moving vehicle, or from a turning PTZ camera; and for human action recognition. Since the input data in these applications are video stream, optical flow can be derived from the video sequences, and then used for real time foreground segmentation using a low power CPU, or being used as discriminant features for recognizing human actions. For detecting and classifying moving objects in moving background, instead of using the inefficiently sliding windows to generate candidate ROI, optical flow is used to segment foreground objects from moving background and generate candidate windows, and distance is estimated from a single image. This technology can be used in advanced driving assistance systems or autonomous vehicles and in security surveillance systems. In this research, Deformable Part Model (DPM) is enhanced with scale adaptive to reducing computation speed for an order of magnitude, and is used to classify if the candidate window contains a pedestrian or vehicle, and the object distance is estimated by the bottom of object bounding boxes to determine if a collision is imminent. For human action recognition, optical flow information is used to extract Motion

Saliency Region (MSR) as region of interest, as well as being used as raw feature vector, and this feature vector is fed into a pre-trained deep neural network (CGG-Net) to extract discriminant features for a multi-class SVM to recognize actions from a video.

Part 2 focuses on selecting most discriminant and representative features for computer vision applications in two areas in the medical field, the diagnosis of stomach disease and blood cancer. For stomach disease, endoscopic and gastroscopic image and video are analyzed with computer vision algorithms for disease diagnosis and video key frame extraction; and convolutional neural network is used for tissue level grading of follicular lymphoma as well as classifying and counting leukocytes (white blood cells).

In the first half of this chapter, sparse feature selection algorithms based on $L1$ norm regularization for computer aided endoscopy diagnosis is developed. The endoscopy images are first segmented into superpixels, and the features of each superpixel, such as RGB histogram, HSI intensity histogram, HSV-HV histogram, are used as training data to tune the parameters of an SVM based classifier to classify the sample into two states, healthy or non-healthy. Not only some feature dimensions are not contributing to determining the results and should not be included in the feature set, some type of features, called feature units, should not be selected into the feature set as well. To enforce the sparsity of both the feature units and feature dimension, $L1$ norm regularization is enhanced with additional constraint to select the optimized combination of the most relevant feature units and feature dimensions. For gastroscopic video, there are tens of thousands of frames in a 20 minutes video clip, where only a few frames are most informative; thus we developed an algorithm to automatically extract the key frames of the video that provide the best summary of the video. In order to extract the most representative key frames from gastroscopic video, $L1$ norm regularization is used again to construct a sparse dictionary enforced by self-inhibition and attention constraints; based on the sparse dictionary obtained, the reconstruction error is minimized to obtain the most informative and distinctive key frames.

In the second half of this part, the use of convolutional neural network (ConvNet) to automatically extract most discriminant features for follicular lymphoma grading and leukocytes is explored. In FL grading, instead of the traditional way of manual counting of centroblast cells, we use the entire microscopic image of tissue sample to perform the grading. To overcome the image size limit, each original image is divided into 12 patches, and the patches are used to train neuron network for classification. A voting algorithm is used to determine the overall grade based on the grade of each patch. We take both flat classifier and hierarchical classifier approaches, trained 1 ConvNet for flat classifier for multi-class classification, and trained 3 separate ConvNets, for the hierarchical approach. In hierarchical approach, we first grade the samples into low risk and high risk grades with one CNN, and then for low risk samples, we grade it further into grade 1 and 2 in second CNN, and in high risk grade, we grade it into 3A and 3B in third ConvNet. Comparing the two approaches, we concluded hierarchical approach significantly improved the grading accuracy. Leukocytes counting is important for early screening of leukemia; we developed a system to use color based segmentation in the HSV color space to segment and count the number of leukocytes; we also developed a ConvNet based classifier to determine the type of leukocytes. These information is key to the early screening of leukemia.

Part 3 focuses on the software engineering principles such as modularity, abstraction, adaptive of requirement changes, consistency, etc., for developing real world computer vision systems, including requirement analysis, hardware and infrastructure constraints, architecture design, along with development tools and deployment strategy. A computer vision based water level monitoring system, including software design and hardware setup, is used to illustrate the software engineering principles for real life computer vision application development and deployment. Instead of using traditional pyramid representation to achieve scale invariant in template matching, a more computational efficient scale adaptive affine template matching algorithm is developed to detect object under affine transformation. The region of interest (ROI) on the water gauge is first obtained based on the hue value in HSV space, and then the templates are resized to the size of these candidate blobs; matched templates are used to pinpoint exact location of the marks on the gauge and the water level is inferred by quadratic curve fitting. With this algorithm, the existing painted water level marks can also be recognized, thus further reduces the set up cost and increases flexibility.

The dissertation is organized as follows: chapter 1 outlines the dissertation, chapter 2 reviews the application of computer vision technology, application process pipeline, and computer vision algorithm overview; chapter 3 presents the algorithms and system architecture in intelligent surveillance systems using optical flow value for real time moving object segmentation with moving background, and use optical flow as features for human action recognition; in chapter 4, deep sparse feature selection algorithm and feature extraction using deep learning framework in medical image analysis are presented. Chapter 5 presents developing computer vision for real life applications using software engineering methodology, and present the design, algorithms, and real life deployment of a complete water level monitoring system; Chapter 6 is the conclusion, briefly introducing the current state-of-the-art deep learning based algorithms, and how they can be used for the research topics presented in the dissertation, as well as plans for future research works.

In this research included in this dissertation, several new computer vision algorithms and new applications are presented. The major contributions of this dissertation are:

- Innovative optical flow based object segmentation algorithm for object detection that achieves real time performance using low power mobile CPU; and using pre-trained ConvNet to use optical flow as features as well as object proposal for human action recognition with recognition rate higher than the state-of-the-art;

- Improved $L1$sparse learning algorithms to select most representative feature units and feature dimensions, and incorporate prior knowledge in medical image analysis and medical video compression;

- Overcome the input image dataset size limitation of convolutional neural network for tissue level classification, and incorporating prior knowledge of the cancer grading to improve classification performance;
- Demonstrated how to use software engineering methodology to develop and deploy a real life computer vision application for full life cycle for environment monitoring, which can be used as foundation and development framework for many similar real life computer vision applications.

Chapter 2

COMPUTER VISION APPLICATION, ARCHITECTURE, AND ALGORITHM OVERVIEW

Computer vision applications take visual data, such as images or videos, as input, analyze the data and extracts relevant information, and perform higher level functions on it. Computer vision transforms the visual input and output results that are meaningful to human users or other systems, such as detecting intruders or traffic violations from surveillance cameras. Major computer vision problem domains include object recognition, object detection, image synthesizing and generation, super-resolution, etc. Though computer vision applications have been developed since 1950s, their use had not been widespread until late 1990s when the digital cameras and smart phones became common household items for digital image acquisition. The explosive growth of the Internet makes photo sharing as easy as clicking a button, and it provides unlimited resource of training samples for computer vision algorithm development, which greatly accelerated its growth. The advancement in computer hardware, the proliferation of open source computer vision software platform such as OpenCV, TensorFlow, etc., and the development of computer vision algorithms, especially the deep learning algorithms in the last several years, greatly accelerated the adaption of this technology in our daily life and work places. In this chapter, the computer vision algorithms are briefly reviewed.

2.1 Computer Vision Application Overview

Computer vision applications are ubiquitous today, ranging from human computer interface, augmented reality, industrial robots, space exploration and satellite image analysis, smart weapon systems, medical image analysis and diagnosis, intelligent traffic and surveillance systems, to name a few; and now it has made its way into our daily lives through smart home, Apple Face ID, and advanced driving assistance system and autonomous vehicles, etc.

Computer vision technology plays the vital role in the on-going fourth industrial revolution, so called Industry 4.0, the automated manufacturing revolution, which will profoundly transform the society. Today most industry robots used in automated manufacturing still require being pre-programmed to do routine job and have little flexibility; with computer vision technology, the positions of components on the assembly line and the robotic arms can be precisely determined in real time, and the robots can visually determine the location and type of the materials they work on, thus bridging the last and critical gap between transforming human manual work to the automated work performed by robots; thus the computer vision technology has the potential to revolutionize the manufacturing industry and transform our society.

Besides discriminant tasks such as object recognition, detection, and tracking, there are also generative tasks, most noticeably visual content synthesis by using Generative Adversarial Network (GAN), which can be used to generate new images based on multiple existing images, which is growing its popularity for content creation, such as generating new art works, or synthesizing high resolution medical images from low resolution images.

The following are some examples of the computer vision applications; the actual list will be much longer and keep expanding:

- Intelligence surveillance and Traffic Enforcement
- Industrial robots in manufacturing and transportation
- Image retrieval
- Product quality control and inspection
- Autonomous vehicles
- Interactive gaming and entertainment systems
- Medical image analysis
- Augmented reality application
- Autonomous vehicles
- Space exploration navigation
- Aerial and satellite surveillance
- Precision guided weapons

Though computer vision technology has demonstrated great capability in practical use, and showing even greater potential, it is still under-used in many areas where it can assist human beings work and live better with the existing technology. Developing computer vision applications is still considered as a difficult software development task by many organizations because of the complexity of the underline mathematics. However, the common task for most computer vision algorithms is to extract high level information from the raw visual input data, then the computer vision systems must share many common characteristics, such as similar processing pipeline and common processing components, common classification frameworks, etc. In this dissertation, these common processing modules in computer vision applications are studied, and it is demonstrated that many computer vision applications can be quickly developed by combining these common components as building blocks, and how algorithms can be tailored for the specific tasks.

In this research, we focus on one type of computer vision application that are widely used in practice, the computer vision based object detection and recognition algorithms, which is the foundation for most computer vision applications. More complicated applications, such as medical expert system, intelligent traffic system, visual defect detection system, etc., are all built on top, or make use of object detection and recognition algorithms.

2.2 A Brief Overview on Object Classification and Object Detection Algorithms

Object classification and detection are the foundation of most practical computer vision algorithms. In this section some of the most important concepts of object classification and detection are highlighted.

2.2.1 Object Classification and Detection

Object classification or recognition is the process to determine which pre-defined category or label the object belongs to, and detection determines both the category and location for an object. Object classification and detection are the essential functions in computer vision technology and they are the core algorithms in most real life computer vision applications.

Since the object in real life images to be classified usually only occupies a portion of the input image, to classify it, the first step often involves segmenting it from the background. This process is called object segmentation. The object of interest may appear to be present in multiple location of the image, though each location may or may not contain the actually object; but narrowing down the search area will greatly reduce the search time, so one of the most important parts of computer vision algorithms is the region of interest (ROI) generation, where the areas that may contain object of interest have to be identified and segmented from the background, and then the object in ROI can be classified. ROI generation can rely on prior knowledge of the objects, such as color, shape, etc., or it can be segmented using the saliency features of the image. Foreground segmentation is a difficult task for cluttered and moving background and the solution is often ad hoc. In this research, we innovated algorithm using optical flow to segment moving foreground objects from the also moving background.

The process of object classification can be considered as a dimension reduction problem. Feeding a high dimensional input into the algorithm, such as an image, which is a matrix of pixel intensity values, into the computer vision algorithm, and a lower dimensional result, such as object type, will be returned. E.g., a facial recognition algorithm determines if a face image matches a specific person; a medical image analysis algorithm often returns a binary result, determining if a sample is positive or negative, or a scaler result, determining the grade or severity of the diseases, etc. Traditionally, most computer vision applications perform discriminant tasks, such as classification or detection; in recent years, with advances in new algorithms such as Generative Adversarial Network, more generative applications are developed, including generating new images from existing samples, blending real world environment with computer generated images in Augmented Reality, synthesizing people's voice, and constructing 3D images from optical or sonar sensors, etc.

Most computer vision problems involve classification; a high dimensional visual input, often a matrix of pixels (digital image) or a series of matrix of pixels (video), by going through several stages of transformation and dimension reduction, is transformed into a much lower dimensional, and often linearly separable feature set, and then the feature set is fed into some classifiers such as SVM or Softmax to return a class label. E.g., object detection takes an image and returns the location for the object of interest; facial recognition determines if a facial image corresponds to a specific person; a medical image analysis often returns a binary result, telling if a sample is positive or negative, or the grade of the diseases, etc.

All research projects involved in this dissertation research are about object recognition and detection techniques. Each research project targets a specific part in the object recognition and detection algorithms. The general process flow and architecture are briefly discussed in the next few sections.

2.2.2 Process Pipeline

Object recognition identifies the what type of object is present in an image, and it is the fundamental task for computer vision application. To detect an object, not only the object type needs to be identified, its location, normally represented by the bounding box, also need to be determined. From recognizing human face to unlocking iPhone X, to recognizing target for video guided missile, to recognizing cancer cells among normal cells, the object recognition and detection applications share many common characteristics and processing modules. The following diagram is a high level illustration of computer vision based object recognition pipeline.

The current research on computer vision algorithm in the computer vision community largely focus on the feature extraction (including both feature engineering and deep learning) and foreground segmentation (such as object proposal). They are also the focus of this dissertation.

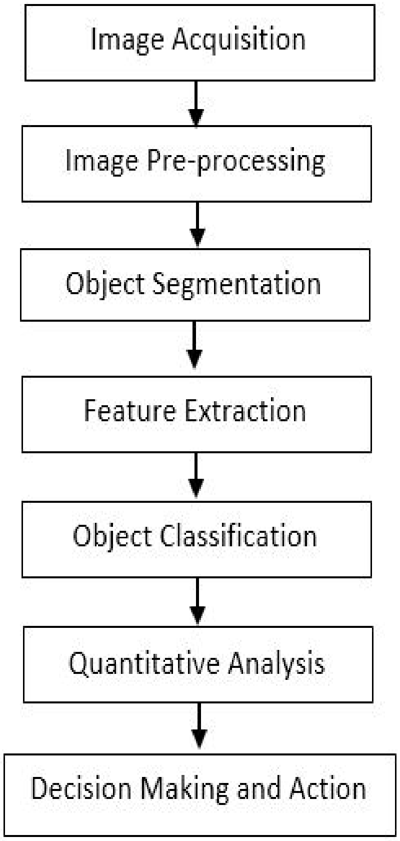


Figure 2.1: Computer Vision Process Flow

2.2.3 Computer Vision Processing Components Overview

Even the computer vision tasks and the input image/video sources can be drastically different in nature, they many share very similar process pipeline, and each stage in the processing pipeline share similar functionality. So each processing stage can be modularized and abstracted. Each stage in the diagram is briefly introduced in the following sections. This research mainly focus on the ROI generation stage and feature extraction stage, which will be elaborated in the next few chapters. There are many different ways to implement each stage of the computer vision processing pipeline. The more detailed description of computer vision application architecture

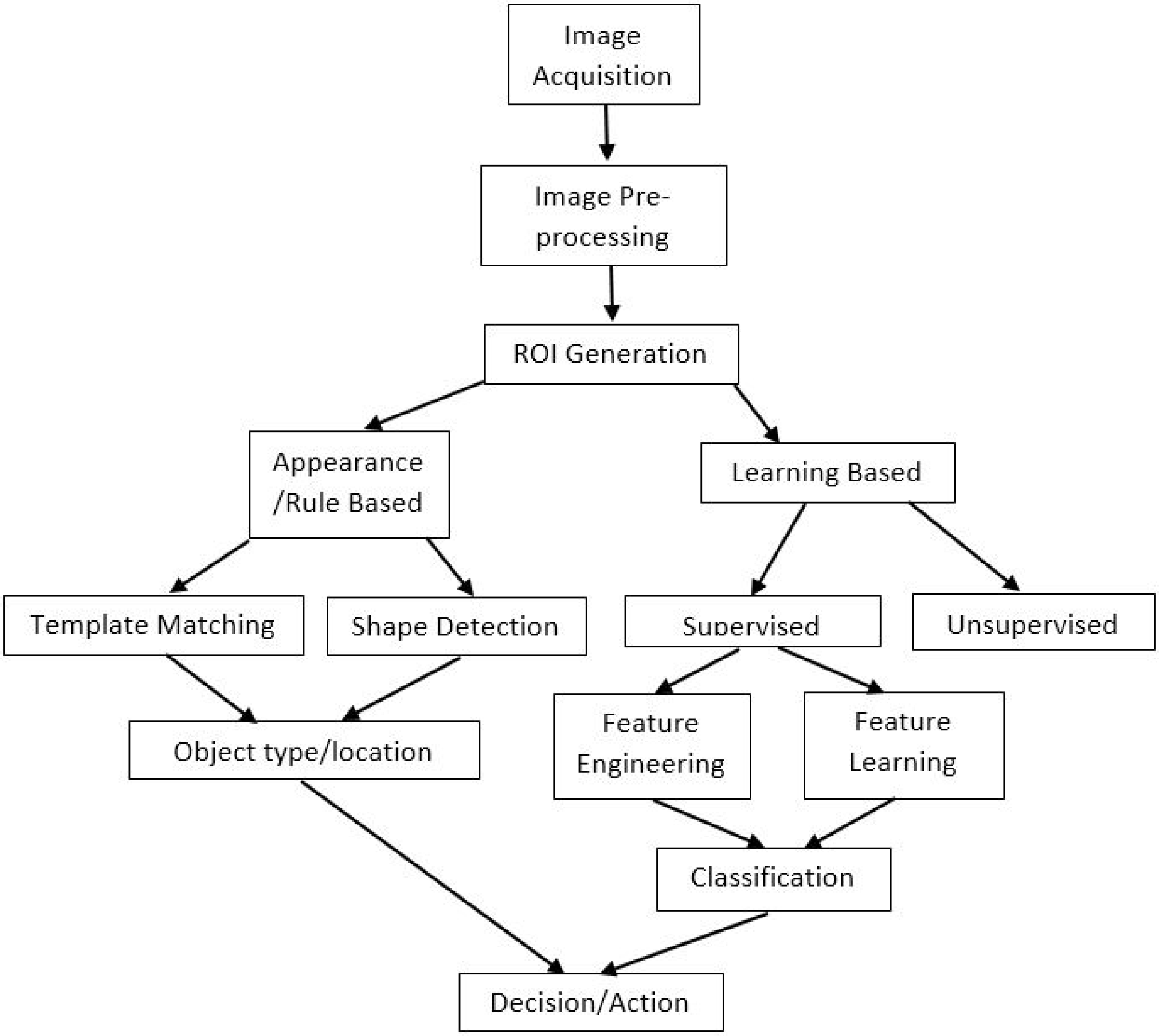


Figure 2.2: Computer Vision Based Object Detection Overview

can be found in Figure 2.2

Though the vision algorithms used in each step in the processing flow pipeline have many common functionalities, each also have their unique challenges in their specific application domain. Many kind of algorithms have been developed for different processing tasks; some algorithms are used to pre-process the images, and others are used to extract discriminant features. Algorithms can be image processing based, such as Gaussian denoising, thresholding, edge detection, hough transform to find lines or shapes, or machine learning based, including unsupervised algorithms, such as clustering pixels with similar colors, and supervised algorithms, such as object classification, image generation, etc. In this chapter, we outline the computer vision process pipeline and main categories of algorithms used in each stage. In next several chapters, the computer vision algorithms for specific usage scenarios will be developed to demonstrate how computer vision algorithms are developed towards real life applications.

2.2.4 Image Acquiring and Pre-processing

Image Acquisition: Images are not only the input to the applications; they are also needed to tune the model parameter, test and validate the solution. The first step of developing a computer vision application usually is the take the images or videos, normalize and categorizes them, mark the ground truth. Then the dataset can be used for training, validating, and testing the models.

Appearance based computer vision algorithms usually are used for the object recognition scenarios where the object can be described with simple rules; for example, the object has distinctive color or color distribution, texture, or shape that can be recognized with simple transformation. Some appearance based algorithms are also used for pre-processing the image; for example, if the color of the target objects is known, RGB to HSV conversion can be used to segment out the ROI.

2.2.5 Smoothing and Denoising

The image quality is often impacted by noise. The noise can be introduced by many factors, such as random noise at capturing, transmission, etc. Gaussian denoising is the most commonly used algorithms to remove random noise from the input image, and is often the first step in the computer vision processing pipeline.

2.2.6 Segmentation Using Intensity Thresholding

The objects of interest and background may have different brightness. To segment objects with different intensity values, Otus's algorithm is used for adaptive thresholding.

Edge Detection: Edges are the boundary between the object and the background; by detecting the edges, the object might be able to be separated from the background. Common edge detection algorithms include detecting the intensity gradient changes.

Color Based Segmentation: To segment object with a specific color, the algorithm has to be robust to impact such as lighting condition change, shading, random noise, etc. The raw images are often in RGB format, where each pixel consists of intensity of red, green, and blue components, often ranging from 0-255 for each component. However, the same color will have different RGB value under different lighting conditions. To mitigate the impact of shading and changing lighting condition, HSV (Hue, Saturation, Value) or similar color space is often used. The color value can be identified using the Hue value, and this can be used to segment objects with known color.

Shape Detection: Simple shapes such as lines, circles, squares can be directly detected with their geometric properties. One of the most popular algorithms is Hough

Transform, which uses a voting procedure to identify lines, circles, and other primitive shapes. Shapes can be recognized with other features such as number of corners and edges, the perimeter to area ratio, or with template matching.

2.2.7 Object Recognition

Object recognition is to classify an object into one of pre-defined categories; most tasks in computer vision involve some object recognition algorithms. It is the foundation for more sophisticated functions such as detection.

Rule Based Object Recognition Algorithm

Some objects have distinctive colors and shapes that can be described by limited numbers of rules. For example, to recognize a tennis ball, one can recognize it by its distinctive color (bright yellow) and shape (sphere). For this type of objects, combination of simple rules can be sufficient to recognize them with salient features. Some other objects maybe have complicated but rigid shape, and they can be matched with a pre-defined template (often under transformation).

Template Matching Template matching locates a part of a big image (target) that matches a small image (template). Exact matching is not likely unless the template is carved out of the target image, so many techniques are used when deformation exists. The template matching algorithm will decide the similarity measure, such as sum of squared error (SSD), between the template and the matching part of the target image; and a threshold is used to find all possible matchings.

Learning Based Algorithm

For many objects, because of the large intra-class variation and inter-class variation, and the challenges such as changing lighting, posing, shading, etc., it is impossible to use a finite set of rules to classify these objects. For example, to recognize a specific person, because of the changes of this person's facial expression, hair style, viewing angle, lighting, etc., use a set of pre-defined rules or templates will not work. To overcome this, learning based algorithms are used. Those algorithms are based on generic models with tunable parameters which are trained on training samples. The models will capture the essential characteristics of the object from the training samples during the training phase, and then the trained model will be used to classify the object.

2.2.8 Unsupervised Computer Vision Algorithms

Computer vision algorithms are used to implement each stages in the processing flow pipeline. Some are used to pre-process the images for intermediate results and facilitate the next stage of processing, and others are used to extract higher level information. Algorithms can be image processing based, such as Gaussian denoising, thresholding, edge detection, hough transform to find lines or shapes, where no tunable model is used; or machine learning based, including unsupervised algorithms, such as clustering pixels with similar colors, and supervised algorithms, such as object classification, image generation, etc.

Even without the label information, some important information can still be inferred from the input images. Those algorithms without other labeled training data are unsupervised learning algorithms. Two important unsupervised algorithms are template matching and clustering.

2.2.9 Clustering

Clustering is the most commonly used unsupervised algorithms that groups pixels with similar properties into a cluster; this algorithm transforms a raw image, which are basically a matrix of pixel intensity, into limited numbers of clusters, or blobs, to enable further processing. K-Means, DBScan, andMeanShift are popular clustering algorithms.

2.3 Supervised Computer Vision Algorithms

In supervised learning algorithms, the labeled datasets are used to train a generic parametric model to perform specific generative or discriminant tasks. Examples of supervised algorithms include naive Bayes classifier, random forest, support vector machine, nearest neighbor classifier, artificial neural network, to name a few. Supervised machine learning algorithms have two phases, the training and inference. In the training phase, the module parameters are determined in iterations of training steps, often through Stochastic Gradient Descent algorithm, to minimize the loss functions (or equivalently maximize the objective function), until the predicted results of the model is as close to the ground truth as possible. In the inference phase, the new data is fed into the trained model to obtain the prediction. In real life, the inference often takes place on a different environment than the training environment; the training environment normally is a powerful computer server or server cluster with specialized hardware such as GPU or TPU; the inference environment is at the user end; it can be a mobile phone or other low powered embedded devices.

2.3.1 Feature Selection

The original data are often very high dimensional, and feature vectors are usually first exacted from the raw image and then used as the input to the classifiers such as support vector machine or softmax. Most of the classifiers are linear classifier, and requires the features to be linearly separable, or can be transformed into a space where they are linearly separable. There are global features such as color histogram, image intensity, etc. that describe certain aspect of the entire image, and local features such as Histogram of Gradient, SIFT, etc. that describes each local areas in the image. The features are higher level abstraction of the original images and represent characteristics that are useful for certain discriminant tasks. The selection of the features are application specific, and often relies on heuristics.

2.3.2 Feature Learning with Deep Neural Network

In recent years with the popularity of deep learning techniques, features are started to be learned automatically by frameworks such as convolutional neural network from the raw image. Ten years ago, no one expected that we would achieve such amazing results on machine perception problems by using simple parametric models trained with gradient descent. Now, it turns out that all you need is sufficiently large parametric models trained with gradient descent on sufficient number of examples. The key function of the convolutional neural network is localized dimension reduction.

2.3.3 Objective Function and Optimization

In parametric model of machine learning, objective function is used to measure how well the model predicts the results; it is usually measured by how close the ground truth and predicted results are. The parameters of the model are fine tuned by the optimization process until the result converges.

2.3.4 Automated Feature Extraction Using Deep Learning

Feature learning can be done in many ways, such as PCA, k-means, independent component analysis, and in recent years, deep neural networks, which has more than one hidden layer to learn data representation with multiple layers of abstraction [1], have been used to learn features end-to-end; instead hand engineering features from the input data, DNN automatically extracts features through multi-stage neural network .

2.4 Foreground Segmentation and ROI Generation

2.4.1 Foreground Segmentation

In real life applications, it is very rare that only the object of interest appears in the image data; in many tasks, such as intelligent surveillance, not only objects need to be classified, their precise locations also have to be determined. Sliding window is simple but extremely ineffective, due to the shear number of possible sub-windows for a given image. For detection and tracking tasks, in addition to identify if the target object is in the image frame, the target object often needs to be located in the image or video as well, so the possible areas for the target object to appear, called candidate area or Region of Interest (ROI), need to be segmented out from the raw image first, and then the precise location can be further obtained from ROI.

2.4.2 Candidate Region Generation By Object Proposal

Sliding window is one of the most commonly used techniques to generate ROI; in this method, all possible sub-windows are searched for the objects; however, it suffered from its low efficiency due to the huge number of sub-windows, and it is rarely used in its raw form. Many image proposal methods are used, such as pre-process the image based on color or shape or other prior knowledge to generate far less amount of candidate windows. In recent years, more sophisticated learning based methods are used for candidate region generation, generating ROI using ConvNet such as in Faster RCNN, YOLO, SSD, etc.

2.5 Toolkits and Pre-Trained Models for Image Processing and Computer Vision

Due to its complexity, it is almost impossible to develop computer vision algorithms from scratch; many open source and commercial toolkits are available and provides the basic building blocks for computer vision applications. They often provide high level APIs that can be invoked from customized applications. There are many open source or commercial image processing and computer vision tools available to provide many machine learning and computer vision algorithms, so there is no need to reinvent the wheel for common functions such as edge detection or stochastic gradient descent (SGD). Nowadays building computer vision applications will largely rely on these tools and frameworks as well as pre-trained models.

One of the most widely used open source toolkit for computer vision is OpenCV, originally developed by Intel and currently supported and maintained by non-profit OpenCV.org. It is cross-platform and free of charge under BSD license. Both Python and C++ APIs are provided in OpenCV. Much of the research work presented in this dissertation used OpenCV library as the primary computer vision tool.

Other popular open source computer vision libraries include SimpleCV, BoofCV, Scikit-image, etc. Each has their unique advantages for specific areas in computer vision.

Matlab has been a popular tool in academic research for computer vision, however it is an expensive software tool and their real life usage is limited compared with other open source tools, and now its popularity has been replaced with tools with Python API such as OpenCV.

For image classification, machine learning toolkits are usually used to implement data structure and functions such as linear regression, logistic regression, back propagation.

Some machine learning tools are specifically optimized for hardware, especially in the computationally intensive deep learning area. TensorFlow from Google is currently the very popular framework, usually used in Python environment; Caffe from UC Berkeley is another popular framework, which can be used by change configuration without coding.

For deep neural networks, many frameworks, such as TensorFlow, PyTorch, etc., are available to alleviate developers from low level implementation details. Some pre-trained models that are trained on popular image dataset are also available for transfer learning. The following are the toolkit used in this dissertation. In Appendix A, the currently (as in year 2018) available toolsets are listed. The deep learning toolkits often take advantage of the parallel computing architecture of special hardware such as GPU or ASIC. A large portion of them utilize NVidia GPU thourhg CUDA (Compute United Device Architecture), and several most widely used toolkit, include Theano, TensorFlow, PyTorch. Many ASIC chips are designed to carry out specific machine learning tasks, and often APIs are developed for these AI chips, such as OpenVINO.

The following are toolkits and pre-trained models used in the research projects included in the dissertation:

Pandas: Pandas is a popular data analytical tool that provides easy-to-use data structures and data analysis tools. It is almost indispensable for storing and manipulating the input data.

Tensorflow: Tensorflow, an open source deep learning framework developed by Google, is currently the most popular deep learning framework with near 50% of the market share as of year 2018. It takes advantage of the hardware accelerator by utilizing the mass parallel computing unit in NVidia GPU and TPU (Tensor processing unit from Google).

OpenVINO: A new tool released by Intel, called OpenVINO Toolkit, which is based on Intel architecture, can take advantage of Intel based AI frameworks such as Intel processors, Movidius Neural COmput Stick, FPGA, etc., with common API.

VGG Neural Network: VGG is a popular network architecture; it has been pretrained for many popular dataset and has several variations, such as VGG7, VGG16, VGG19, etc., depending on the level of complexity. VGG7 is used in this research for feature extraction from video data for human action recognition.

Chapter 3

ALGORITHMS AND APPLICATIONS OF OBJECT AND ACTION DETECTION IN VIDEO

Videos are common data source today, even more ubiquitous than still images. They are being generated by surveillance cameras, vehicle dashboard and backup cameras, body cameras, GoPro, etc. around the clock. Videos are a sequence of still images; besides the spatial information, it provides additional temporal information to characterize the movement as well. Optical flow, which calculates a pixel-wise displacement fields between two images, can be roughly considered as how much a pixel is moved from one frame to the next, and this provide an importance cue for detecting and classifying movements in video data.

One of the most common applications of optical flow is detecting moving objects. Efficient Region of Interest (ROI) generation, especially in real time applications, is essential for the performance of object detection algorithms, since the objects of interest often only occupy a small portion of the image, and ROIs have to be generated to narrow down the search area. Many algorithms for ROI generation have been proposed by researchers, but most of these algorithms are for still images, and very few focus on the ROI generation in video data. In this chapter, we focus on real time ROI generation algorithm in video data using the optical flow (OF) value that can be computed from the video efficiently. In addition, optical flow value also provides information on how the objects move, so the discriminant features can be extracted out of the optical flow information, to classify the type of movement, or action types.

In this dissertation, we present the use of optical flow in two common surveillance scenarios, one for pedestrian and vehicle detection with moving camera and moving background problem, where innovative algorithms using optical flow for efficient foreground segmentation are developed, and then a deformable part based model is developed for object classification. Then we address action recognition problem with optical flow to segment out the motion salient region (MRS), and then segmenting out the most active parts in the MRS using optical flow again. Then the motion features represented by optical flow, as well as the RGB features in the MRS, are extracted using pre-trained convolutional neural networks; the features are combined in a feature vector, which is fed into a multi-class SVM, to classify the action.

3.1 Optical Flow and Its Estimation

For moving pedestrians and vehicles, because they have relative motion to the background, even background is moving itself, motion based segmentation can be used to create ROIs for the foreground objects by group optical flow value over a dense grid on the image.

Optical flow estimation calculates a pixel-wise displacement vector field between two images, usually 2 adjacent frames in a video, and the vector filed value normally indicates how the object moves from one frame to the next. Optical flow method is used almost in all algorithms where motion detection is important. For most of the optical flow calculation, brightness constancy constraint is assumed, which assumes a pixel's intensity value will remain constant in the next frame, and it can be formulated as follows:

$$I(x, y, t) = I(x + \Delta x, y + \Delta y, t + \Delta t) \tag{3.1}$$

Where $I(x, y, t)$ is the pixel intensity at location (x, y) at time t. One of the most widely used method for optical flow estimation is the Lucas-Kanade method, which assumes the optical flow value is constant in a small neighborhood of the patch under consideration. Though in recent years some deep learning algorithms are used to estimate optimal flow values for situations where Lucas-Kanade method faces difficulties, such as in a constantly changing lighting condition, Lucas-Kanade method is still widely used in practice due to its simplicity and computation efficiency. This method has been implemented in many open source software packages and are very convenient for free use, and it is used throughout the research presented in this dissertation.

The appearance of motion in video will cause pixel intensity to change between frames, and this property can be used to detect the present of motion for surveillance and other purposes. However, the pixel intensity change can be caused by many other factors as well, such as camera shaking, lighting change, random noise, random movement like tree leaves, static noise (especially in infred images), etc.; the optical flow value will exhibit specific local characteristics for those scenarios. For example, camera shaking causes all pixels to move uniformly, so the optical flow at all grid points in the video will have similar value and direction; lighting change and random noise cause the optical flow value and direction also random; intruding vehicles and pedestrians will have uniform optical value on the moving objects. These properties of the optical flow vectors can then be used for detecting specific types of motion, or for object tracking, and can be used to make the vision system more robust to random movement or noise.

Since optical flow value roughly corresponds to the movement of the pixel from one frame to another, for a moving rigid body, the optical flow vectors at the surface of the rigid body between frames are similar to each other, and are contrasting with the optical flow values in the background. Optical flow values at different parts of a semirigid body, such as a pedestrian, also has similar values. Thus the candidate areas for target objects can be obtained by grouping areas with similar optical values. We will demonstrate how to use optical flow to generate ROIs in a real time pedestrian and vehicle detection algorithm.

The optical flow indicates the movement of objects from one frame to another, so naturally it is used to segment moving objects and moving parts of objects from the background, as well as to characterize the movement itself, where it can be used as features to classify human actions. For segmentation, the areas in an image with significant optical flow value or the optical flow values are different from the rest of the image can be used as candidate areas (ROI); for action recognition, the optical vectors at different areas in the video can be used as features to characterize the motion, thus can be used either as raw input to a deep neuron network for feature extraction, or higher level features can be extracted from the optical flow vectors to classify the action.

There are many ways to calculate optical flow value using two video frames. Some methods, such as the variational method in [cite: Variational method for joint optical flow estimation and edge-aware image restoration], has the advantage of providing accuracy optical flow value calculation and estimate the object boundary simultaneously. However, for segmentation purpose, accurate optical value is not needed; we just need to have quick OF estimation over the grid so similar optical flow values can be grouped. In this research, to calculate optical flow, Robust Local Optical Flow algorithm proposed by Senst [2] is used for real time OF estimation. In this algorithm, the local optical flow value is estimated through robust regression with linear models which is robust to various practical problems such as changing lighting condition and appearing pixels that violate the standard Lucas-Kanade assumptions.

3.2 Pedestrian and Vehicle Detection with DPM Using Optical Flow for Fast Segmentation

To demonstrate how optical flow can be used for efficient moving object segmentation, we develop a real-time object detection system using vehicle back-up camera to alert for potential back-up collisions, and this algorithm also works with moving PTZ (Pan-Tilt-Zoom) cameras where both foreground objects and the background are moving. This work was performed at Arizona State University with collaboration with Yiling Wang and Baoxin Li, where the author collected the data set, developed the original algorithm and code, as well as testing and analysis with the collaborators. It is published as the title *Real-time Vehicle Back-up Warning System with a Single Camera* in [3].

Since the objects or actions performed only occupy a small portion of the video frame in most cases in the surveillance video stream, not only we need to determine the existence of the objects or actions, we also need to pin-point their locations in order to take appropriate actions. But exhaustively searching the entire frame is computationally prohibitive. So to detect objects or actions of interest, the first step usually involves narrowing down the search area, and segmenting out the region of interest, which is called foreground segmentation. The simplest and most widely used motion and object detection algorithms in intelligent surveillance systems often use background subtraction to detect changes between frames to segment the foreground objects; this method simply subtracts the background, or reference frame, from the video frame, and what leaves out is the foreground objects. However, this simple algorithm suffers severe drawbacks in situations such as moving background, lighting condition change, moving or shaking camera, occlusion, scaling, and rotation of the objects, etc., where the background and object of interest change from frame to frame. So more advanced algorithms are often required to address these situations. Though many algorithms in the surveillance systems are task specific and heuristic methods are employed, there are many common characteristics in those situations that similar algorithms can be used to address those scenarios.

In applying the optical flow based ROI generation, architecture and algorithms used for one important area where computer vision technology is widely used, the intelligent video surveillance system, are studied. Intelligent surveillance system analyzes the content of from the video stream captured by the surveillance cameras, either in visible light or infrared, identifies and classifies the pattern of interest, and responds to it. The basic functions in intelligent surveillance systems are to detect, locate, and track objects of interest, such as pedestrians or vehicles; or to detect specific actions of the objects, such as intrusion, loitering, fighting, people falling, traffic violation, etc. By extracting meaningful information from the video stream, when a pre-defined event such as intrusion happens, a real time alarm can be set off and appropriate actions will be taken by human or other automated systems. The same computer vision function is also used in self driving vehicles or advanced driving assistance systems, where nearby objects such as pedestrians, other vehicles, or road obstacles need to be detected for collision avoidance. So the common function in these applications is to detect the presence of the object of interest from live video stream, and trigger alerts for some pre-defined events or actions.

For most vehicles sold in US today, though many are equipped with back-up cameras, the video captured is just displayed on the vehicle dashboard without further processing, and the collision warning systems for autonomous vehicles and advanced driving assistance systems (ADAS) require powerful computer servers to process the video. We developed a highly efficient algorithm that combines segmenting pedestrians and vehicles from moving background using local optical flow values, and a scale adaptive method using Deformable Part Model to detect objects at different distances, and it is highly optimized for speed and can be executed on a low power mobile CPU; we also take a novel approach and take advantage of the known hardware setup and camera calibration information to estimate the distance between the object and the single digital camera without requiring special equipments such as dual camera, depth camera, LIDAR or sonar.

3.2.1 Overview of Pedestrian and Vehicle Detection Process

The general process pipeline follows the general computer vision application process pipeline very closes; it can be illustrate in 3.1. First the video data is captured in the data acquisition step, denoised and normalized, and then ROIs are segmented from the video data using optical flow. HOG (Histogram of Gradient) is a type of feature that are often used in people detection [4], and its value is calculated from the ROIs and used as features for the classifier. Deformable part model uses HOG as the input feature, is used as the classifier to identify if the ROI contains a pedestrian or a vehicle in this research. The distance between the objects and camera are then estimated based on the lower edge of the object bounding box, and proper actions are taken based on the object type and distance.

3.2.2 Data Acquisition and Collection

After a comprehensive search of the popular datasets for traffic data, such as Caltech Pedestrians dataset, INRIA Person Dataset and MIT Pedestrian Dataset, we found that there is no specific dataset designated for the back-up scenarios of slow moving or stationary host vehicles, and there is especially a lack of test data set for the near

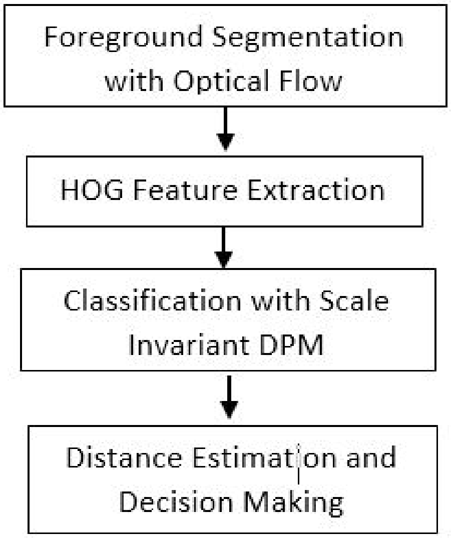

Foreground Segmentation with Optical Flow
HOG Feature Extraction
Classification with Scale Invariant DPM
Distance Estimation and Decision Making

Figure 3.1: Processing Pipeline for Pedestrian and Vehicle Detection

collision scenarios. So we created our own dataset with videos taken by the authors under different location and scenes, as well as changing lighting conditions.

Equipment Setup

To capture the videos for our own dataset, our camera setup is as shown in Fig. 2. We use a Wi-Fi controlled GoPro clip-on camera with fisheye lens of 170 viewing angle, mounted on the license plate at the rear of the car at 110 cm above the ground, and the axis of the camera lens is tilted 10° down to the horizontal line. It is worth noting that the fisheye camera is good enough when compared to some other more advanced cameras, e.g, panorama cameras. It has the following advantages: first, the fisheye camera is much cheaper than panorama camera, which is a main consideration of car manufacturer. Second, taking the panorama picture needs more computation for snitching the scenes, which makes it hard to achieve real time communication to the mobile CPU. Last, the distortion of panorama images make it more difficult than fisheye camera for detection algorithms.

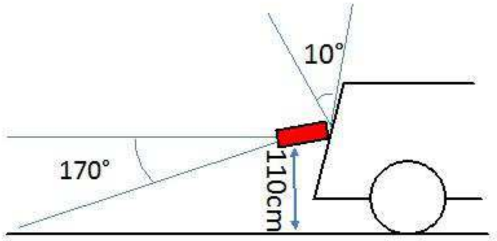


Figure 3.2: Demonstration of backup camera setup.

In the proposed approach, we define two zones. The alarm zone is shown in Fig. 3.3, which is in 3 meter radius from the camera, and the alert zone, which is in 6 meter radius from the camera. We will mainly monitor pedestrians in these zones because they are in close proximity and has higher chance to collide. For vehicles, since they are bigger and moving faster, and may post danger even they are out of the zones, so we still track those who are in close proximity of the alert zones. Because of the fisheye distortion, it is difficult to derive a closed form formula for the 3 and 6 meter radius curves shown the image Fig. 3. So we first draw the lines on the ground, capture an image with these 2 curves, and use quadratic fitting to derive 2 smooth curves to represent the 3 and 6 meter zones. Because of the fixed camera parameters, it is easy to determine if the objects are within the 3 and 6 meter zones by judging if the lower boundary of the bounding box falls into the curves.

Taking the Video

To simulate common vehicle back-up scenarios, we took video footage from many public parking lots as the host vehicle was stationary or moving. To simulate near collision scenarios, we drove two vehicles, with the camera mounted to the back of one of the vehicles, and drive to each other at low speed. To avoid accident, all the pedestrians walk in front of the camera with car stopped. There are three categories

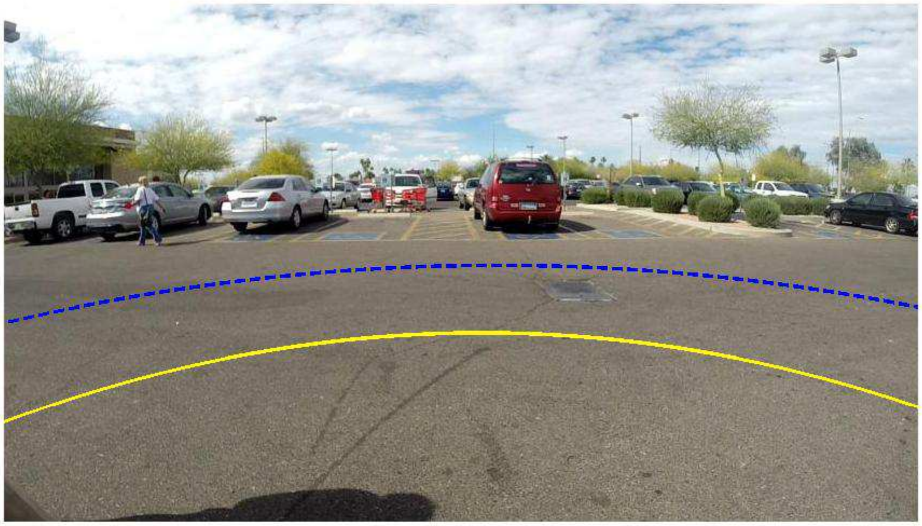


Figure 3.3: Demonstration of Alarm and Alarm Zones (3 meter and 6 meter radius) from backup camera.

in our dataset: 1. pedestrian walking or vehicles moving at rear of the vehicle. 2. vehicles backing up in the parking lot. 3. two vehicles front-rear near collision simulation. The sample frames are shown in Fig 4.

Images and Ground Truth

The dataset contains 45 videos in public parking lots. Each video in the dataset lasts from 30 seconds to 60 seconds. The frame rate of each video is 45 FPS. The video resolution is 848×480. There are minor variation in the camera position due to repeated mounting of the camera. The video was stabilized to remove effects of the vehicle pitching. The dataset summary is described in Table. 1.

There are 50,000 labeled frames describing whether there is any object of interest in the alarm zone in the dataset or not. Moreover, we uniformly sampled 800 frames

Dataset Summary	
total frames	$\sim$100K
labeled frame	$\sim$50K
#annotated frame	$\sim$800
#bounding boxes	$\sim$1700
#parking lots	$\sim$4
category 1	$\sim$15
category 2	$\sim$16
category 3	$\sim$14
labeling effort	$\sim$ 100h

and annotated for total 1700 bounding boxes for pedestrians and vehicles. It was worth noting that for every annotated frame which only visible object is drew by a tight bounding box, occluded objects such as pedestrian are marked as 'truncated', this scheme is also employed in PASCAL labeling scheme [5]. To further analyze the objects in the dataset, we group pedestrians and vehicles by their image size (height in pixels)into three scales: *near*(100 or more for pedestrian and 220 for vehicle), *medium*(70 $\sim$ 100,150 $\sim$ 220),*far*(less 70 and 150). Noting that around 94% pedestrians and vehicles in the dataset lie in *near,medium*, detection in these scale objects is also essential in automotive back up assistant system.

3.2.3 Computer Vision System Architecture

In our approach, the computer vision system for pedestrian and vehicle detection using backup camera as video data source is divided into the following 5 major software function modules:

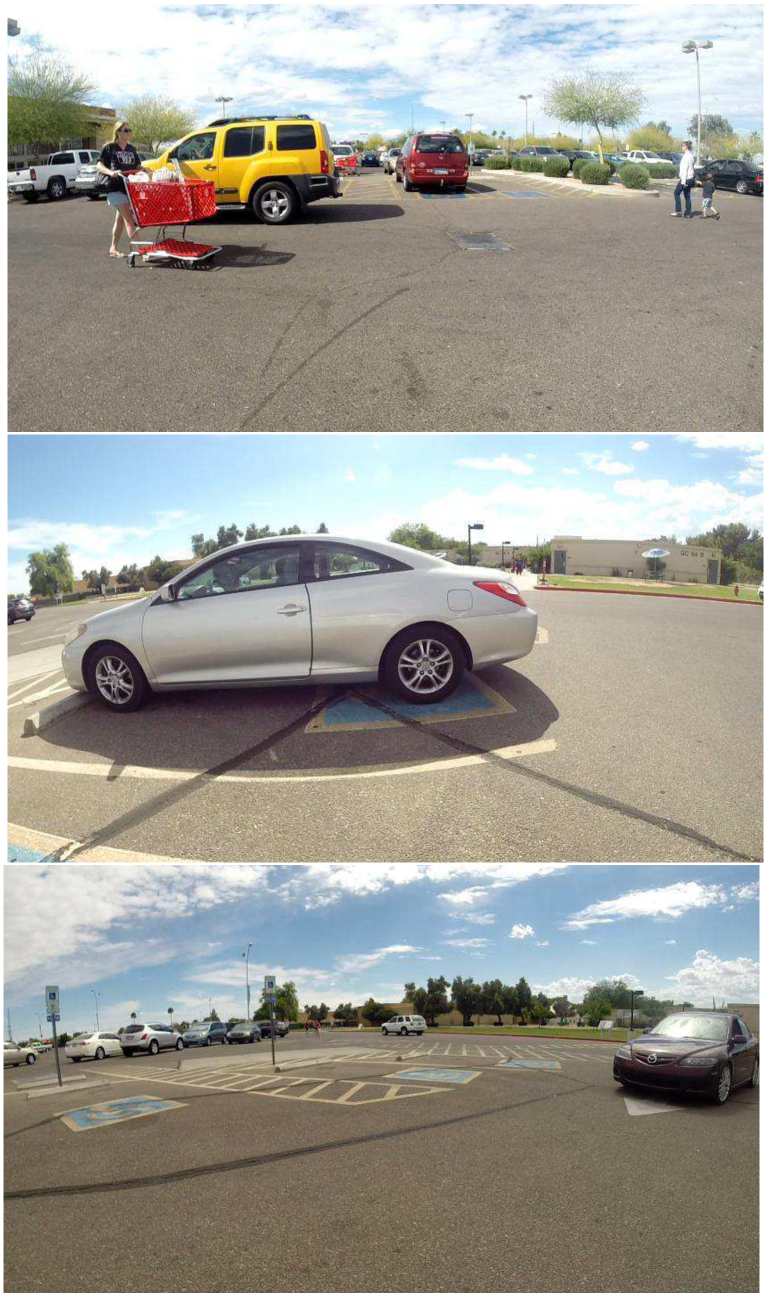


Figure 3.4: Three categories in our dataset. Top: walking pedestrian, middle: vehicle backing, bottom: two vehicles approaching.

1. Standby module: this module detects if there is significant movement with sum of the absolute value of optical flow of the current frame. When the value is below a preset threshold, then object detection module will run at 1 frame per second rate over the warning zones to detect if there are any objects of interest in alarm or alert zone.
2. Pre-processing module: if optical flow detected is above the threshold, then the optical flow will be calculated at higher frame rate in order to provide near real time detection.
3. Object proposal module: this module segments each frame by optical flow and provides estimation of the object location, as well as the size and shape of the objects detected, and then feed it into the object detection and categorization module.
4. Object detection and categorization module: In this module, the objects are detected and categorized in the segmented foreground area, and their precise locations are determined within these ROIs.
5. Relevance assessment and alerting module: In this module, the relevance of the objects are estimated based on the zones that they are located in, and the objects will be highlighted with yellow frames if they are in alert zone and red frames if they are in yellow zone.

The work flow is illustrated in the Figure 3.5.

3.2.4 Candidate Object Proposal

To alert drivers for potential collision, the presence and locations of the pedestrians and vehicles in proximity need to be determined in real time. Sliding window is

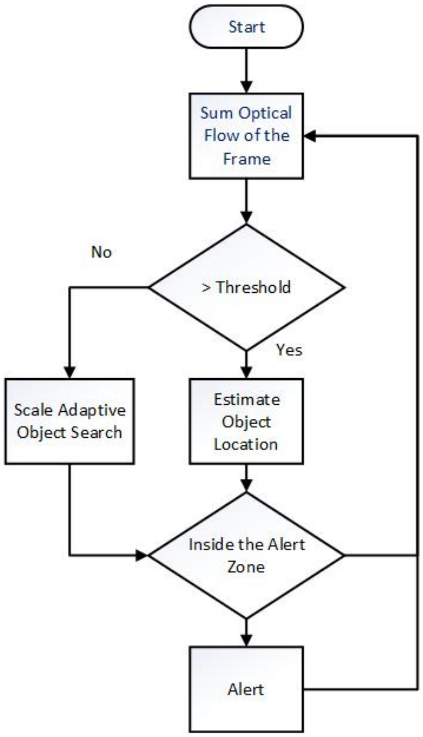


Figure 3.5: Work flow of proposed method.

a common practice in object detection and locating. However, there are about one billion rectangular sub-images even in a low resolution 320 by 240 image, and the number of sub-images grows as n^4 for images of sizes $n \times n$ [6], so it is computationally prohibitive to search all possible sub-images in full frame. Many works have been done in this area, including Binarized Normed Gradients Model , and Efficient Sub-window Search [7], which uses branch-and-bound framework, to generate a small set of candidate windows. However, even with the optimizations, it takes about one second to detect the objects of interest. Also, in this application, we are particularly interested in detecting the moving objects. In our approach, we utilize the optical flow information and estimate a bounding box that contains the moving objects, so we can only detect objects in those small sub-windows instead of exhaustive search for the entire window. By doing so we achieve real time processing with a mobile CPU.

3.2.5 Scale Adaptive Deformable Part Model

Deformable part model (DPM) is a graphical model for detection, first proposed in [8]. Unlike traditional approach, the DPM approach represents the object as visual grammar using a hierarchical structure. Each part of the object is defined directly and trained by latent SVM. This model also allows for structure variation so different model can share same object parts. Though DPM has many advantages such as high detection accuracy, the most challenging part is training stages that richer model suffers difficult training samples. For example, in practice, the training image are only labeled with bounding boxes which contain the interest area of objects, but the parts are not explicitly labeled. Since the part information is unknown, it trains the part as a hidden (latent) variable.

In practice, to detect objects such as a pedestrian, the human body is modeled as a collection of body parts such as head, arms and legs, torso, etc. And these parts are arranged in a deformable configurations, like they are connected with elastic springs. For human beings or vehicles, the star model is often used, where a human is modeled as a root (entire body) and multiple parts. Histogram of the Gradient (HOP) feature is used in DPM. The color image is first converted into gray scale, and the HOG is computed from the gray scale image. To recognize the entire human body, the root filter is used. The score for the detection is formulated in the following equation:

$$score(model, \mathbf{x}) = score(root, \mathbf{x}) + \sum_{p \in parts} \max_{\mathbf{y}}[score(p, \mathbf{y}) - cost(p, \mathbf{x}, \mathbf{y})] \quad (3.2)$$

DPM is parameterized by the part appearance and a structure model maintained the spatial relationship between different parts by using latent SVM. When the training images are completely labeled, each part has a unique bounding box and simple clas-

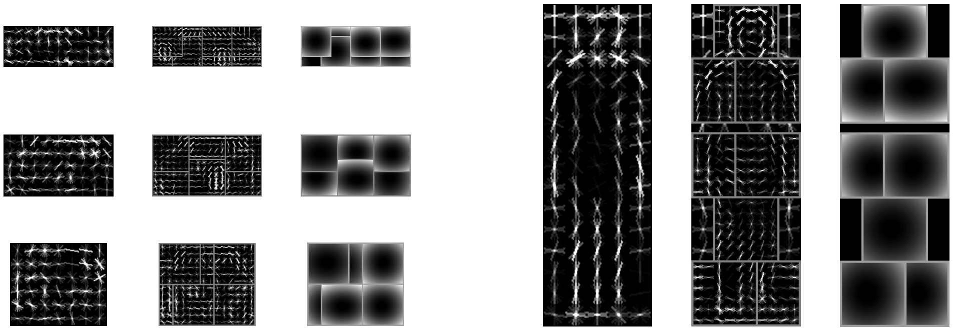


Figure 3.6: Visualization of root filter and part filters of car and human respectively.

sification method, e.g., linear SVM or LDA, can be applied [9] . While in weakly supervised setting, Expectation Maximization [10] can be used to estimate the unknown location of parts, discriminative method optimizes the error mistake by determining the decision boundary of positive and negative samples directly. The latent SVM is formulated as the following:

$$f_\beta(x) = argmax\, \beta\dot{\phi}(x,z) \tag{3.3}$$

where β is model parameters and $z \in Z(x)$ is latent value. Here $Z(x)$ is a set of all possible values for an example of x. As classical SVM, it trains β from binary labeled data $D = (< x_1, y_1 >, < x_2, y_2 > ..... < x_n, y_n >)$ where y is the label and $y \in (-1, 1)$.

The objective function is defined as following:

$$L_D(\beta) = \frac{1}{2}||\beta||^2 + C\sum_{i=1}^{n} max(0, 1 - f_\beta(x_i)y_i) \tag{3.4}$$

where $1 - f_\beta(x_i)y_i$ is the hinge loss. Since solving the objective function is non-convex optimization problem, f is a linear function of β, then the hinge loss is maximum of two convex function when set $y_i = -1$. In [8], coordinate descent method

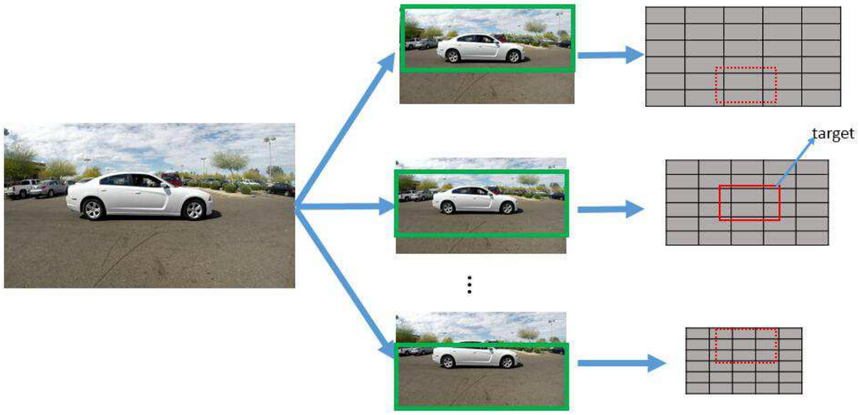


Figure 3.7: Demonstration of feature map in proposed method.

was employed for optimization. First, for Eq.2, it optimizes the L_D in term of z and in second step, it optimizes L_D over β by solving the convex function.

In [8], use of Histogram of Gradient(HOG) as discriminant feature ϕ has emerged as predominant object detection paradigm. However, pyramid feature map has to be used for detecting objects in multi-scales. Given a scale of image I, it follows sliding window paradigm, extract a d-dimensional feature vector $\phi(I, b)$ at window b across all the windows using a scoring function:

$$f(I, b) = w^T \phi(I, b) \tag{3.5}$$

In our experiment, we observe that DPM algorithm suffers from expensive computation on deciding object type and locations when the scale is unknown. Even worse, its results are undetermined when multiple objects present in the field of view. Moreover, the objects that near the rear part of the car are in more danger than those which are far away. Thus, the center part of the field of view is more important than the surroundings. To tackle those limitations, we proposed a scale adaptive DPM (SaDPM)for our situation. SaDPM estimates the scale and location of the object based on the distance from the backup camera. In Fig 4. We mark the 3m and 6m lines far away from the rear backup cameras . Since the field of the view is divided into several parts based on these two lines, we using different scale of feature map for different positions. The geometry analysis indicates that the distance d can be related to the size of the object s as follows:

$$d \propto \frac{1}{\sqrt{s}} \tag{3.6}$$

Similar to [8], object detector is a filter window $F(x', y')$, the score of confident map is the filter response (the visualization of part filters are shown in xxx Fig. 6) of a given scale feature map G, (the G is defined as sub-window of I). The matching score is defined as dot product of each sub-window(fixed) in G. For a sub-window, its top-left coordinate is (x, y). The matching score for each candidate region is computed as the following equation:

$$\sum_{x',y'} d^{\lambda} \dot{F}(x', y') \dot{G}[x : x + x', y : y + y'] \tag{3.7}$$

The score of filter responses higher than the threshold value will be the possible object locations, where the d^{λ} is the scale space that controls the corresponding scale of feature map. The illustration is shown in xxx Fig. 7.

3.2.6 Dataset and Results

Because the lack of publicly available dataset in our specific scenario, we took the video clips and created our own dataset with video at different locations, scenes, and lighting conditions from a moving vehicle. There are three categories in our dataset: 1. pedestrians walking at rear of the vehicle. 2. vehicles backing in the parking lot.

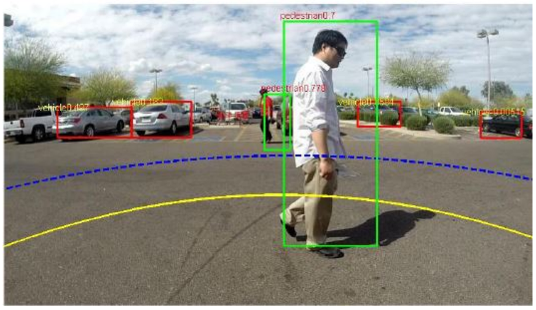


Figure 3.8: Pedestrian and Vehicle detected

3. two vehicles near collision simulation. A sample result is shown in 3.8.

The comparison results are shown in Table 3.1. The performance of raw optical flow, Deformable Part Model without object segmentation, and Scale Adaptive DPM are compared.

	alarm accuracy	speed	object category
optical flow	31.21%	0.13s	N/A
DPM	69.52%	8.34s	People/Car
SaDPM	**73.33%**	0.63s	People/Car

Table 3.1: Comparison of different methods on video level

The accuracy is computed by averaging all frame level object detection accuracies. For each frame, we also show the time cost and whether the method can distinguish among object categories. The result in Table 3.1 indicates that our method achieves highest accuracy with low time cost among them. Though optical flow is the fastest method, it has lowest accuracy and can not identify the object category. The DPM without foreground segmentation has high computation cost on object searching, more than 10 times slower than ours, because their search area for each detection is far bigger. So this demonstrated that using optical flow to narrow down the search window will greatly improve the detection efficiency.

3.2.7 Analysis SaDPM Over Original DPM

Deformable part model was the best model in terms of recognition accuracy for deformable objects such as human being, animals, and vehicles before the popularity of Deep Learning. However, it has been proven by the original author that DPM can be formulated as a convolutional neural network [Deformable Part Models are Convolutional Neural Networks]. Even though convolutional neural networks are more versatile and could yield better performance, the drawback is that CNNs are often much bigger model and require a lot more computational power, training samples, and training time. So even DPM has become less popular after the CNN gaining popularity, DPM based solution still has some unique advantage, especially in places where the computing power is limited.

The results demonstrate that our algorithm achieves very high object detection rate in near real time. Since our implementation is in Matlab, we expect significant performance gain when re-implemented in C++. With our set up and algorithm, we are able to detect and alert the driver in real time with no additional equipment cost for the auto manufacturers. This also demonstrates the effectiveness of using dense optical flow to segment moving objects from background. This is very useful for real time application where the computing power is limited and the execution time is critical.

However, SVM is a linear classifier in nature, and engineering linearly separable features are not always feasible. For features that are not linearly separable, in SVM, kernel tricks [citation here] are the common techniques used. However, there is no universal rule to select the right kernel function, and the researchers may have to rely on trial and error or heuristic method, which is time consuming and may not yield the optimal results.

At the present time, the non-linear classifier that yields best results for image classification are deep neural network based, such as AlexNet. In these models, the features are extracted automatically by multi-layers of neurons, and the extracted features are fed into Softmax or SVM classifiers for classification. We will briefly introduce several popular deep neural network architectures and pre-trained networks that can be used in transfer learning.

3.3 Object Detection and Classification with Deep Neural Network

Deep learning is a loose term, roughly referring to the multi-layer artificial neural network, as illustrated in 3.10. According to the famous Nature paper by LeCun, Bengio and Hinto [cite Deep Learning], deep learning allows computational models that are composed of multiple processing layers to learn representations of data with multiple levels of abstraction. This model exploits the properties that the features of signals are hierarchical, and the multi-layer system can capture or detect the features in a hierarchical way. This technique enables end-to-end feature learning that takes raw data instead of engineered feature as the input, and most discriminant features are automatically learned. Mathematically speaking, deep neural network is able to approximate any non-linear function with layers of combination of weight sum of input data and non-linear activation functions, thus distorting the linearly inseparable input space into a linearly separable feature space. The most popular type of deep neural network used for image recognition domain is convolutional neural network (CNN), which literally led the popularization of artificial intelligence since 2012, when deep learning algorithms started beating the many recording, including image recognition [cite], speech recognition[cite], and beating the world's best Go players [cite], which was considered mission impossible prior to the advent of deep learning techniques. Deep learning offers a great advantage of using a generic learn-

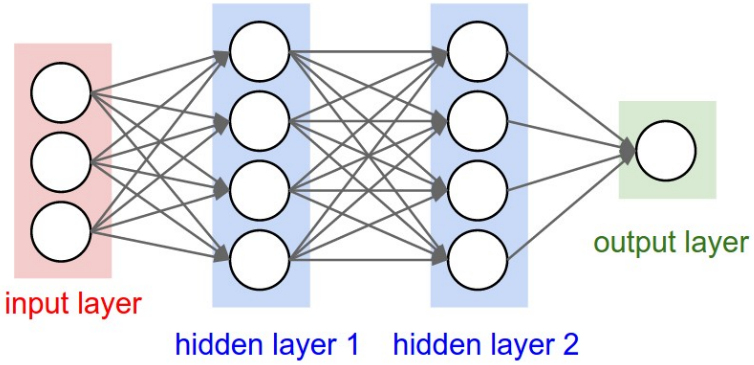


Figure 3.9: Multi-layer Neural Network

ing procedure to automatically extract the most discriminant features; on the other hand, featuring engineering is a difficult procedure, requires domain knowledge and time consuming, and often yield sub-optimal features, according to many industrial practitioners.

However, fully connected multi-layer neural network in its raw form has way too many parameters and are difficult to train. In last few decades, several deep learning architectures have been developed with great success. Among them, the most widely used ones are Convolutional Neural Network, which is especially effective for two dimensional data such as images; and Recurrent Neural Network, which is useful for time series data.

3.3.1 Convolutional Neural Network

One of the major challenges of deep neural network is the large number of parameters. To determine the values of a big number of parameters, a very large number of training samples are required; also the computing time can be very long, or even computationally intractable. Convolutional Neural Network, or CNN in short, is inspired by the structure and function of visual cortex of animals, which is adapted for visual perception of the surroundings. According to the inventor of CNN, Yan

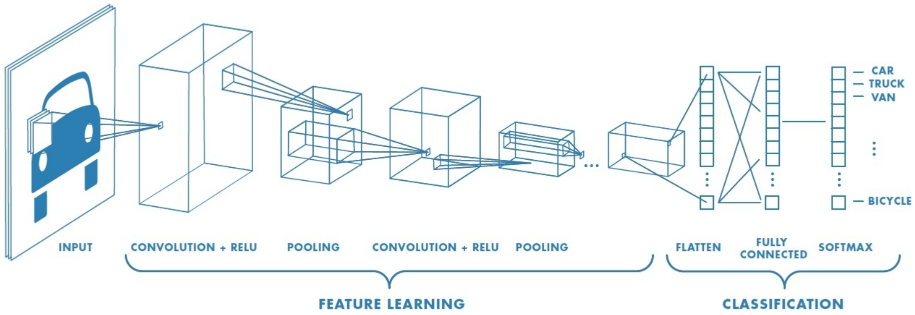


Figure 3.10: A popular Convolutional Neural Network Architecture Illustration; image courtesy to the Internet

LeCun, CNN has four properties that are specifically suitable for object recognition in images, local connections, shared weights, pooling and the use of many layers. A ConvNet can be illustrated as in 3.10

ConvNet is used to extract features that are linearly separable between classes. CNN uses layers of repetitive filters to learn hierarchical features from images. For non-linearity, activation function is used to discriminate non-linear features, and pooling is used to find the maximum features. A technique called dropout is used to prevent overfitting. Features extracted are fed into a fully connected layer and use a linear classifier such as softmax or SVM for final classification.

The key character that differs CNN with other type of deep neural network is the use of repetitive and relatives small (normally 3×3) filters that slides over the entire image; the benefit of using this technology is 2 fold; first it greatly reduces the number of model parameters by reusing a small filter over an entire image; second is that the sliding windows will in translational invariant of the feature; for example, an edge detector learned at one part of the image can be used at other parts of the image. The set of filters learned are called filter maps.

The advantage of ConvNet is summarized as follows:

- Reducing number of parameters with shared filter;

- Non-linearity with activation function;

- Further dimension reduction with Max pooling;

3.3.2 Transfer Learning and Pre-trained Convolutional Neural Network

Transfer learning in supervised learning is to reuse the knowledge learned from to problem to solve other problems. For example, a convolutional neural network trained to recognize dogs can be reused to recognize cats by only retraining the last few layers. Since several successfully implemented convolutional networks have demonstrated high performance for object recognition and they are trained over very large data sets such as ImageNet, they are often used as feature extractor to extract features used in classifiers for classification purpose.

Nowadays to recognize common objects, it is rare to train the CNN from scratch; instead, it is much more popular to take advantage of the pre-trained neural networks to either fine tune the parameters of the model with a small set of samples specific to the problem, or use the pre-trained neural networks as feature extractors and just re-train the fully connected layers with the new sample to classify the new samples.

3.3.3 Using Pre-trained Deep Learning Framework

CNN learns the features in a hierarchical way from the image. For example, we have a convolutional neural network to recognize human face; the first convolution layer learned often acts as edge detector and it detects the primitives edges; then the second convolutional layer will detect more complicated structures such as corners and blob; the layers following those will detect higher level constructs such as eyes or nose. Because other types of common objects share common visual primitives such as edges, corners, blobs, etc., those layers can be reused to recognize other objects, such as cars; also training a large ConvNet such as AlexNet could take a month on a very powerful computing platform that may not be available to many researchers. Also for some large ConvNets that have been trained over large data set such as ImageNet where over 1,000 types of objects are included, the trained ConvNets are already capable of detecting many high level constructs or even objects. So re-using these pre-trained ConvNets not only saves training time; it also utilizes the knowledge learned from the previous training over other objects, a process called transfer learning.

Training a deep neuron network not only is time consuming, it also requires enormous amount of training samples and computing resources such as GPU equipped workstations, which might be difficult or too expensive to obtain, especially for small research groups. However, using pre-trained networks that have been trained with large dataset such as ImageNet, the vast majority of the model parameters will be retained, and the computing time and hardware requirement will be greatly reduced.

Most popular pre-trained ConvNets include AlexNet, VGG16, InceptionV3, ResNet, MobileNet, Xception, InceptionResNetV2, etc., and they have their implementations in popular frameworks including Tensorflow, Keras, and Matlab, among others. VGG16 is used in this research for human action recognition tasks, as elaborated in the next section.

As stated in Stanford cs231n course note [http://cs231n.stanford.edu/], one can take a ConvNet that is pre-trained on ImageNet, then remove the last fully-connected layer (this layers outputs are the 1000 class scores for a different task like ImageNet), then treat the rest of the convolutional neural network as a fixed feature extractor for the new dataset. In an AlexNet, this would compute a 4096-D vector for every image that contains the activations of the hidden layer immediately before the classifier. We call these features CNN codes.

3.4 Extracting Optical Flow Feature with Pre-trained ConvNet in Motion Salient Region

In this section we exploit using pre-trained convolutional neural network as feature extractors to automatically extract most discriminant features for human action recognition. This work is collaborated with Zhigang Tu, Yikang Li, Baoxin Li at Arizona State Unversity and the author performed algorithm development, coding, testing, and analysis with the team members. This work is published as the title *MSR-CNN: Applying Motion Salient Region based Descriptors for Action Recognition* in the reference paper [11]

Human action recognition has many practical uses in gaming, surveillance and health care, and has been studied for decades. There are some algorithms using hand engineering feature, such as MLP [cite:] Today the most popular human action recognition methods for video rely on extracting spatio-temporal visual features using Convolutional Neural Networks (CNN) to represent video, and then use these representations to classify actions. In this work, we propose a fast and accurate video representation that is derived from the motion salient region (MSR), since only the motion features are useful for the action labeling. By improving the well-performed foreground detection technique, the Block-sparse Robust Principal Component Analysis (B-RPCA), the region of interest (ROI), humans in the foreground, in both the appearance and the motion field can be clearly detected under various realistic challenges. We also propose a complementary motion salient measure to select a secondary region of interest, the major moving part of the human, and a MSR-based CNN (MSR-CNN) descriptor is formulated to recognize human action, where the descriptor incorporates appearance and motion features along with tracks of MSR.

3.4.1 Feature Extraction using ConvNet

Even though convolutional neural network has the capability for end-to-end learning, the high dimension of the video data will make the CNN so complicated that it will require not only large amount of training data, it also takes extremely long time for the stochastic gradient decent algorithm to converge. In this research, the MSR's are used to extract motion related features. By doing so, not only the input size(dimension) are greatly reduced, it also excludes the areas where no action is taking place, thus mitigating the interference of the background. In our algorithm, first Block-sparse Robust Principle Component Analysis (B-RPCA) technique is used to detect the moving human in the foreground, and then two CNNs, one RGB CNN that is used to extract static appearance feature, and another optical flow CNN that is used to extract motion feature, are used to extract features, and finally the features extracted from each video clip are aggregated into one feature vector and fed into a multi-class linear SVM to determine its class label.

The flow of our algorithm is illustrated as in Figure 2.3.

3.4.2 CNN Descriptors

To extract MSR-CNN features from the input video data, we adopt the same architecture and training procedure as in [12]. We apply the MatConvNet toolbox [13] for the convolutional networks. Below, a brief description of our training process is given.

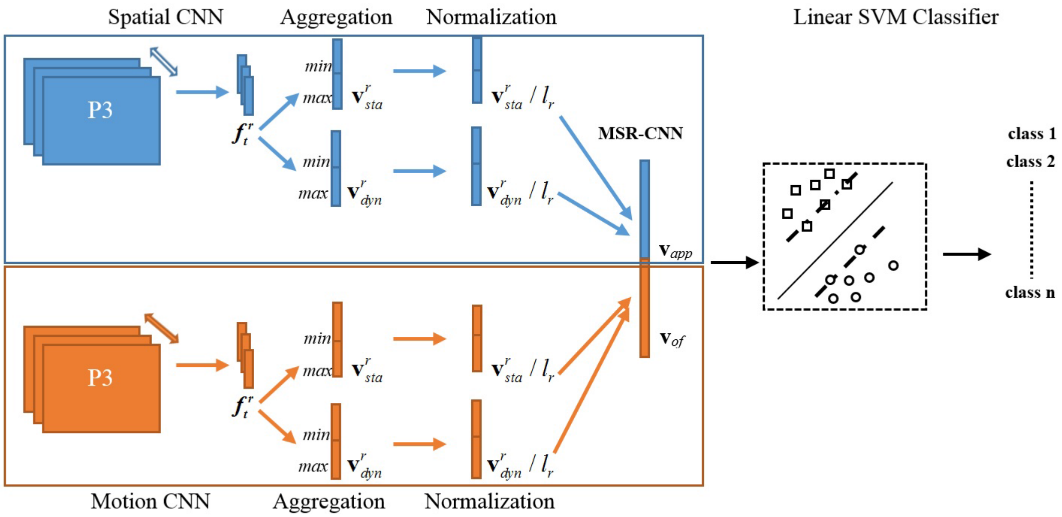


Figure 3.11: Action Recognition Architecture

Step 1: Processing the input data

To construct a motion-CNN that captures the motion feature from optical flow values, the optical flow is first calculated for each successive pair of frames according to the method described in [14]. Optical flow [15], which encodes the pattern of apparent motion of objects in a visual scene, is critically important for action recognition in video. Because optical values and RGB values could be in a very different range, they need to be normalized in comparable range. To do so, the *x*-component (i.e. *u*), the *y*-component (i.e. *v*) and the magnitude of the optical flow are rescaled to the range of [0, 255] as the values of the input RGB images as follows: first set $[\widehat{u}, \widehat{v}] = \gamma[u, v] + 128$, where $\gamma = 16$ is the rescale factor. The values that are smaller than 0 or larger than 255 are discarded. Then, the three components of every flow are stacked to form a 3D image as the input for the motion-CNN. During the training phase, for each selected MSR, we resize it to 224×224 pixels to fit the ConvNet input layer. To construct a spatial-CNN, for each selected MSR in the RGB image, we also resize it to 224×224 in order to be fed to the ConvNet input layer.

Step 2: Selecting/Training CNN model

Instead of using hand engineered features, pre-trained convolutional neural network can be used as automated feature extractor. In our research, two different ConvNets with identical architecture (similar to the network architecture described in [16], with 5 convolutional and 3 fully-connected layers) are employed to extract the representations of the MSRs on appearance and motion field respectively. The publicly available model ”VGG-f” [17], which has been pre-trained on the ImageNet challenge dataset [18], is chosen for spatial-CNN. The state-of-the-art motion network [19], which has been pre-trained on the UCF101 dataset [20], is selected for motion-CNN. The reason that these two pre-trained CNNs are chosen is because they are trained with similar objects or motions, thus they are more likely to extract relevant and discriminant features for our classification purpose.

Step 3: Aggregation

The Aggregation step is used to combine RGB and motion descriptors extracted by these two CNNs into a unified feature vector. The aggregation process is describes as follows:

(1) First a *video descriptor* is formulated by aggregating all frame descriptor f_t^r (*r* represents the MSR, *t* denotes the frame at time *t*). In particular, the frame descriptor f_t^r contains $n = 4096$ values which is the output of the second fully-connected layer.

(2) Then the *min* and *max aggregation* are formulated by calculating the minimum and maximum values for each descriptor dimensions *i* over T frames:

$$\begin{aligned} m_i &= \min_{1 \le t \le T} f_t^r(i) \\ M_i &= \max_{1 \le t \le T} f_t^r(i) \end{aligned} \tag{3.8}$$

(3) The *static video descriptor* v^r_{sta} is formulated by concatenating the time-aggregated frame descriptors:

$$\mathrm{v}^r_{sta} = [m_1, \ldots, m_n, M_1, \ldots, M_n]^T \tag{3.9}$$

(4) The *dynamic video descriptor* v^r_{dyn} is formulated by concatenating the minimum $\triangle m_i$ and maximum $\triangle m_i$ aggregations of $\triangle f^r_t$

$$\mathrm{v}^r_{dyn} = [\triangle m_1, \ldots, \triangle m_n, \triangle M_1, \ldots, \triangle M_n]^T \tag{3.10}$$

where $\triangle f^r_t = f^r_{t+\triangle t} - f^r_t$, $\triangle t = 4$ is the time interval.

(5) A *spatio-temporal MSR-CNN descriptor* is formulated by aggregating all the normalized video descriptors for both appearance and motion of all MSRs and different aggregation strategies.

Step 4: Classification

After the features are extracted by CNNs and then aggregated into a unified feature vector, the actions are categorized by using a linear SVM classifier trained on the spatio-temporal representations produced by our MSR-CNN.

3.4.3 Segmentation of Motion-salient Regions

Detecting moving objects is an extensively investigated subject [21] and significant progresses have been achieved in the past few years. Because of the impact of moving camera, moving background, etc., there are still many difficulty segmenting the moving objects. In this work, the Block-sparse Robust Principal Component Analysis (B-RPCA) technique [22] is employed for its overall good performance. Furthermore, an improved version of B-RPCA is presented to detect the foreground human in the input image. Besides, a MSM which is based on the improved B-RPCA, is

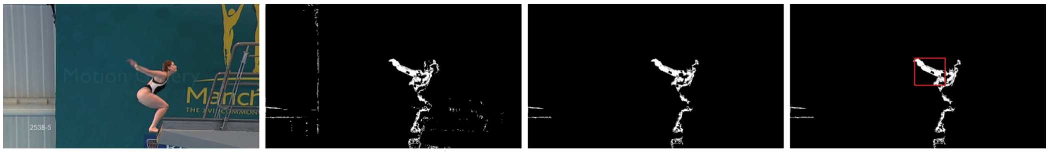


Figure 3.12: From left to right: 1. Original RGB image; OF obtained from RGB image; 2. extracting MSR candidates; 3. Discarding insignificant MSR candidates; 4. Keep the most motion salient region (the red rectangle is the selected MSR)

exploited to extract one MSR in the human body detected from the previous step. The process is illustrate in Figure 3.12.

3.4.4 The B-RPCA Method

To overcome the difficulties of detecting moving objects with moving background, Gao *et al*. [22] imposed few constraints to the background and only supposed that its appearance variation are highly constrained. The background can be identified according to a low-rank conditional matrix. Mathematically, the observed video frames can be considered as a matrix $\boldsymbol{M}$, which is a sum of a low-rank matrix $\boldsymbol{L}$ that represents the background, and a sparse outlier matrix $\boldsymbol{S}$ that consists of the moving objects. In general, the foreground moving objects can be captured by solving the decomposition in terms of the RPCA approach [23].

To improve RPCA algorithm, [22] introduced a feedback scheme, and proposed a B-RPCA technique which consists a hierarchical two-pass process to handle the decomposition problem. In particular, three major steps are carried out, which are summarized in the following subsections.

Step 1: First-pass RPCA

In this step, a first-pass RPCA in a sub-sampled resolution is applied for fast detection of the candidate foreground regions as in 3.11:

$$\min_{\boldsymbol{L},\boldsymbol{S}} \parallel \boldsymbol{L} \parallel_* + \lambda \parallel \boldsymbol{S} \parallel_1, \qquad s.t. \quad \boldsymbol{M} = \boldsymbol{L} + \boldsymbol{S} \tag{3.11}$$

where $\parallel L \parallel^*$ is the nuclear norm of the background matrix ***L***, λ is a regularizing parameter which constraints no foreground regions will be overlooked. The appropriate value $\lambda = 1/\sqrt{\max(m,n)}$. Equation (3.11) is a convex optimization problem, and the inexact augmented Lagrange multiplier method (ALM) [24] is used to solve it. Through this first-pass RPCA, all outliers can be identified and stored in the outlier matrix ***S***.

Step 2: Motion Saliency Estimation

In this step, a motion consistency strategy is utilized to assess the motion saliency of the detected foreground regions and the probability of a block containing the foreground moving objects. Pixels within the blocks captured in the first round of RPCA are tracked by optical flow. After tracking, dense point trajectories are extracted. Firstly, the short trajectories, such as $k - j <= 10$ (*j*, *k* represent the frame index, $j, k \in [1, n]$ rely on the trajectory *l*), are removed. Secondly, the method of [25] is applied to estimate the motion saliency of the remaining trajectories according to the consistency of the motion direction. There are two benefits of using the motion saliency estimation: (1) the foreground objects moving in a slow but consistent manner can be better identified; (2) the small local motion resulted from inconsistent motions of the background can be further discarded. After that, most of the non-stationary background motions identified and stored in the outlier matrix ***S*** in the first step are eliminated or suppressed.

Step 3: Second-pass RPCA

In this step, the λ value is reset based on the motion saliency, which ensures the changes derived from the foreground motion can be completely transferred to the outlier matrix ***S*** and will not stay in the background. The second pass RPCA (which is a block-sparse version of the traditional RPCA) is implemented as follows:

$$\min_{\boldsymbol{L},\boldsymbol{S}} \| \boldsymbol{L} \|_* + \sum_i \lambda_i \| \boldsymbol{P}_i(\boldsymbol{S}) \|_F, \quad s.t. \quad \boldsymbol{M} = \boldsymbol{L} + \boldsymbol{S} \tag{3.12}$$

Where $\| \cdot \|_F$ is the Frobenius norm of the matrix. $\boldsymbol{P}_i$ is an operator which unstacks every column of matrix ***S*** and returns a matrix that represents the block *i*. The inexact ALM algorithm is used again to solve this equation. Most importantly, the B-RPCA imposes the spatial coherence constraints to the foreground objects in the outlier matrix ***S***.

3.4.5 The Improved B-RPCA Method

The motion saliency estimation in [22] utilizes the trajectory length and the motion direction of the point trajectories to remove the non-stationary background motion. This strategy is effective for detecting the foreground moving objects that keep moving constantly in the scene. If the object stops occasionally, the foreground object cannot be detected via the B-RPCA technique due to the Step 2 operation, the motion saliency estimation. This is especially true for the action recognition datasets, such as JHMDB, where the actors may have little motion for some periods in the video clip. To overcome such difficulties of the B-RPCA approach, we propose the following improvements:

1) Relaxing the constraint of the trajectory length to $k - j <= 5$. In this way, falsely-removed foreground due to trajectory length (e.g., when the actor suddenly stops

for short moment) will be significantly reduced. Meanwhile, to avoid noise arising from the background, we add the motion derivative constraint similar to MBH [26]. By calculating derivatives of the optical flow components *u* and *v*, the background motion due to locally-constant camera motion will be excluded.

2) Enhancing the consistency measure of the motion direction. Not only the negative direction or positive direction of *u* and *v* along the trajectory, but also the direction variation should be considered. Hence, we add a velocity angle measure as follows:

$$\triangle\theta = \arctan(u_{t+1}/v_{t+1}) - \arctan(u_t/v_t) \in [-\pi/4, \pi/4] \tag{3.13}$$

where $[u, v] \neq 0$. Same as the motion direction consistency operation, this velocity angle measure is also conducted at positions where the velocity is no-zero along the trajectory.

3) If u_t and v_t satisfy either of the following conditions (Equation 3.14 or Equation 3.15), we consider the actor is static between frame *t* and frame $t + 1$. Then, we only perform the first step as described in ***Step 1*** to detect the actor in the RGB images.

$$range(u_t) < 0.5 \wedge range(mGflow) < 0.5 \tag{3.14}$$

or

$$range(v_t) < 0.5 \wedge range(mGflow) < 0.5 \tag{3.15}$$

where $range(u_t)$ denotes the difference between the maximum value of u_t and the minimum value of u_t. ***mGflow*** is the magnitude of the gradients of the optical flow (u_t, v_t), $mGflow = \sqrt{(u_t)_x^2 + (u_t)_y^2 + (v_t)_x^2 + (v_t)_x^2}$, 0.5 is an empirically selected threshold and denotes a half pixel distance.

3.4.6 Selecting the MSR of human

As suggested in [12, 19, 27], selecting suitable MSRs of the actor body is essential for the performance of human action recognition, as these body parts are complementary and are potentially helpful for improving action recognition when combined in an appropriate manner. Based on the captured human information of the improved B-RPCA, we introduce a MSM to select the MSR of the human body, where the motion is most distinguishable.

In this method, the region of the foreground actor body is detected and localized via the improved B-RPCA. Accordingly, the location information is useful for identifying other informative body parts which are discriminative. We employ the following steps, as illustrated in Fig. 3.13.

(1) Extracting MSR candidates from the human body detected according to a conditional measure defined as:

$$LabH \wedge (mGflow > AmGflow) \wedge (mflow > Amflow) \tag{3.16}$$

and

$$(|u| > Au) \vee (|v| > Av) \tag{3.17}$$

where *LabH* is the human body that is already obtained from the previous step. *AmGflow* is the mean value of $mGflow$. $mflow$ is the magnitude of the optical flow $flow = (u, v)$, $mflow = \sqrt{u^2 + v^2}$. *Amflow* is the mean value of $mflow$. Au and Av is the mean value of the horizontal flow u and the vertical flow v.

(2) Discarding the small MSR candidates. Different body parts have different motion patterns. In addition, some background motion around the human body may be inaccurately identified by the B-RPCA technique. Due to this implementation, the incorrectly captured background motions could be removed once again. The third

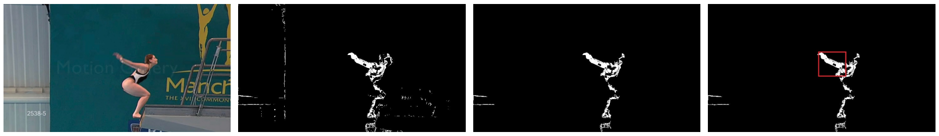


Figure 3.13: An outline of our method to select one MSR in the human body. From Left to Right: the result of step 1 - extracting MSR candidates, the result of step 2 - discarding the small MSR candidates, and the result of step 3 - selecting the most salient motion region (the red rectangle).

subfigure in Fig. 3.13 demonstrates that most of the outliers are suppressed with this method.

$$MSR(i) > \tau \tag{3.18}$$

where *i* is the index of *MSR* candidates. τ is a threshold. If the area of one candidate $MSR(i)$ is smaller than τ, then it will be removed. In this research we set $\tau = 10 \times 10$ as an empirical value.

(3) Capturing the first two largest MSR candidates. The simple MSM method described as [19] is adopted to select the final MSR by comparing the normalized magnitude of the optical flow between these two candidates as the following:

$$flow_m(R_i) = \frac{1}{|R_i|} \sum_{j \in R_i} flow(j) \tag{3.19}$$

where $flow_m(R_i)$ is the normalized magnitude of the optical flow in the *i*-th MSR candidate, *j* is the index of the optical flow. The MSR candidate with larger $flow_m(R_i)$ will be finally selected.

3.4.7 Experiments

In this section, we evaluate our MSR-CNN method by testing it on two challenging datasets – UCF Sports [28] and JHMDB [29], and compare the results with the state-of-the-art algorithms. In particular, in each dataset, we assess our method in two

Table 3.2: Results (% mean average precision (mAP)) of the spatial-motion MSR based CNN on the UCF Sport dataset.

Patches	Div.	Golf	Kick.	Lift.	Rid.	Run	S.Board.	Swing1	Swing2	Walk	mAP
P1	100	100	100	100	100	63.89	0	87.67	100	100	85.16
P2	100	100	52.50	100	100	63.89	0	100	100	100	81.64
P3	100	100	52.50	100	100	63.89	63.89	63.89	100	100	84.42
P1+P3	100	100	100	100	100	29.17	52.50	63.89	100	100	84.56
P2+P3	100	100	100	100	100	29.17	25.0	100	100	100	85.42
All	100	100	100	100	100	63.89	100	87.67	100	100	**96.39**

aspects: (1) whether the improved B-RPCA is effective in detecting the foreground human under complicated situations and background, and the proposed MSM can extract the MSR of the body part; (2) whether the extracted secondary MSR is complementary to the first one, and if it can further improve the performance.

Evaluation on UCF Sports

Fig. 3.14 shows the two detected MSRs on 6 different action categories. These actions are performed in various challenging conditions, such as multiple actors and the displacements are larger than the object scale (the 1th and 3th subfigures), the moving area with no texture (the 2nd subfigure), occlusion (the 4th subfigure), motion blur (the 5th subfigure) and illumination changes (the 6th subfigure); it is demonstrated that our two detectors can successfully deal with these difficulties.

Table 3.2 shows the results of our proposed MSR based spatial-temporal CNN technique on using different patches. Comparing the first row and the second row, we can find that for some sequences, the extracted ConvNet features from these two MSRs are different and complementary. Consequently, extracting both of these two MSRs are necessary as they have different contributions for action recognition. Integrating the three patches together, significant gain is achieved, where the *mAP* is increased

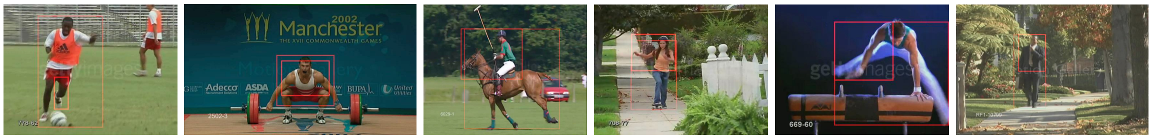


Figure 3.14: Results on UCF Sports. Each column represents an action class. The big rectangle corresponds to the extracted foreground human body via the Improved B-RPCA method, the small one corresponds to the extracted secondary MSR via the proposed MSM.

from 84.02% (*P3*) to 96.39% (*All*).

Evaluation on JHMDB

Fig. 3.15 shows the action detection and localization performance of our improved B-RPCA as well as the MSM. It is clear that our detectors perform well in complex and realistic situations. For example, in the 5th subfigure, even encountering with the extremely motion blur, the human body and one of his body part are accurately captured.

Table 3.3 demonstrates again that different patches play different but also significant roles in action recognition, and incorporating them can further improve the action recognition performance. The results, 71.1% from *All*, outperforms other approach more than 4% (comparing with the second best result 68.2% of *P2+P3*).

Table 3.3: Results (% mean average precision (mAP)) of the spatial-motion MSR based CNN on the JHMDB dataset. The respective performance of P1, P2 and P3, and the combination performance by integrating them in different ways are shown.

Patches	brushhair	catch	clap	climbstairs	golf	jump	kickball	pick	pour	pullup	push	run	shootball	shootbow	shootgun	sit	stand	wingbaseball	throw	walk	wave
P1	76.6	54.6	63.3	58.5	88.8	43.4	48.0	57.7	87.4	98.8	82.0	55.3	38.6	80.1	60.1	74.8	72.9	63.4	8.9	85.8	57.3
P2	93.6	54.0	71.9	45.9	80.8	54.1	54.6	61.8	80.8	97.1	93.0	65.6	48.4	70.1	63.7	65.9	69.5	60.6	22.5	64.7	39.2
P3	66.9	53.1	51.2	65.9	91.3	47.3	55.2	56.3	97.9	100	85.6	52.8	42.4	91.4	72.2	61.4	66.3	32.7	29.2	86.4	46.9
P1+P3	76.4	56.5	52.1	53.4	91.3	56.6	59.8	56.3	92.6	100	86.8	52.2	47.5	93.1	67.7	59.7	66.4	49.0	16.8	88.0	65.5
P2+P3	86.0	51.0	65.5	53.4	91.3	57.0	52.3	61.8	92.6	100	87.3	67.0	50.0	91.8	68.0	64.0	65.6	55.5	20.9	88.0	62.9
All	89.1	47.3	61.3	54.1	91.3	60.1	59.5	69.4	97.6	100	96.0	71.8	50.8	92.1	71.0	65.5	72.9	60.1	40.3	86.9	55.2

The results on JHMDB dataset is shown as 3.15.

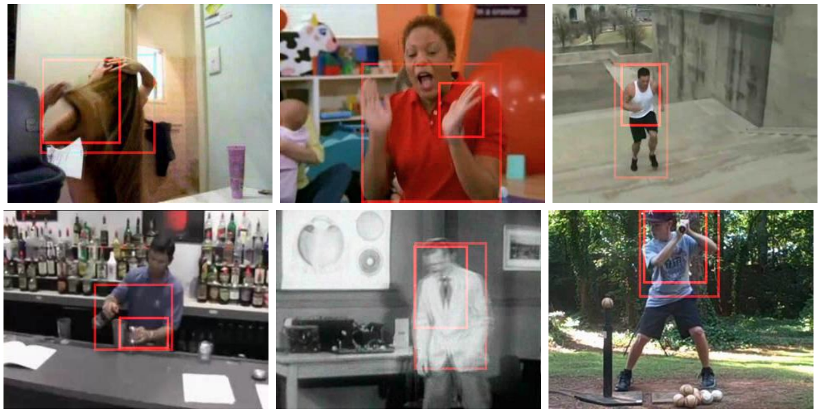


Figure 3.15: Results on JHMDB. The big rectangle corresponds to the extracted foreground human body via the Improved B-RPCA method, the small one corresponds to the extracted secondary MSR via the proposed MSM.

Comparing to previous human action recognition algorithms, our algorithm can be more efficiently implemented mainly due to two characteristics: (1) motion is involved in only part of the RGB image, and only motion field need to be processed to extract the discriminant features; (2) fewer data is needed as input for feature extraction using ConvNet. Comparative evaluation on JHMDB and UCF Sports datasets demonstrates that our method outperforms the state-of-the-art in both accuracy and efficiency.

3.5 Using Optical Flow for Object Proposal and Action Classification

In this chapter, we demonstrated the use of optical flow for the algorithm development in two important application areas, real time moving pedestrian and vehicle detection and human action recognition from video data. For pedestrian and vehicle detection, we presented a scale adaptive algorithm for object detection and classification in moving background with an efficient ROI generation algorithm using optical flow, which can be generalized for real time object detection for moving objects in a moving background. Based on estimation of the distance and object size, the deformable part model only needs to be used to search ROIs that highly likely contains the objects of interest, thus avoids using time consuming and computationally expensive feature pyramid. Also Human action recognition algorithm with pre-trained convolutional neural network as feature extractor is presented, where optical flow and RGB features from motion saliency regions are extracted using two pre-trained ConvNets, and then aggregated into a unified feature vector for classification using SVM as classifier. This transfer learning methodology can be applied for human action recognition beyond the two datasets used, and can be used for real life action recognition application in surveillance and health care industry. In both case, optical flow values derived from video data play significant role to increase the computational efficiency and improve classification accuracy.

Chapter 4

BUILDING COMPUTATIONAL MODELS FOR MEDICAL IMAGING APPLICATIONS

In this chapter, the application of computer vision to one of the areas where computer vision can be used to benefit the well being of the mankind, the medical care, is introduced. In health care, computer vision application is to analyze medical images, either to perform low level tasks such as image denoising, contrast enhancement, resolution enhancement, or perform higher level tasks, such as cell counting, or locate cancer cells. They are also used to help physicians perform diagnosis based on medical imagery and other medical test results. The recent advance in machine learning, especially with the advance in deep learning, significantly improved the performance of computer aided medical image analysis systems. In this chapter, we will be focusing on how to extract the most discriminant features from medical data sources such as endoscopic and microscopic images and videos, and microscopic images for blood and lymph fluid.

4.1 Background and Motivation

The health care industry in United States is the largest sector of the economy, accounting for about 18% of US GDP, vs. 12% of manufacturing, in 2017. One of the reasons that health care is so expensive and inefficient is because the health care system is very labor intensive, involving manual works of many highly paid medical professionals. Automation, just like it did in the manufacturing, could hold a key role to making health care more efficient and affordable. In health care industry today, most of the diagnosis works are still carried out manually; pathologists and clinic lab technicians together spend millions of hours each year to count cells in blood and tissue samples under microscope; yet for the same tissue samples, different pathologists may come up with different diagnosis results. Automating the medical image processing and analysis potential may not only same millions hours of human labor, but also improve the accuracy and consistency of the diagnosis.

Though Computer Aided Diagnosis is gaining more momentum in medical research area, their clinic usage is still limited due to lack of communication between medical researchers and computer scientists, and the lack of sophisticated computer algorithms designed for this purpose [30]. The computer scientists often lack medical domain knowledges, and the medical researchers think very little on how to automate their jobs. This gap can be bridges by machine learning, where the domain knowledge of the medical doctors are represented in the labels they provided for the medical dataset, and the computer scientists can use the data set to train a generic machine learning algorithm for classifying the disease. In this research, we collaborated with hospital to collect samples from patients and healthy individuals and build a dataset labeled by medical doctors, and uses deep sparse SVM to determine if a sample represents healthy or non-healthy state.

Comparing to recognition for daily life objects, medical image analysis has their unique characteristics and challenges, and how to effectively extract the most representative and discriminant features is essential for diagnosis. We developed several algorithms in these fields; due to the limited availability of medical images, the research is focused on two areas, processing endoscopic images and videos for stomach disease, and analyzing microscopic images for blood cancers, including follicular lymphoma and leukemia. It is also demonstrated that feature engineering based algorithms using linear classifier and feature learning based algorithms feature extraction using end-to-end learning algorithm with deep neural network both work in their respective domain, though deep learning based automated feature extraction has been showing better performance with their recent advancement since year 2016.

4.2 Evaluation Criterion For Medical Diagnostic Algorithms

In medical diagnosis, there are four possible diagnosis outcomes, True Positive (TP), where the diagnostic result is positive and the ground truth is also positive; True Negative (TN), where the diagnostic result is negative and the ground truth is also negative; False Positive (FP), where the diagnostic result is positive but the ground truth is negative; False Negative (FN), where the diagnostic result is negative but the ground truth is positive. The Receiver Operating Characteristic (ROC) curve is usually used to measure the accuracy for multiple threshold values. The ROC consists of true positive rate (TPR) and false positive rate (FPR), of which TPR determines a classifier or a diagnostic test performance on classifying positive instances correctly among all positive samples available during the test, and FPR, on the other hand, defines how many incorrect positive results occur among all negative samples available during the test. These measures are given by the formulas in Eq.(4.1) and Eq. (4.2):

$$TPR = \frac{\text{True positive}}{\text{True positive} + \text{False negative}} \tag{4.1}$$

and

$$FPR = \frac{\text{False positive}}{\text{False positive} + \text{True negative}}, \tag{4.2}$$

where True Positive (TP) is the correctly labeled abnormal samples; False Negative (FN) is incorrectly labeled normal samples; False positive (FP) is incorrectly labeled abnormal samples; and True negative (TN) is correctly labeled abnormal samples.

This can be expressed in the form of confusion matrix:

True positives (TP)	False Negatives (FN)
False Positives (FP)	True Negatives (TN)

Table 4.1: Confusion Matrix

For pixel-level and frame-level, we choice different thresholds and compute the TPR and FPR accordingly to generate the ROC curve. Furthermore, the Area Under Curve (AUC) is used for statistical evaluation. In this research, we use these criteria to evaluate the performance of the medical image analysis algorithms.

4.3 Feature Extraction in Medical Image Analysis

The diagnosis of many diseases, such as stomachache ulcer, many types of tumors or cancers, blood diseases, blood clogs in brain, etc., is relying on medical imaging technologies. Classifying a medical image into positive (unhealthy) or negative (healthy) is the essential task for disease diagnosis based on medical image analysis. The medical image can be x-ray image, MRI image, endoscopic image, microscopic image, to name a few. These images are often analyzed by medical experts such as pathologists or oncologists. Based on certain visual characteristics, the medical experts can determine if the human tissue in the medical image is healthy or unhealthy. This process is slow and tedious, and subject to human error and inconsistency. In the last few decades, computer vision technology had been introduced to analyze the images to assist human experts by alleviating them from manual and repetitive work, or partially replace the human physicians and automate medical diagnosis in the future.

At the beginning, image processing and computer vision technologies were used to enhance the medical images using techniques such as contrast enhancement [31] so the images will become easier for human experts to analyze; then more sophisticated algorithms were developed for medical expert systems to automatically diagonalize the disease [32]. Human experts diagnose medical images using certain visual features of the problematic area; for example, in the stomachache, the area with ulcer appears darker than the surrounding areas, thus brightness or intensity value can be used as a feature to separate healthy tissues from unhealthy ones. To accurately diagnose a disease, multiple visual features are needed. To enable the algorithms for automated diagnosis, optimal features also are needed to achieve optimal performance.

The raw input values, such as raw images, are normally very high dimensional. For example, a 1000 by 1000 pixel color image has 3 million dimensions, normally far exceeding the number of available training samples, and suffer from the *curse of dimensionality*. For a specific classification task, the dimension of the input images can be well above the number of available training samples. As a result, the features, which are higher level abstraction of the raw data, are extracted from the original data, and those extracted feature vectors instead are used as the input values to the classifier. Traditionally, features are hand-engineered, and since year 2008, automatic feature extraction using deep neural networks is gaining more popularity [1] . In computer vision algorithms, both global features, with reflect some properties of the overall image such as average brightness and color distribution of the whole image, and local features, which reflect properties of local areas inside an image, such as LBP and SIFT, are used. The example of global features are average pixel intensity, RGB histogram, HSV histogram, etc., and the example of local features includes SIFT (Scale Invariant Feature Transform) [33], SURF(Speeded Up Robust Features) [34], LBP (Local Binary Patter), etc. In machine learning algorithms, in the training phase, the feature vectors are used as training data to determine the model parameters by minimizing the loss function, usually through an iterative stochastic gradient decent in backpropagation process. After the model (classifier) is trained, *i.e.*, the model parameters are determined, the feature vectors extracted from the raw data is fed into the trained classifier to determine what class it belongs to; in the medical applications, usually the the classifier will output if the sample is positive or negative of a medical condition.

Selecting the most representative and discriminant features plays a vital role for performance of the classifier. Since one feature only reflects one aspect of the properties of the input image, such as color or texture distribution, multiple features are often combined to form a feature vector that will better represent the image for the specific discriminant tasks such as classification, and the feature vector is used as input data for machine learning algorithms such as logistic regression, SVM, artificial neural network, etc. To obtain the optimal combination of the features, not only we need to identify which features are to be used, we also need to determine their weights when combining the features into one feature vector, since one feature could be more important than the others for a specific task. Heuristics are often used for a specific problem; however, systematic approach needs be developed for problems with similar properties. In this research, we use two examples to demonstrate how optimal feature vectors can be generated algorithmically by selecting the most discriminant features using sparse learning techniques; one example is a supervised learning problem, which is to select the best combination of the features to classify the gastronomy images, and the other is an unsupervised learning problem, which is to generate a sparse visual dictionary and use it to select the key frames in gastronomy video. In both cases, sparse learning, combined with additional domain specific constraints, can be used to select the most discriminant and representative features algorithmically.

4.3.1 Sparsity with $L1$ Norm Regularization

The assumption behind of sparse learning is that good-behaved images have a sparse representation; i.e., the image can be constructed with a limited number of elements. For example, in JPEG format, the cosine functions are a dictionary, and there are only a few non-zero coefficients for the dictionary, which implies that the image can be constructed with only few cosines functions. If the image is pure random noise, then no sparsity can be achieved. The same assumption can be imposed to video as well. In order to learn the most important features for classification to optimize for speed and accuracy, we want to minimize the number of features, and we only want to have the most discriminant and representative features being selected. Similarly, in sparse coding, we want to minimize the number of atoms in the learned dictionary so the original data can be represented with linear combination of minimal number of atoms, thus the video can be compressed with maximum efficiency. So in both scenarios we try to minimize the number of non-zeros elements in weight matrix or dictionary entry, which is to minimize the $L0$ norm of the coefficient vectors. However, $L0$ norm is not differentiable and non-convex, so it is not computationally tractable. So in practice the $L0$ norm constraint is relaxed to $L1$ norm, which is the sum of absolution value of the coefficient matrix. So the $L1$ norm loss function, also called $Lasso$ loss function, is used to achieve the sparsity for both sparse learning and sparse coding, and they have similar mathematical formulations. A special type of regularization technique, namely $L1$ norm regularization, has been used to minimize the number of useful parameters in the model, so that only the most relevant features are extracted.

In machine learning, in order to prevent over-fitting, *i.e.*, the trained model fits the training samples well but fails to generalize, normally a regularization term is used to minimize the model coefficients. The regularization term is added to the loss function of the model and will be minimized during model optimization, thus penalizing big coefficients. Several regularization methods can be used, such as $L0$ norm regularization, which is the number of non-zero coefficients; or $L1$ norm regularization, which is the sum of the absolute value of all coefficients. Or $L2$ norm regularization, which is the sum of the square of all coefficients. Each regularization method has their own advantage and disadvantage. $L0$ norm regularization not only are used to prevent over-fitting; it also has the nice property that forces the selected features to be sparse [35].

To make a matrix sparse, most of the matrix elements have to be zeros, *i.e.*, the number of non-zero elements need to be minimized, which is equivalent to minimizing the $L0$ norm (defined as the total number of non-zeros terms). However, $L0$ norm minimization is non-convex and not computational tractable, fortunately, in most situations, it can be approximated by $L1$ norm minimization. The purpose of using sparse matrix for feature selection is that since most of the matrix members are zeros, only the most representative features are needed and used for classification. For dictionary learning where the purpose is to learn a set of dictionary entry call atom, the sparsity enforced that the number of dictionary entries are minimized, and this learned dictionary can re-construct the original data with minimal difference, thus maximize the reconstruction fidelity.

For both sparse learning and sparse coding, the objective function can be formulated in the following equation:

$$\min : similarity\_measure + \lambda\|\mathbf{X}\|_1 + additional\_constriant \tag{4.3}$$

Where the $similarity\_measure$ is the difference between predicted values and the labeled ground true in classification, or the difference between original and reconstructed signal (images) in dictionary learning or coding. $\lambda\|\mathbf{X}\|_1$ is the $L1$ norm regularization term to enforce sparsity, and $additional\_constriant$ imposes domain specific or prior knowledge constraint to the objective function.

The sparsity coding can be formulated as in equation (4.4), where the objective function is to minimize the re-construction loss:

$$\min : \frac{1}{2}\|\mathbf{A} - \mathbf{B}\mathbf{X}\|_F^2 + \lambda\|\mathbf{X}\|_{2,1} \tag{4.4}$$

4.3.2 Feature Extraction using Deep Neural Network

One of the biggest problem of using popular feature based classifier, such as logistic regression or support vector machines, is that they are linear classifiers; the input features need to be linearly separable the the high dimensional hyperspace in order to classify the samples. Even though techniques such as kernel method [36] can be used for transforming linear inseparable feature space into linear separable feature space, the choice of kernel function is difficult and often relies on heuristics. Also there is no guarantee that a suitable kernel function can be found, or the kernel function found is the best one.

Instead of hand engineering the features, in recent years, with the rapid development of deep learning (Science citation for Hitton), it has been popular to use multi-layer neural networks to automatically extract features from the raw data, and the extracted features are fed into a classifier, often a softmax or SVM, to classify the data. In medical image analysis, this method has the advantage of requiring less prior knowledge for the application developers; the model is often a generic, often pre-trained, convolutional neural network, and the expert's knowledge is embedded in the labeled training samples. The drawback of this method is that it often requires a lot more training samples, and the training time can be considerably longer than using the hand engineered features. So even this method has yielded many state-of-the-art results, it is not a panacea, and using traditional hand engineered featured or features learned from deep neural network depends on the specific requirement of the applications.

4.4 Computer Aided Endoscopy Diagnosis with Deep Sparse Feature Selection

A large number of people suffer from gastropathy around the world in human history [37]. In United States alone, every year there are about 24000 stomach cancer cases diagnosed, and about the health of over 4 million people are impacted by stomach ulcer. Early screening of these diseases can result in more effective treatment before the diseases getting more serious. However, early screening for a large patient population require a large number of physicians spending huge amount of time reading the endoscopy images, and this is not widely available to the general public due to the constraint of both doctors and medical facilities. Automating the process using computer vision technologies may provide a faster and less expensive solution. Though there are already many other computer vision based diagnostic systems for various type of medical image analysis, they are mainly in the X-ray image area that appeared as early as in 1990s [38]; comparing to the X-ray imaging, the gastroscopic images are in color and appear in much more complicated shape and texture, and the image quality is greatly impacted by the poor lighting condition in human organs, and no practical solution has been implemented in clinic use today. The gastropathy specialist diagnose the problem areas in the stomach mainly relying on the unique color and texture in these areas. One of the major technical difficulty for implementing computer vision solution for this is, due to the complexity of the endoscopy images, there are many visual features, both local and global, can be used for classification, but which features are the most representing and discriminating?

In this project, we focus on detecting various endoscopy lesions in esophagus and stomach with computer vision technology, and aim to design a computer aided diagnosis system to detect various esophagopathy and gastropathy abnormalities. Rather than making the final decisions to replace human experts such as pathologists and physicians, our algorithm is designed to be used as an early warning system to assist a medical experts and reduce their manual work amount, as well as improve the accuracy of medical diagnosis. As the passive WCE [39] with lower resolution are unable to precisely set the position and view angle to make a clearly observation in the internal organs, especially in stomach, it is only suitable in small intestine and colon areas, but not in large area in stomach. Therefore, we adopt the traditional gastroscope data, which has higher resolution and more flexible operation.

Many features can be obtained from the raw data, however, some features are more relevant to the classification results than the others. When using SVM as the classifier, in order to automatically select most discriminating features, $L1$ normalization is imposed on the weights, and that forces most coefficients to become zero, thus forcing the sparsity of the weight matrix. In this research, we apply this technique to a computer aided endoscopy diagnosis algorithm to automatically detect various abnormalities, such as ulcer, bleeding, cancer in oesophagus and stomach, from gastroscopic images . This research is collaborated with researchers at State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, China, and the author conducted algorithm development, coding, test, and analysis, and published as the title *Deep sparse feature selection for computer* *aided endoscopy diagnosis* [40] in *Pattern Recognition* journal.

4.4.1 Problem Statement

Optimized image representation is crucial for endoscopy image analysis, and various color and texture features have been designed, as shown in Fig. 4.1 (a). Each feature just represents one aspect of the local or global features, and no single feature can represent the entire image properly. Thus to better represent the image, multiple features are heuristically combined into one high-dimensional feature vector to complement each other, e.g., four kinds of features are combined in Fig. 4.1 (b). One other big problem is that the researchers do not have enough prior knowledge or domain knowledge to select the most representative features, and some useless features are also inevitably included into the feature vector, resulting decreased accuracy and increased computational complexity. To overcome these issues, some feature selection models [41] are designed to improve the quality of the feature vectors in terms of classification accuracy. For example [42] uses a neural network with feature selection to detect H.pylori infection, [43] designs a two-stage algorithm by first finding useful feature dimensions with sequential forward floating selection (SFFS) and then use the SVM to train a classifier accordingly, and [41] takes a step further and used the recursive feature elimination based on SVM (SVM-REF) for feature selection. Generally, the idea is to assign a greater weight to the more important feature dimensions, e.g. in Fig. 4.1(c), the deeper the color is, the greater the weight and the more important the corresponding feature dimension will be. However the problem is that the selected dimensions are always distributed in all feature types as in Fig.4.1(c), therefore we still need to extract all feature types in a time consuming way. More over, some noisy feature dimensions or units may be assigned with wrong weights.

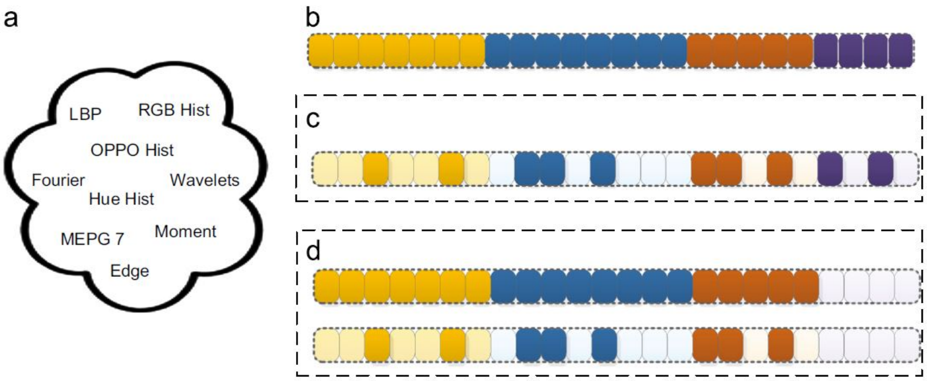


Figure 4.1: Illustration of Sparsity over Features Units and Feature Dimensions. (a) is all feature types considered (feature unit); (b) is the full feature vector (feature dimension) ; (c) is sparse feature selected that excluding non-discriminant feature dimensions; (d) is the result of deep group sparse feature selection, which excludes both useless feature units and useless feature dimensions

To overcome this, we can select the most relevant features and assign proper weights to the important feature dimensions simultaneously. In this way we not only improve the accuracy, but also reduce the computation complexity, because the useless feature units will not be extracted anymore. As shown in Fig. 4.1(d), the first three types of features are selected and the weights are assigned accordingly at the same time. To achieve this, we take into account the group sparsity of feature units and design a new model, i.e., Deep Sparse SVM (DSSVM).

4.4.2 Selecting Most Representative Features Using $L1$ Norm Regularization

After taking account of the group sparsity, not only the individual feature vector elements, or the so-called feature dimensions that not directly contributing to classification, is excluded from the feature vector, the type of features, or the so-called feature units, that are not directly contributing to the classification, is also entirely excluded. This will greatly reduces the model and computational complexity.

Since each endoscopy image represents a large area inside the stomach, not only we need to classify if the patient is healthy or non-healthy, we also need to locate the

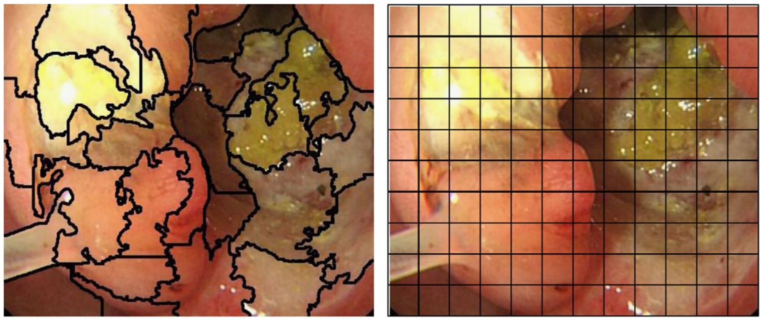


Figure 4.2: Super-pixel based feature extraction vs. patch based feature extraction

problematic area. So one important concern in algorithm development is how to divide the original image into smaller parts. Most current algorithms extract features from rectangular shaped sub-image patches. If the patch size is too small, it will not contain enough information; if the patch size is too big, it may contain information that not contributing to the discriminant characteristics, i.e., too many disturbed pixels. So determining the optimized patch size is difficult. Also, the problematic areas are usually not squared shaped; they are irregular shape with their own color and texture properties. Thus, we adopt the superpixel method to segment the medical images in a more flexible and adaptive way. Additionally, since the image quality is often impacted by lighting conditions and internal structure of the human organs, we need to assess the image quality and discard regions with poor image quality. The super-pixel based segmentation is illustration in figure 4.2, where the SLIC algorithm [44] is used to segment the image into superpixels.

The process pipeline of the algorithm can be illustrated in figure 4.3.

In our approach, to obtain the optimal feature representation, the image is first segmented into superpixels, which fits the natural texture pattern of the human tissue than the square patch-based methods, and several color and texture feature units are combined to create one feature vector. We designed a new feature selection model with group sparsity, called Deep Sparse SVM(DSSVM) by using $L1$ normalization,

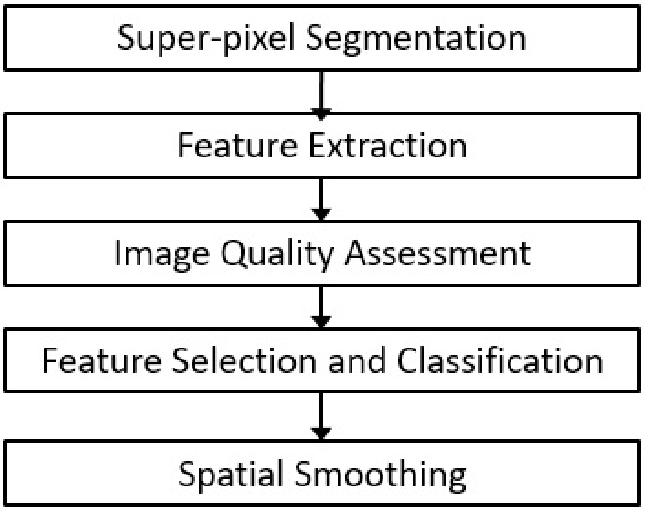


Figure 4.3: Processing pipeline for computer aided endoscopy image diagnosis

which overcomes a common difficulty of most of the state-of-the-art algorithms, where though they can select useful feature dimensions, they always need to extract all features, including the useless feature types. Therefore, our DSSVM model can reduce computational time and improve the robustness of the algorithm.

4.4.3 Deep Sparse Support Vector Machines (DSSVM)

The computer aided endoscopic diagnosis can be formulated as a supervised classification problem, i.e., first train the generic model with labeled training samples (ground true) and tune the model parameters, which is called the training phase, and the determine whether each sample is normal or abnormal by the trained model, which is called the inference phase. To implement this, we first collect and label the training data as $\mathcal{D} = \{(\mathbf{x}_i, y_i)|\mathbf{x}_i \in R^d, y_i \in \{-1, 1\}\}_{i=1}^n$, where n is the size of training data set; $\mathbf{x}_i$ is a d-dimensional feature vector represented the sample; and y_i is either -1 or 1 indicating the class to which the point $\mathbf{x}_i$ belongs, i.e. normal or abnormal. Usually for medical image analysis, $\mathbf{x}$ is combined by several independent feature units, e.g. various color and texture features, $\mathbf{x} = [\mathbf{x}'_1, \mathbf{x}'_2, \ldots, \mathbf{x}'_K], \mathbf{x}'_k \in R^{d_k}, d = \sum_{k=1}^{K} d_k$ and K is the number of feature units. Traditionally, the Support Vector Machine (SVM) model with L_2 norm [45] is used for classification by estimating a decision hyper-plane and maximize the margin between the data points that representing two classes:

$$L_2\text{SVM:} \min_{\mathbf{w}\in\mathbb{R}^n,\ b\in\mathbb{R}} : \sum_{i=1}^{n} h(\mathbf{w}, b, \mathbf{x}_i, y_i) + \lambda\|\mathbf{w}\|_2^2, \tag{4.5}$$

where $\mathbf{w} \in R^d$ and $b \in R$ is the parameters of the decision hyper-plane, λ is the model parameter tuned with training samples. The first term in the equation is to minimize the overall reconstruction error, and $h(\mathbf{w}, b, \mathbf{x}_i, y_i)$ denotes the hinge loss function, which is defined as $h(\mathbf{w}, b, \mathbf{x}_i, y_i) = \max\{1 - y_i(\mathbf{w}^T\mathbf{x}_i + b), 0\}$.

Unfortunately, in practice we do not always have enough prior knowledge to identify the most discriminant feature units $\mathbf{x}'_k$ that have greatest impact on classification result, and then combine them into feature vectors. Redundant or noisy features are inevitably included in the feature set, and that will not only increase the computation time, but also deteriorate the classification accuracy. We call these features disturbing feature units and disturbing feature dimensions. Therefore, if we could only select the most useful features, the classification performance can be improved. In theory, the value of $\mathbf{w}_i$ is the weight imposed on each element in the overall feature vector, and it can be considered as the importance of the corresponding feature dimension $\mathbf{x}_i$ for feature selection, i.e., the greater the value of $\mathbf{w}_i$ is, the more important the feature $\mathbf{x}_i$ is for classification, so $\mathbf{w}_i = 0$ means $\mathbf{x}_i$ plays no roles; in another words, they should be removed from the feature vector.

However, the $\mathbf{x}$ generated by traditional L_2 SVM model in Eq. (4.5) is usually dense, i.e. nearly all the feature dimensions are selected and the usability of feature selection is lost accordingly. Thus, to pursue a sparse feature set where most feature dimensions are zero, we consider the L_1 SVM [46] model:

$$L_1\text{SVM:} \min_{\mathbf{w}\in\mathbb{R}^n,\ b\in\mathbb{R}} : \sum_{i=1}^{n} h(\mathbf{w}, b, \mathbf{x}_i, y_i) + \lambda\|\mathbf{w}\|_1, \tag{4.6}$$

where $\|\mathbf{w}\|_1 = \sum_{i=1}^{d} |\mathbf{w}_i|$ is the convex envelope of the cardinality function of $\mathbf{w}$ used to pursue a sparse solution. Eq. (4.6) can make a more sparse result of $\mathbf{w}$, i.e., the more effective feature dimensions $\mathbf{x}_i$ are given more weight, and the feature dimensions of $\mathbf{x}_i$ that not contributing to the discriminant power are ignored and eliminated. Some feature types that are originally included in the feature vector may not contributing to minimizing the cost function and not contribution to the performance of the classifier, so we want to exclude the feature type (or so-called feature unit) altogether. However, the original feature vector contains all feature types, and the selected feature dimensions may distribute on all the feature units. Therefore we still need to extract all the feature units and some noisy feature dimensions/units may be assigned with wrong weights, which is time consuming and will negatively impact the robustness of the algorithm.

To overcome this, our approach is that if we can select the useful feature units and assign a greater weight to the more important dimensions of the selected feature units, and we do not need to extract the discarded feature units. By considering the group sparsity of feature unit set, we define the following model:

Then by considering both the $L1$ regularization term and group sparsity term, the objective function of our model can be defined as:

$$\text{DSSVM:} \min_{\mathbf{w}\in\mathbb{R}^n,\ b\in\mathbb{R}} : \sum_{i=1}^{N} h(\mathbf{w}, b, \mathbf{x}_i, y_i) + \lambda_1\|\mathbf{w}\|_1 + \lambda_2\|\mathbf{w}\|_G, \tag{4.7}$$

where the third item is the group sparsity function defined as $\|\mathbf{w}\|_G = \sum_{i=1}^{K} \|\mathbf{w}_{g_i}\|$, and $G = \{g_i :\ i = 1, \cdots, K\}$ is a set of index sets used to select the useful features

g_i from the whole feature set G. The main advantage of our model is that it can select the useful feature units and feature dimensions concurrently.

4.4.4 Model Training

There are multiple non-smooth terms in Eq. (4.9), so the traditional gradient descend methods can not be used directly for model training/optimization. To solve this problem, the alternative directional multiplier method (ADMM) is employed to solve Eq. (4.9). The alternating direction method of multipliers (ADMM) is an algorithm that solves convex optimization problems by breaking them into smaller pieces, each of which are potentially differentiable, thus easier to handle. We first denote $\mathbf{z}_i = y_i\mathbf{x}_i$ and construct $\mathbf{Z} = [\mathbf{z}_1, \cdots, \mathbf{z}_n]$ and $\mathbf{Y} = [y_1, \cdots, y_n]^T$. By introducing the following substitution:

$$\mathbf{u} = \mathbf{Z}^T\mathbf{w} + \mathbf{Y}b - \mathbf{1}_n,\ \mathbf{v} = \mathbf{w}, \tag{4.8}$$

we can rewrite the problem in Eq. (4.9) as

$$\begin{aligned} \min_{\mathbf{w},b} :\ & \sum_{i=1}^{n} \max\{\mathbf{u}_i, 0\} + \lambda_1\|\mathbf{w}\| + \lambda_2\|\mathbf{v}\|_G \\ s.t. :\ & \mathbf{v} = \mathbf{w} \\ & \mathbf{u} = \mathbf{Z}^T\mathbf{w} + \mathbf{Y}b - \mathbf{1}_n. \end{aligned} \tag{4.9}$$

Then, the Augmented Lagrangian function can be defined as

$$\begin{aligned} L_\rho(\mathbf{w}, b, \mathbf{u}, \mathbf{v}, \alpha, \beta) = & \sum_{i=1}^{n} \max\{\mathbf{u}_i, 0\} + \lambda_1\|\mathbf{w}\|_1 + \lambda_2\|\mathbf{v}\|_G \\ & + \langle \alpha, \mathbf{v} - \mathbf{w} \rangle + \langle \beta, \mathbf{u} + \mathbf{1}_n - \mathbf{Z}^T\mathbf{w} - \mathbf{Y}b \rangle \\ & + \frac{\rho}{2}\|\mathbf{v} - \mathbf{w}\|^2 + \frac{\rho}{2}\|\mathbf{u} + \mathbf{1}_n - \mathbf{Z}^T\mathbf{w} - \mathbf{Y}b\|^2. \end{aligned} \tag{4.10}$$

To solve Eq. (4.10), we apply ADMM methods and obtain the procedure in ADMM algorithm. Step 1 in ADMM is equivalent to solve the following problem:

$$\min_{\mathbf{u}} : \frac{1}{2}\|\mathbf{u} - \mathbf{t}\|^2 + \lambda \sum_{i=1}^{n} \max\{\mathbf{u}_i, 0\}. \tag{4.11}$$

One can see that the closed form is

$$\mathbf{u}^* = \begin{cases} \mathbf{t}, & \mathbf{t} \leq 0 \\ 0, & 0 < \mathbf{t}_i < \lambda \\ \mathbf{t}_i - \lambda, & \mathbf{t}_i \geq \lambda \end{cases} \tag{4.12}$$

Step 2 in ADMM algorithm is equivalent to solve the following equation:

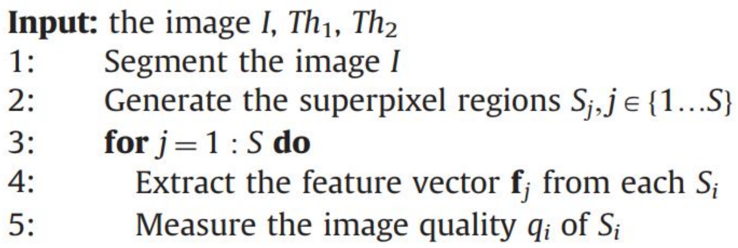
$$\min_{\mathbf{v}} : \frac{1}{2}\|\mathbf{v}_{g_i} - \mathbf{t}\|^2 + \lambda\|\mathbf{v}_{g_i}\|. \tag{4.13}$$

The solution to this problem is

$$\mathbf{v}^*_{g_i} = \begin{cases} (1 - \lambda/\|\mathbf{t}\|)\mathbf{t}, & \|\mathbf{t}\| \geq \lambda \\ 0, & otherwise \end{cases} \tag{4.14}$$

This step involves in solving a LASSO formulation. One can use any solver to solve it. The closed form solution of this step can be easily obtained by solving a least square minimization problem. Therefore, following the ADMM algorithm, we can optimize our model in Eq. (4.9) properly.

The algorithm can be illustrated as follows 4.4:



```
Input: the image $I$, $Th_1$, $Th_2$
    Segment the image $I$
    Generate the superpixel regions $S_j, j \in \{1 \ldots S\}$
    for $j = 1 : S$ do
        Extract the feature vector $\mathbf{f}_j$ from each $S_i$
        Measure the image quality $q_i$ of $S_i$
```

Figure 4.4: General Framework for Group Sparsity Algorithm

4.4.5 Results and Analysis

Our algorithm is evaluated against L2-SVM, L1-SVM, Group-SVM. In this experiment, 7 color features, HSI Intensity histogram (15d), HSV-HV histogram (30d), Hue histogram (15d), Opponent RGB histogram (45d), Normalized RGB histogram (45d), RG histogram (30d), and RGB histogram (45d) are included in the feature vector. In addition, 2 texture features, statistic LBP (6d) and LBP histogram (15d) are also used.

The comparison results are shown in Table 4.2.

	L2-SVM	L1-SVM	Group-SVM	**DSSVM**
feature unit/feature dimension	9/176d	9/96d	8/165d	**6/84d**

Table 4.2: Comparison of the number of feature units and feature dimension selected by L2-SVM, L1-SVM, Group-SVM and Deep Sparse SVM

The table clearly demonstrates that our Deep Sparse SVM select the most sparse feature unit as well as feature dimension among the 4 methods.

In this research, a computer aided endoscopy diagnosis method was developed to automatically detect various abnormalities, such as ulcer, bleeding, cancer in esophagus and stomach, from gastroscopic images. For optimal feature representation, the image is first segmented into superpixels in a more flexible way than the popular patch-based methods, and several color and texture feature units are combined to create one feature vector. Our main contribution is to develop a new feature selection algorithm with group sparsity, Deep Sparse SVM (DSSVM), which overcomes a common difficulty that although most of the state-of-the-art algorithms can select useful feature dimensions, they always need to extract all feature units or types. Therefore, our DSSVM model can reduce computational complexity and improve the robustness accordingly. For experiments, to our best knowledge, we build the largest endoscopy dataset with pixel-level and frame-level ground truth. In our experiment, our DSSVM model can obtain a slightly better accuracy using fewer feature units and feature dimensions than the state-of-the-arts; in addition, when adopting the same number of feature dimensions, ours outperforms other feature selection models with a big margin, which demonstrates the effectiveness and efficiency of our algorithm.

4.5 Sparse Dictionary Learning with Similar-inhibition Constraint and Attention Prior

Video summarization, or key frame extraction of video, is to extract the most informative frames from a video. This is a very important technology used for video indexing and video retrieval, popularly used by search engines. Medical videos have some unique properties compared to more structured or staged videos such as a TV show, and they post unique challenges summering them. This research project is a collaboration with researchers Shuai Wang, Yang Cong, etc. from Shenyang Institute of Automation, Chinese Academy of Sciences, and the author participated in algorithm development, coding, and analysis. This research work has been published as the title *Scalable gastroscopic video summarization via similar-inhibition dictionary selection* in [47].

A video clip consists of a series of images call frames. These frames neither independent nor appear randomly; they are correlated in the temporal axis, and many useful information can be obtained from the video. In video processing, one important application is video summarization; i.e., a minimum set of key frames are selected to represent the key information of the original video clip. In this research, we demonstrates the use of dictionary learning and sparse learning to obtain the key frames and summarize the video. In this research, an automated gastroscopic video summarization algorithm for assist clinicians to spot the abnormal contents of the video more efficiently is developed, which is essentially a key frame selection problem. [47]

4.5.1 Dictionary Selection Algorithm

In image processing, an image can be reconstructed or restored with a linear combination of a set of image patches call visual words [48]. A good dictionary is a trade-off between minimizing the reconstruction error and dictionary size. Bigger dictionary tends to yield more accurate reconstruction, while smaller dictionary will decrease reconstruction complexity. An optimized dictionary is a balance between the two factors, subject to the constraints. To generate an optimized dictionary, the most representative and informative frames from the original video sequence need to be selected. Thus, the problem of gastroscopic video summarization is formulated as a dictionary selection issue, i.e., selecting an optimal subset from the original video frames via dictionary learning under various constraints that can be used to best reconstruct the original frames. In order to minimize the dictionary size, $L1$ norm regularization term is introduced to force the sparsity of the dictionary.

In addition, the selected frames should be as diverse as possible, thus the new frames will contain as much new information as possible. The similar-inhibition constraint is introduced in our model to reinforce the diversity of selected key frames. Also, prior information, which representing some expert knowledge not represented in the training sample it self, also needs to be incorporated into the cost function. Normally, when the physician takes video, the longer the physician stays on one area, the higher interest we can consider he/she has for this areas. This prior is called attention cost, and we want to incorporate this expert knowledge into our model. The attention cost is evaluated by merging both gaze and content change into *a prior* cue to help select the frames with more high-level semantic information. Moreover, we adopt an image quality evaluation process to eliminate the imapct of the poor quality images and a segmentation process to reduce the computational complexity. The process flow for video summarization and key frame selection is illustrated in Figure 4.5.

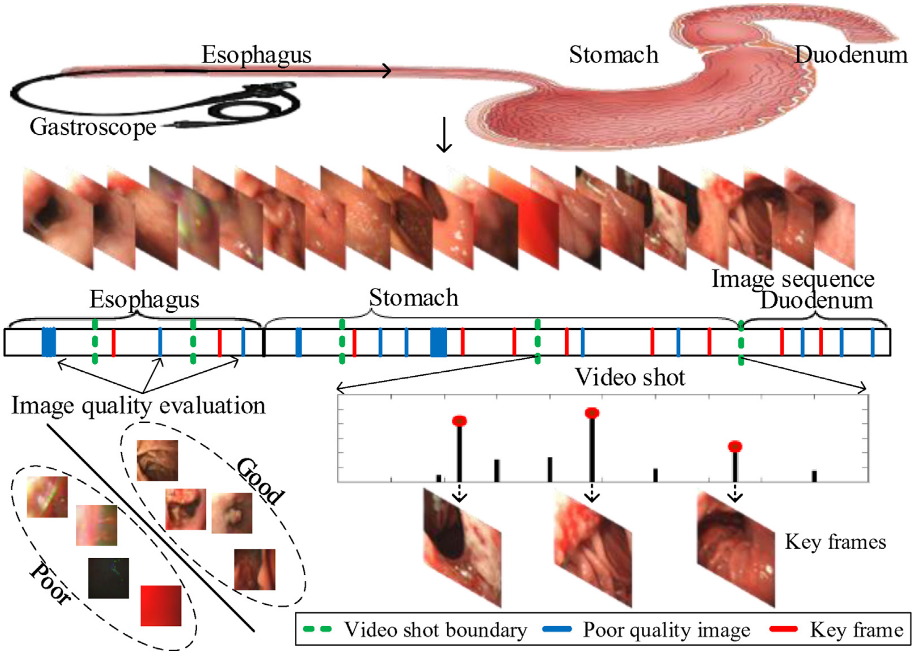


Figure 4.5: Video Summary for Gastroscope Video

4.5.2 Similar-inhibition dictionary selection model

For humans, finding interesting moments from unstructured video such as surveillance video or medical procedure video usually requires people to go through the entire video clip, which is a very time consuming procedure. Video summarization can identify a set of most important frames from a video clip. The goal of video summarization is to select the most representative frames from the underlying video source that represent the video contents properly. However, the definition of ”most important” is a subjective term and could vary from person to person, thus bringing the need for an objective criteria for key frame selection. In this research, we formulate the problem of gastroscopic video summarization as a dictionary selection issue, i.e., to select an optimal subset from the original video frames via dictionary learning under various constraints. The video sequence can be represented as an initial dictionary $B = [b_1, b_2, ..., b_N] \in \mathbb{R}^{d \times N}$, where N is the number of frames and d is the feature dimension, and each column vector $b_i \in \mathbb{R}^d$ denotes a video frame representing a feature vector. In the traditional dictionary selection based method, the reconstruction error, which measures the discrepancy between the original frame and the reconstructed frame, should be minimized, so that the reconstructed frame can be as close to the original frame as possible. If we denote the matrix representing the original image $\mathbf{A}$, and $\mathbf{X}$ is the coefficients to linearly combine the elements in dictionary b to reconstruct the original frame $\mathbf{A}$, then the difference between two frames is $\mathbf{A} - \mathbf{XB}$, and its magnitude is normally measured by the Frobenius norm of the matrix, which is defined as $\|X\|_F = (\sum_{i,j} X_{ij}^2)^{1/2}$. The reconstruction error is then defined as $\|\mathbf{A} - \mathbf{BX}\|_F$.

The primary objective function of the reconstruction is to minimize the reconstruction error. However, to prevent over-fitting where the model fails to generalize, a regularization term λ is introduced, which tries to minimize the value of the weight matrix. So with the introduction of $L1$ regularization term, the objective function becomes:

$$\min : \frac{1}{2}\|\mathbf{A} - \mathbf{BX}\|_F^2 + \lambda\|\mathbf{X}\|_{2,1} \tag{4.15}$$

The $L1$ norm regularization is used to force the weight matrix be sparse, i.e., to force most of the elements in the matrix to be zeros. Not only it prevents over-fitting, this also enforces that only the most representative elements will be non-zero. In this case, since a matrix is a 2 dimensional vector, we can use its Frobenious norm, which is defined as $\|X\|_F = (\sum_{i,j} X_{ij}^2)^{1/2}$, and $l_{2,1}$ is defined as $\|X\|_{2,1} = \sum_{i=1}^N \|X_i\|_2$; it is essentially the two dimensional equivalent of the $L1$ norm in one dimensional space, since if x is a one dimensional vector, then $\|X\|_{2,1} = \|X\|_1$.

The sparsity constraint enforces the number of selected key frames are minimized, but it does not prevent similar frames being selected, and similar frames do not provide much additional information. To force each selected frame being as informative as possible, in addition to the $L1$ normalization term that enforces the sparsity constraints, we also need to incorporate a term that enforces the diversity of the key frames selected; i.e., we penalize the similarity between selected key frames, and make the key frames as different from each other as possible. We denote X_i as the ith frame in the video sequence, the similarity of two frames can be formulated as $\|X_i\|_2 S_{ij} \|X_j\|_2$, where $S = \{S_{ij}\} \in \mathbb{R}^{N \times N}$ is the matrix that correlates the 2 frames. We call this the self-inhibition constraint. So the new objective function becomes:

$$\min : \frac{1}{2}\|\mathbf{A} - \mathbf{B}\mathbf{X}\|_F^2 + \lambda\|\mathbf{X}\|_{2,1} \tag{4.16}$$

With the introduction of the additional self-inhibition constraint term, the objective function evolves into:

$$\min : \frac{1}{2}\|\mathbf{A} - \mathbf{B}\mathbf{X}\|_F^2 + \lambda\|\mathbf{X}\|_{2,1} + \beta \sum_{i,j=1}^{N} \|X_i\|_2 S_{ij} \|X_j\|_2 \tag{4.17}$$

where the third term is the similar-inhibition constraint, which is introduced to reinforce the diversity and penalize the situation that the rows of two similar elements are nonzero at the same time. For each sample pair (bi, bj), if b_i is similar to b_j, then one should assign a larger weight to S_{ij}, which helps prevent choosing two similar frames as key frames. Otherwise, S_{ij} should be set to a very small value, which has a negligible impact on choosing both b_i and b_j as key frames. The weight matrix used in our research is defined as:

$$S_{ij} = e^{-\frac{\|b_i - b_j\|^2}{\sigma_s^2}}, i, j = 1, ...N \tag{4.18}$$

Where σ_s is a constant, with $\sigma_s^2 = 24$ set empirically in this research. From this equation, we can see the larger S_{ij} value is, the more similar two frames b_i and b_j are.

4.5.3 Incorporation of the Prior Knowledge

Prior knowledge generally refers to the knowledge that is not embedded in the training samples, such as domain knowledge. For the case of gastroscopic video, the physicians tend to focus on the abnormal lesions more than the other parts during the operation procedure, since it is what their interest lies on, and that is where the diagnosis will be based on, which reflects the human psychological preferences, i.e., their attention. Therefore, we take both gaze and content change information as the attention prior knowledge to reduce the gap between low-level features and high-level concepts, so the result of summarization reflects the physicians actual preferences and original intention better. This prior knowledge contains the medical knowledge of the physician, and the algorithm needs to capture this medical expertise through the gaze and content change constraints.

a. Gaze prior: The human eye is usually attracted by the salient regions of a significant color distribution or strong contrast in the frame [23]. The locations of these salient regions indicate whether the corresponding frames can provide sufficient contents and better observation angles for clinicians to make the final diagnosis. For frames with similar contents, the closer the salient region is to the center of the corresponding frame, the more important the corresponding frame is and the large probability that it is selected as a key frame. Thus, to model gaze, we first obtain the saliency map via the method proposed and then compute the Euclidean distance of the saliency map centroid to the frame center. Finally, the gaze attention score for each frame is defined as the following equation:

$$M_g^i = 1 - \|C_i - O_i\|_2 \tag{4.19}$$

where C_i is the normalized coordinate of the saliency map centroid and Oi is the normalized coordinate of the frame center, with i as the frame index. A large $M_g^i \in [0, 1]$ indicates more attention. Then two frames which are close in the temporal sequence and they show the same lesion region under different observation angles, respectively. We can see that the frame with a better observation angle can obtain a large gaze attention score compared with the frame with similar lesion content. This indicates that the gaze attention prior provides a reasonable value to evaluate the importance of frame content.

b. Content change prior: In general, rapid video content change usually implies the physician does not find this part of video interesting so they do not pay much attention to it, thus the camera is moved quickly at these areas, and this means low attention or content with little interest, which should be excluded from the key frame set.

Rapid content change between frames indicates that the corresponding frames have limited ability to reconstruct other frames, which is consistent with the reconstruction error. Moreover, rapid content change has a great negative impact on the quality of images. To model content change, we estimate the optical flow [49] for each frame and then compute the mean amplitude u_i of optical flow in each frame. The content change attention score for each frame is defined as the following equation:

$$M_c^i = 1 - \frac{\overline{u_i}}{\max_{j=1...N} \overline{u_j}} \tag{4.20}$$

where N denotes the number of frames, $\overline{u_j}$ represents the mean amplitude of the motion field (optical flow value) of frame j, and $M_c^i \in [0, 1]$. The larger the value of M_i^c is, the more attention this frame gets from the physician. To model the attention prior, linear fusion schemes are used to fuse various attention scores in order to generate an aggregated attention score [50]. The general formation of linear fusion schemes is defined in the following equation:

$$M = \sum_{j=1}^{n} \lambda_i M_j \tag{4.21}$$

where $\sum_{j=1}^{n} \lambda_i = 1$.

λ_j is the weight of the attention value M_j that reflects the relative importance among $M_j(j = 1...n)$, and N is the number of various attention scores. The weight λ_j of the attention value M_j is determined as:

$$\lambda_j = \frac{\sigma_j}{\sum_{k=1}^{n} \sigma_k} \tag{4.22}$$

where σ_j indicates the standard deviation of the attention score M_j. The final attention prior score is the weighted sum of the above two factors, as shown in the following formula:

$$M^i = \lambda_g M_g^i + \lambda_c M_c^i \tag{4.23}$$

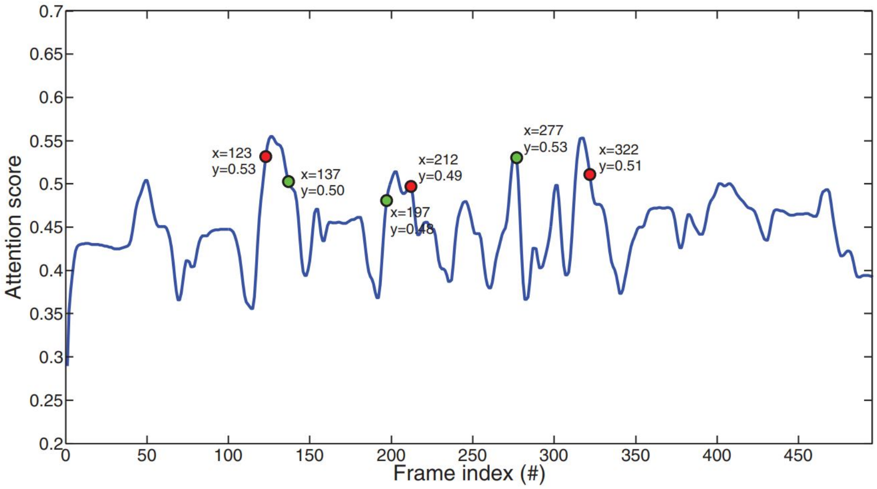
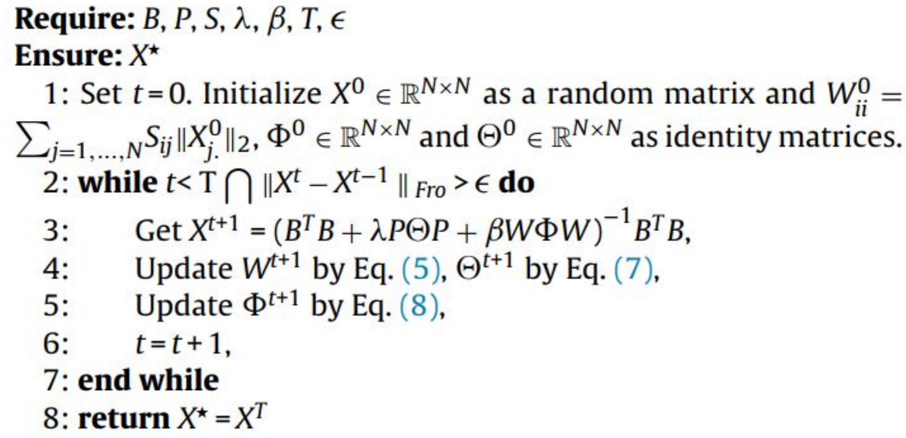


Figure 4.6: An attention prior curve within a shot. The red points denote the locations of the ground truth and the green points denote the locations of key frames selected by our method. x is the frame index and y is the attention score of the corresponding points



```
Require: $B, P, S, \lambda, \beta, T, \epsilon$
Ensure: $X^\star$
  Set $t=0$. Initialize $X^0 \in \mathbb{R}^{N\times N}$ as a random matrix and $W^0_{ii} = \sum_{j=1,\ldots,N} S_{ij}\|X^0_{j.}\|_2$, $\Phi^0 \in \mathbb{R}^{N\times N}$ and $\Theta^0 \in \mathbb{R}^{N\times N}$ as identity matrices.
  while $t<$ T $\bigcap \|X^t - X^{t-1}\|_{Fro} > \epsilon$ do
        Get $X^{t+1} = (B^T B + \lambda P \Theta P + \beta W \Phi W)^{-1} B^T B$,
        Update $W^{t+1}$ by Eq. (5), $\Theta^{t+1}$ by Eq. (7),
        Update $\Phi^{t+1}$ by Eq. (8),
        $t = t + 1$,
  end while
  return $X^\star = X^T$
```

Figure 4.7: Optimization of the model

The attention *a prior* is illustrated in 4.6. There are three key frames selected within this video clip, the frames numbered 123, 212, and 322. The ground truth is the selected key frames, which have a relatively large attention score.

The algorithm for model optimization is shown as:

4.5.4 Experiment and Analysis

In this section, a new gastroscopic video summarization dataset was built and it validates our method by comparing it with the state-of-the-arts. The video summarization algorithms can be roughly divided into three categories [51], i.e., sequential algorithms, clustering-based algorithms and optimization-based algorithms. To evaluate the performance of our algorithm, several algorithms from each category are selected to compare with our algorithm, including evenly spaced key frames (ESKF) [52], clustering algorithms K-means based method [53] and k-medoids-based method [54], and dictionary selection based video summarization (DSVS) algorithm [55]. For the sake of fairness, each method used in comparison selects the same number of key frames as the ground truth in our experiments.

Dataset Preparation

A gastroscopic video dataset captured from 30 volunteers with more than 400k images are built, and our method is compared with the state-of-the-arts using the content consistency, index consistency and content-index consistency with the ground truth. Compared with all competitors, our method obtains the best results in 23 of 30 videos evaluated based on content consistency, 24 of 30 videos evaluated based on index consistency, and all videos evaluated based on content-index consistency.

For stroboscopic video summarization, we propose an automated annotation method via similar-inhibition dictionary selection. Our model can achieve better performance compared with other state-of-the-art models and supplies more suitable key frames for diagnosis. The developed algorithm can be automatically adapted to various real applications, such as the training of young clinicians, computer-aided diagnosis or medical report generation, as shown in the paper [47]

Content Consistency

To evaluate the content similarity between the selected key frames and the corresponding ground truth, we define the content consistency score (CCS) as following:

$$CCS = \frac{\Sigma_{k=1}^{K} \delta_c(e_k, g_k)}{K} \tag{4.24}$$

Index Consistency

We also verify the frame index consistency between the selected key frames and the ground truth [65 xxx]. Based on the frame index distance (the difference in frame number), we assess each selected key frame in 4 categories: better matching, good matching, general matching, and bad matching, with the quantified scores assigned for each as: 3, 2, 1, 0. The detailed definition of the index consistency score (ICS) can be described as:

$$ICS = \frac{\Sigma_{k=1}^{K} \delta_i(e_k, g_k)}{K} \tag{4.25}$$

Content-index Consistency

Finally, both image content and time differences are considered for the performance. For comparison purpose, two frames are considered as matching each other only if they are similar in scene content and occur within a short period. We named this criterion the content index consistency score (CICS), which is defined as:

$$CICS = \frac{\Sigma_{k=1}^{K} \delta(e_k, g_k)}{K} \tag{4.26}$$

The content index consistency score test results are shown in Table 4.3:

	ESKF	K-means	k-medoids	DSVS	**SIDSVS**
Average	0.311	0.357	0.341	0.340	**0.626**

Table 4.3: Comparison of content index consistency scores obtained by ESKF, K-means, k-medoids, DSVS, SIDSVS

The worst average result is from ESKF (0.311), which just selects evenly spaced video frames without considering its content. The results of the k-means-based method, k-medoids-based method and DSVS are very similar to each other (0.357 vs. 0.341 vs. 0.340); this can be explained by that these methods are very similar clustering algorithm based on low-level image features and neglect the semantic value of the data. From Table 4.3, it can be seen that SIDSVS developed in this research outperforms all other methods with an average CICS = 0.626. Specifically, we obtain near perfect results with large lead in accuracy to other methods on all videos. This can largely attribute to the similar inhibition constraint that makes the selected key frames more diverse in terms of content, and the attention prior that provides a method to quantitatively evaluate the relative importance of each frame. Using the three terms in the reconstruction loss function, i.e., the reconstruction error, group sparsity constraint and similar-inhibition constraint, our SIDSVS model provides best results with other comparable algorithms and in consistency with the ground truth marked by the physicians.

Conclusion

In this research, we enhanced the sparse dictionary learning algorithm to summarize the content of the gastroscopic video in the semantically meaningful way to better navigate gastroscopic video content for diagnosis and future reference. A new algorithm of gastroscopic video summarization has been proposed in this research by utilizing the similar-inhibition constraint and attention prior. By representing each video frame as a feature vector, the gastroscopic video summarization problem is converted into a sparse dictionary selection problem under three constraints, reconstruction error, group sparsity, and similar inhibition. And the attention score is computed by merging two cues, i.e., gaze and content change, and then it is added into the loss function of the model as a prior cue. This method not only provides a configurable solution to video summarization, which allows us to select arbitrary number of key frames, but it also can reinforce the diversity of the selected key frames, and can be generalized for use in similar usage scenarios such as summarizing the content of the dashboard camera of an autonomous vehicle or a surveillance camera logs. However, because the self-inhibition term is non-convex, the result is a local minimum and may or may not be globally optimized, thus it may fail to generalize to other dataset; its validity needs to be tested with broader dataset when try to generalize its use. The preliminary results obtained on the new gastroscopic video dataset validate that this method outperforms other state-ofthe-art video summarization models.

4.6 Medical Image Feature Extraction using ConvNet

The objects in medical images, such as organs, tissues, and cells in the MRI, CAT and PET scan, ultra-sound images, etcs, have very different visual appearance from everyday objects in terms of their shape, color, and texture, since they often do not have a rigid shape or consistent color. It is very difficult to hand engineer the best discriminant features as illustrated in the last two sections. With the advancement in deep neural network technology in recent years, especially the convolutional neural network (ConvNet, or CNN), the current state-of-the-art in the research community is to use ConvNet as the feature extractor to automatically extract the most discriminant an representative features [56]. The success of using deep learning for medical image analysis has been demonstrated in many recent cases, as one of the most cited one is the *Nature* paper *Dermatologist-level classification of skin cancer with deep neural networks* published on January 2017 [57], where it is demonstrated that the diagnosis provided by a deep neural network outperformed the best medical doctors in skin cancer diagnosis.

As an evidence that deep learning based solutions have become the mainstream algorithms for medical image analysis, at least 80% of the published papers in MICCAI (MICCAI 2018, the 21st International Conference on Medical Image Computing and Computer Assisted Intervention,) 2018 involve using ConvNet; hand engineering features are almost all replaced with features automatically extracted by deep neural network based algorithms in medical image analysis area nowadays. Generative algorithms such as GAN (Generative Adversarial Network) has also been widely adopted for tasks [58] such as super resolution, where a 10 minutes procedure of low resolution CAT scan can be up-resoluted to one that would have taken 40 minutes, thus reduce the time the patients are required to be exposed to radiation. In our latest research, convolutional neural network is used as feature extractor to extract features for Follicular Lymphoma grading and white blood cell classification.

4.6.1 Grading Follicular Lymphoma Grading

Follicular lymphoma (FL for short) is a type of blood cancer, and their treatment is depending on the severity, grade, of the disease. This work is collaborated with Medical Doctor Kai Fu from University of Nebraska Medical Center and Medical Doctor Qinglong Hu from Tuscon Pathology Associates, who provided the labeled dataset as well as the medical background knowledge. The author performed algorithm development, coding, testing, and analysis. The result is reported as a poster presentation at 2018 *The United States and Canadian Academy of Pathology Annual Conference*.

The diagnosis of many types of cancers relies on pathologists inspecting the tissue samples under microscope. Due to the irregular shape of the tissue, a learning based solution is a natural choice since it is difficult to come up with rules that defining the visual appearance of the tissue, organ, or cells. However, the image size in terms of pixels of the training samples of typical convolutional neural network is rather small, such as 128×128 or 256×256, or even smaller, and this is due to the limitation of the computer hardware, such as memory and CPU. However, high resolution microscopic tissue images are over 1 megapixels or even more, and whole slide scanning images can even be as big as $100,000 \times 100,000$. Thus these images can not be directly used as the input data to the deep neural networks. Simply down-sampling images does not work, since it will lose details internal to the cells. In this research, the tissue level image is divided into smaller patches, and the patches are used to train a convolutional neural network for classification. To classify an image, this image is divided into patches of the sample size, and the final result is decided by majority voting; i.e., if the image is divided into N patches, and these patches are classified into M categories, the category that most patches are classified into will be designated as the category for the whole image. This algorithm is very effective and without the need for expensive segmentation, and can be used for many types of tissue samples where the tissue texture is uniformed distributed, and we use the tissue level grading of Follicular Lymphoma to demonstrate this algorithm.

Another issue of training models for medical image classification is the number of labeled training samples are very limited, since each sample has to be labeled by medical experts. By some estimates, the number of training samples for classifying medical images using convolutional neural network is around $5,000$. By dividing an image into multiple patches and each patch is used as an independent training sample, the number of training samples also increases.

Follicular Lymphoma (FL) is the second most common non-Hodgkins lymphoma [59], and its grading is essential for the proper treatment for FL patients; according to World Health Organization (WHO), FL is graded into 4 grades according to its severity. Among them, grade 3A and 3B are high risk and require more aggressive chemotherapy while grade 1 and 2 are considered as indolent and normally only require conservative treatment. In the WHO histological grading procedure, FL severity is graded by counting the number of larger cancer cells called centroblasts (CB) in a high power field (HPF, defined as 0.159 mm2).

According to [60], the severity of FL can be graded by the number of CBs in each HPE as follows: *Grade 1:* 05 centroblasts (CBs) per HPF; *Grade 2:* 615 centroblasts per HPF; *Grade 3:* More than 15 centroblasts per HPF.

However, The WHO cell counting based procedure has several practical problems. First, the lymph cells under consideration exist in follicles; some follicles can be smaller than the area of a HPF, thus the number of CBs can be smaller in this case than in other cases where follicles fill the entire HPF. Also this procedure is a tedious manual task, and is prone to sampling bias as well. There are already algorithms that grade the FL severity by cell counting [59], but they are not used widely in practice. In most clinic practice, instead of counting individual cells, the pathologist just look at the tissue image in its entirety, and determine the grade of the FL by their previous experience. To mimic the practice of the pathologists, we developed a deep learning based tissue level classification method that does not require cell counting; instead, the entire microscopic tissue image is treated as a whole entity and fed into a convolutional neural network based multi-class classifier to classify the image into 4 classes, and the prior knowledge of the pathologists are embedded in the training samples labeled by them. The sample microscopic images for each grade of follicular lymphoma are in Figure 4.8.

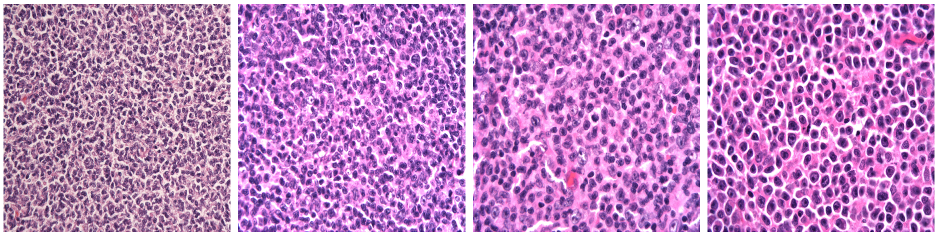


Figure 4.8: Follicular Lymphoma Grades, Grade 1, Grade 2, Grade 3A, Grade 3B

However, the raw tissue images taken from microscopes normally are in big dimension, such as 1024 × 1280 pixels or even much bigger in the case of whole slide scanning, which can even be over 100,000 × 100,000 pixels. The size of the input images used to train convolutional neural networks is constrained by the computer hardware configuration, such as memory and CPU capacity, as well as the power of the graphic processing unit (GPU); for a modestly configured desktop computer, it can normally only take inputs around 256 × 256 pixels or below. There are hand feature based follicular lymphoma grading algorithm [61], but these feature based method only works for a particular type of disease, and for a new type of disease with different visual characteristics, new features will have to be developed, or features need to be selected from a large existing hand engineered feature pool; in addition, there is no guarantee that these features are the most discriminant and representing for the specific classification purpose. Thus we want to have a more generic solution not relying on hand engineered features, and convolutional neural network based algorithms are just a natural choice by medical image processing community [62].

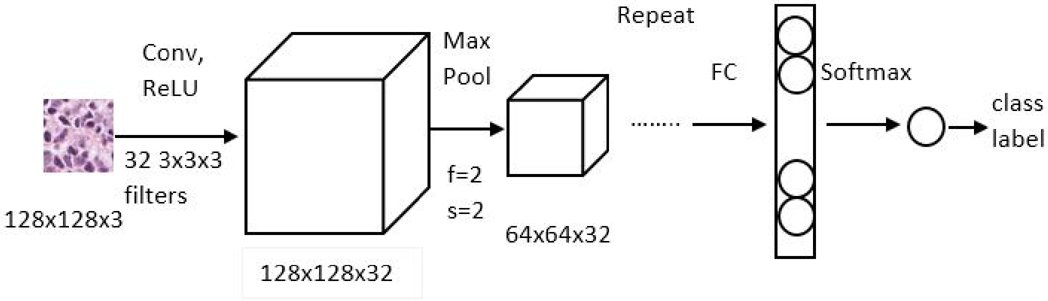


Figure 4.9: ConvNet architecture Diagram for FL Grading

4.6.2 Global Feature Extraction with Convolutional Neuron Network

Though there are already methods using global features to grade FL [63], these features are hand engineered for specific type of images and may fail to generalize, and we want to find a generic feature learning frame instead, where ConvNet is a natural solution [56]. Medical images are quite different from other natural objects people see daily, and pre-trained ConvNet often work poorly since they are not trained for medical images and thus could fail to capture the features that best to characterizes these images. In this case, our model is small enough to be trained from scratch on a modestly configured desktop computer and implemented using Google's Tensorflow framework. It is implemented using a very simple LeNet CovNet architecture, with 3 convlutional layers for feature extraction and 2 fully connected layers to obtain the class score for Softmax classification. The architecture is illustrated as 4.9:

In this model, each convolution layer is followed by a ReLU (Rectified Linear Unit) as the activation function; max pooling is used after that to select the maximum value for each 2×2 grades; after 3 convolutional layers for feature extraction, the features learned are fed into 2 fully connected layers, which are used to obtain the class score for softmax classification.

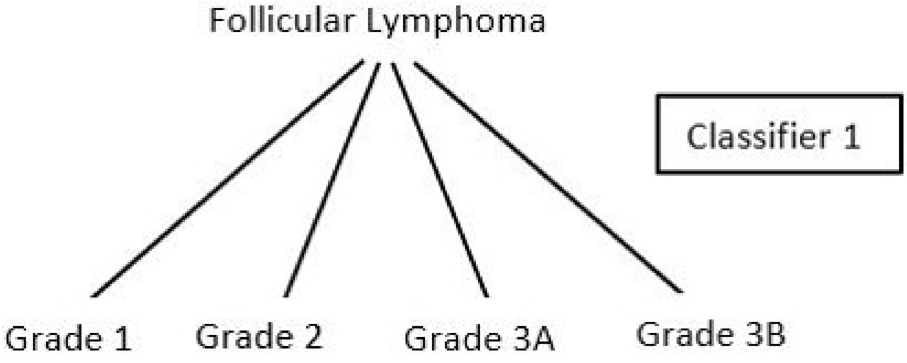


Figure 4.10: Flat Classification

4.6.3 Hierarchical CNN Based Multi-class Classifier

In this research, two different classification architectures are studied; the first one uses only a single ConvNet that extracts features for all 4 grades, and then the features are fed into a softmax classifier for multiple class classification using one-vs-all algorithm. This is illustrated in 4.10:

In the second architecture, to improve the classification accuracy, hierarchical classification method is used; first a given image is graded as low risk (grade 1 and 2) or high risk (grade 3A or 3B) with their own CNNs, and then another binary classification is performed to grade it as 1 or 2 if it is in low risk, or 3A or 3B if it is high risk. The architecture is illustrated in Figure 4.10, where CNN 1 classifies a sample into low risk or high risk, and then CNN 2 classifies low risk samples into grade 1 or 2, and CNN 3 classifies high risk samples into grade 3A or 3B. The high level architecture is illustrated in Figure 4.11.

Since each original image is divided into 12 sub-images, the sub-images from the same original image could be classified into different grades. To solve this dilemma, a majority voting mechanism is proposed by Dr. Qinglong He, that the final grade of the original image is the most popular grade ofits sub-images. For example, out of

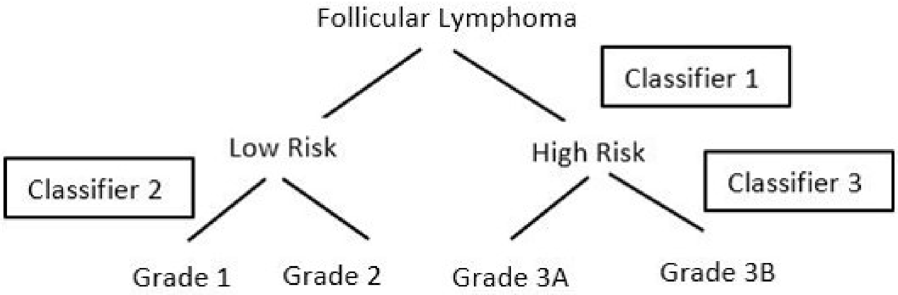


Figure 4.11: hierarchical Classification

the 12 sub-images of the original image, if 5 is classified as grade 2, 3 classified as grade 1, 4 classified as grade 3A, then the final grade is grade 2.

4.6.4 Experiment and Result Analysis

The FL samples we obtained are stained with hematoxilin and eosin (H&E) process, where their nuclei are colored in purple, which is a standard procedure to color the tissue sample to facilitate studying under microscopes.

A total of 47 images of microscopic follicular lymphoma tissues are provided by University of Nebraska Medical Center, with 9 images in Grade 1, 9 images in Grade 2, 15 images in grade 3A, and 14 images in grade 3B; each image is graded by medical experts and the grade labels for each image are considered as ground truth. The size of the original image is 1920×2560. To protect patients privacy, there is no patient information associated with the data.

Data Augmentation: Since the orientation of the cells appeared in a microscopic image is arbitrary, data augmentation techniques are employed to increase the rotation and flipping will not change the nature of the cell image, and we can greatly increase the number of the training samples by rotating and flipping the sample images. to increase the number of training classes, each original images are divided into 12 small images, and each are flipped along the vertical edge, horizontal edge, and diagonal, and then mirror images. So 72 images are generated from each original image, and the total number of images is increased to 3384. The size of each small image is normalized to 128x128.

In our experiment, 20% of the data is reserved for testing, and the remaining 80% of the data is used to train a model. If treating each sub-image is an independent sample, the classification results are in Table 4.4:

	Flat	Hierarchy
H/Low Risk	100%	100%
4 Grades	86%	94%

Table 4.4: Results when treating each sub-image is an independent sample

Because one original image is split into 12 sub-images, and the sub images from one original image should be in the same grade, but the sub-images might be classified into different grades. This can happen when more CBs are concentrating into one or a few sub-images, and the the other sub-images may contain less CBs. Thus a majority voting mechanism is used to decide on the final grade for each original image. The result is shown in Table 4.5.

	Flat	Hierarchy
H/Low Risk	100%	100%
4 Grades	100%	100%

Table 4.5: Grading results for original images when majority voting is used for each sub-image

The experiment on microscopic images collected on 47 patients demonstrates that this algorithm yields results 100% consistent to the labels provided by expert pathologists. It gives instant grading results and smaller tissue samples can be used. In additional, the medical researchers can grade the severity of the disease into more

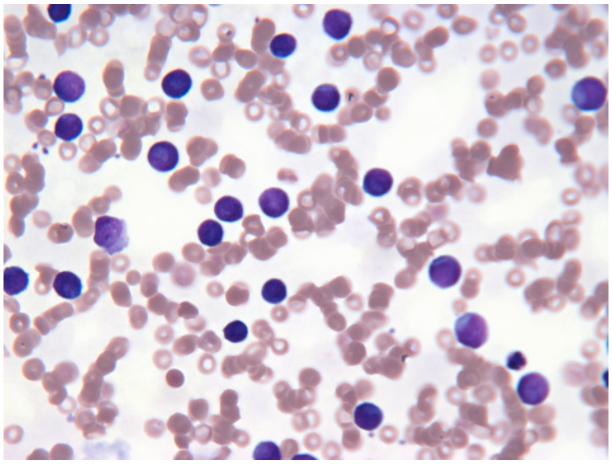


Figure 4.12: White Blood Cell Counting

grades if pathologists provide more granularity on samples, and give quantitative measurement of the severity of the disease. The same framework can be used to classify and grade other types of cancers or other diseases if labeled sample are provided for training. It can be seen that with the majority voting mechanism, the grading results are all 100%. However, due to the limited number of patients, the results could be subject to statistical error; this research is more on the validation of feature extraction using ConvNet. Further research with more patients information should be more statistically meaningful for training a model that can be used for clinic practice.

4.7 White Blood Cell Classification and Counting

Complete White Blood Cell (WBC) counting in peripheral blood is a routine health screening procedure for early detection of many diseases, including leukemia. The change in the number of each type of white blood cells may indicate the change of health condition of the patient. It is also used to diagnose specific types of blood diseases. Normally cell counting and classification is done manually by pathologists using a microscope, and is tedious and prone to human error.

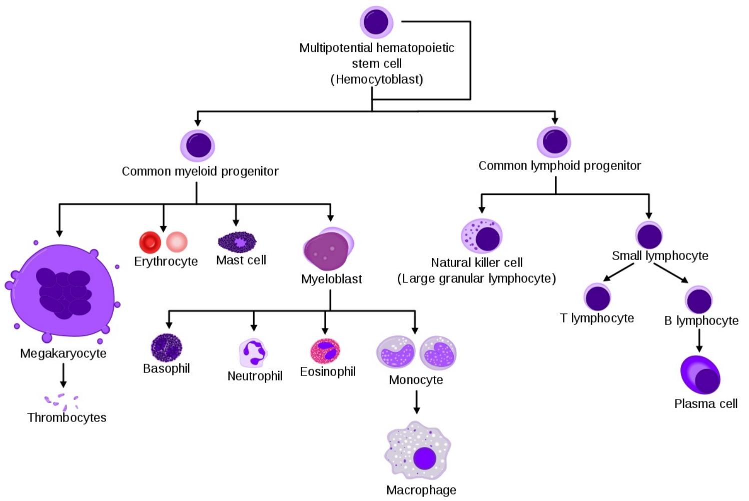
Leukocytes, also called White Blood Cells (WBC), are cells in human immune system against infectious diseases (bacteria, virus and parasites) and foreign invaders. Complete White Blood Cell counting in peripheral blood is a common health screening procedure for early detection of many diseases. The change in the number of each type of white blood cells may indicate the change of health condition of the patient, and is also used to diagnose blood diseases such as leukemia or lymphoma. There are specific medical instruments based on the physical properties of WBCs for this task (find reference), and visual cell counting and classification can also be done manually by pathologists using a microscope, which is a tedious and time consuming process and prone to human error.

To improve the productivity of pathologists and alleviate them from manual labor, we developed a computer vision based white blood cell counting system to automate this procedure. This system combines traditional computer vision techniques such as color based segmentation with the state-of-the-art convolutional neural network for cell classification and counting; this system is also able to mark each type or subtype of the WBC on the microscopic images, and allows pathologists to interact with the images to select or highlight individual cells. This system is highly configurable and can be used for many different types of cell counting tasks.

The WBCs are stained with purple color; a total of 100 WBCs are manually segmented to extract the color information; the WBC images are converted into HSV space to mitigate the impact of different lighting and camera exposure conditions, and the distribution of the hue information is calculated. Since in peripheral blood, the WBCs are sparsely distributed, we can ignore occlusion and assume the WBCs do not overlap or touch each other. However it is still very challenging to segment the WBCs out of each image due to the vast difference in shape, size, and number of nucleus, thus systematic quantitative analysis for peripheral blood images is needed.

There are around 4-10 billion WBCs in a liter of blood for a healthy adult. There are 5 types of white blood cells, Neutrophil, Eosinophil, Basophil, Lymphocyte, and Monocyte, each with different functions. The numbers of each type of WBCs in the blood are vastly different, with Neutrophil constituting 60-70% of the leukocytes, Eosinophil 2-4%, Basophil less than 0.5%. Their sizes are also drastically different, as monocytes are the biggest. Since the change of the number of the white blood cells may indicate health problem, for example, leukemia patients will have leukocyte count much higher than normal, counting them is a way for early detection of diseases such as leukemia.

To improve the productivity of pathologists, we developed a computer vision based complete white blood cell counting system to automate this procedure. This system combines traditional computer vision techniques such as color based clustering with the state-of-the-art convolutional neural network for cell classification and counting; this system is also able to mark each type or subtype of the WBC on the microscopic images, and allows pathologists to interact with the images to select or highlight individual cells. This system consists of classification and counting modules; classification module is a trained convolutional neural network (CNN) to classify types or subtypes of individual WBC. Counting module segments WBCs from the images using color based segmentation in HSV space and clustering techniques such as mean-shift and watershed algorithms. To segment clustered WBs, adaptive hierarchical segmentation technique is used. The CNN is trained with individual WBC images before it is used as multi class WBC classifiers. This tool is currently under development.



Multipotential hematopoietic stem cell (Hemocytoblast)
Common myeloid progenitor
Common lymphoid progenitor
Megakaryocyte
Thrombocytes
Erythrocyte
Mast cell
Myeloblast
Basophil
Neutrophil
Eosinophil
Monocyte
Macrophage
Natural killer cell (Large granular lymphocyte)
Small lymphocyte
T lymphocyte
B lymphocyte
Plasma cell

Figure 4.13: White Blood Cell Types

4.7.1 Peripheral Blood WBC Counting Pipeline

This system consists of segmenting, classifying and counting processes; first the microscopic peripheral blood image is converted from RGB to HSV color space, and then DBSCAN is performed to make the color inside the cell look even. Several WBCs are selected to identify the Hue value of the WBC, and a color mask is made to select WBCs based on its color. Then the segmented WBCs are fed into a trained convolutional neural network to classify types or subtypes of individual WBC. Counting module segments WBCs from the images using color based segmentation in HSV space and clustering techniques such as mean-shift and watershed algorithms. The CNN is trained with individual WBC images before it is used as multi class WBC classifiers. The process pipeline is illustrated as in 4.14.

4.7.2 Color Based Leukocytes Segmentation in HSV Space

The total number of leukocytes is an important indication of the health of a patient, and can be used for early screening of many diseases such as Leukemia. For exam-

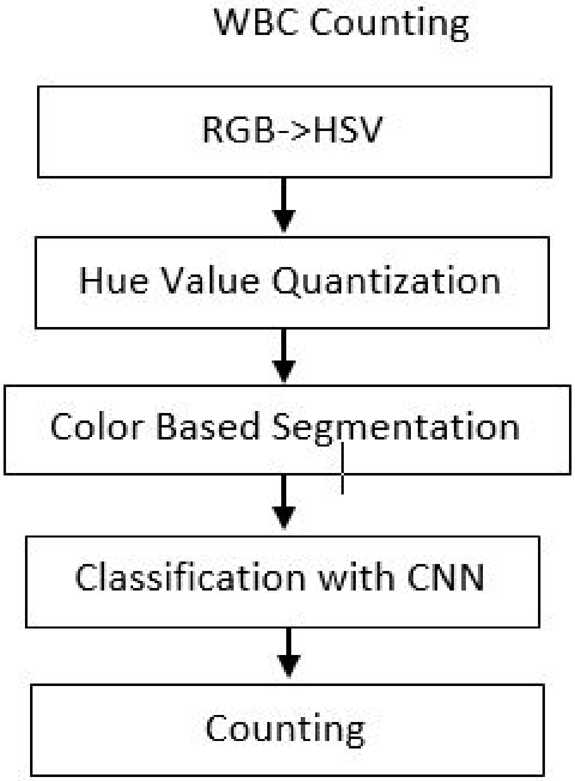


Figure 4.14: White Blood Cell Counting Pipeline

ple, Leukemia patients have far more Lymphocytes in the peripheral blood due to a malfunctioning immune system.

The color of leukocytes is stained to purple with H&E process; since different images may be taken under different lighting conditions, to mitigate the impact of lighting change, the image is first converted from RGB color space to HSV color space, and the hue value is used to segment the white blood cells. In order to smooth the color, it is quantized using DBSCAN clustering algorithms. It can be shown as in 4.15. Because of the ellipse shape of the cell, it can be approximated by ellipse fitting.

The WBC cells in peripheral do not appear in cluttered form and they often do not occlude. To prevent possible occlusion, the blood sample can be further diluted to

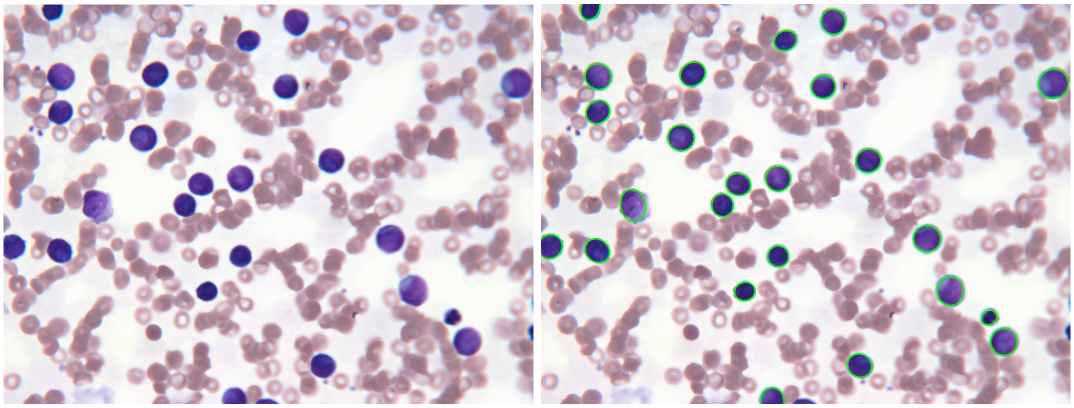


Figure 4.15: Color Based Segmentation

ensure that there will be almost no occlusion for WBCs.

4.7.3 Leukocytes Counting and Classification

There are 5 major leukocyte types, to which belongs polynuclear (granulocyte) and mononuclear (agranulocyte) groups. Mononuclear leukocytes include lymphocytes and monocytes, and polynuclear leukocytes is so named because of the varying shapes of the nucleus, which is usually lobed into three segments, and they include the other three WBC types. In this research, we use two methods, ellipse fitting, and convolutional neural network, to classify the white blood cells into polynuclear and mononuclear.

The segmented leukocytes will be fed into convolutional neural network for classification. Due to the limited available dataset, the leukocytes are only classified as polynuclear and mononuclear. The same ConvNet that is used for Follicular Lymphoma is used for classifying leukocytes here.

4.7.4 Results and Future Work

We have 66 images of blood samples with WBC labeled as polynuclear or mononuclear. Manual Counting shows that there are 173 leukocytes, with 25 mononuclear and 168 polynuclear. The test results show that the counting accuracy for overall WBC number is 100%, and the classification accuracy is 5 wrong classification among 173 leukocytes, which is 97.1%.

The test shows that this WBC counting tool is able to accurately segment WBCs from background, count the number of each type of WBC, it enables the pathologists to search for the each type of WBCs and highlight and mark them; this tool is pure software based with almost no additional cost, and it provides a practical tool to assist pathologist for WBC counting; instead of spending hours to count the cells, this tools can complete the cell counting in seconds and enable the pathologists to focus on the essential diagnostics tasks.

Conclusions and on-going research: The software system enables automated complete WBC counting in a stained periphery blood sample. It processes a given microscopic peripheral blood sample images, counts and marks each type of WBCs. However counting the 5 major WBC types does not resolve all diagnostic issues; WBC counting issue in diagnosing blood disease; many types of leukemia is caused by that certain type of WBCs stop growing before reaches maturity, and it will not be able to function;

The software system enables automated complete WBC counting in a stained periphery blood sample. It processes a given microscopic peripheral blood sample images, counts and marks each type of WBCs. However counting the 5 major WBC types does not resolve all diagnostic issues; WBC counting issue in diagnosing blood disease; many types of leukemia is caused by that certain type of WBCs stop growing

before reaches maturity, and it will not be able to function; thus the subtypes and the development stage of the WBCs also need to be determined.

4.8 From Hand Feature Engineering to Automated Feature Extraction in Medical Image Analysis

In this chapter, we applied computer vision techniques in the medical diagnosis domain, which not only alleviates the pathologists from tedious manual cell counting and image analyzing, it also removes the subjective inconsistency of the pathologists. It has very practical clinical uses, not only making the medical image analysis and diagnosis faster and more accurate, it also makes it more objective.

The classifiers only perform as well as the features they use as input. There are much ways to construct the most representative and discriminant features, and feature engineering has been the center point for medical image analysis research. Traditionally, features are hand engineered by researchers with domain expertise, and then numerous feature selection methods are proposed to select the most discriminant features among the hand engineered feature sets, mainly relying on sparse learning. In recent year, automated feature extraction using deep neural networks, primarily convolutional neural networks, have been gaining popularity, and at present days, ConvNet has become the primary way for medical image feature extraction.

Deep Learning is not panacea; it has its advantages and disadvantages. It provides the ability to automate feature extractions; however, domain knowledge are still required to label the ground truth, preprocessing the image, or even assess if the problem is ”learnable”. Also ConvNet tends to work well with low resolution images, while many of the medical images, such as whole slide scanning images, are very high in resolution and image size, which requires domain specific knowledges for analysis.

One big obstacle of Deep Learning approach is the large number of labeled training samples required to train the model, which is usually very difficult to obtain due to the fact that these images need to be studied and labeled by medical experts, to whom most people have no access.

Chapter 5

CASE STUDY: DEVELOPING AND DEPLOYING REAL-LIFE COMPUTER VISION SYSTEM

In the last two chapters, computer vision algorithms used in two important areas of application, intelligent surveillance systems and medical diagnosis systems, are studied, with emphasis on using optical value in video for real time ROI generation and using optical value for human action recognition feature and motion saliency region extraction, and algorithms to extract most discriminant features with feature engineering and end-to-end feature learning techniques. However, developing and deploying real world computer vision systems is more than just algorithm development; it is also a software engineering problem and need to follow software engineering principles. Developing real world deployment ready computer vision software poses additional challenges to system development, where both software and hardware architecture design, as well as constraints such as time line, budgeting, ease of deployment, etc. of the overall system have to be taken into consideration. In these cases, the software engineering approach would need to be taken to address the software and system development life cycle.

For many computer vision usage scenarios in the manufacturing industry and environmental monitoring, the algorithmic challenges are different from previously discussed usage scenarios, such as in surveillance and medical, as described in the last two chapters. In these cases, one of the major tasks is to detect a rigid body with known shape and color, often two dimensional, such as painted marks, or a thin computer chip, under different lighting conditions and perspective transformations because the camera set up and view angles may vary in different scenarios, and determine their location and dimension measurement. The objects to be detected have relatively minor or no intro-class variations. Appearance based algorithms such as Hough Transform and template matching work in most situations to detect and classify these objects. However, this usage scenario has its unique requirements for real life usage, including robust to lighting change, robust to perspective transformation, scale invariant, high (near 100%) detection rate, real time execution speed, etc.

In this dissertation, first a generic algorithm for real time scale adaptive template matching for color image is introduced, and then the core algorithm is used to implement a complete computer vision application, real time water level monitoring system, including hardware set up, software architecture, and its deployment in real life. This application is also used to exemplify the design principles and computer vision application development life cycle.

5.1 Applying Software Engineering Principles in Machine Vision Application Development

Industrial software development projects are more than just *ad hoc* programming effort. For a practical software product, it needs to be reliable, robust, enhanceable, cost effective, easy to use and deploy. Developing real life computer vision application should also follow the principles of software engineering. For developing a real life computer vision application, there are several major steps:

- Requirement analysis and data collection
- System architecture design
- Algorithm development
- Software development

- Test, integration, and deployment

The computer vision algorithm provides the core functionality of the application, and the overall system performance largely depends on it.

5.1.1 Software Engineering Principles

Developing real life software application requires systematic approach. There are seven widely accepted principles of software engineering [cite: Seven basic principles of software engineering]; they can be summarized as:

- Rigor and formality
- Separation of concerns
- Modularity
- Abstraction
- Anticipation of change
- Generality
- Incrementality

These principles are the best industrial practices people have learned over the last several decades, and following them is essential to develop high quality software product. The use of software engineering principles will be demonstrated in the computer vision based water level monitoring system in the next section.

Besides the basic principles, there are multiple design paradigms that software development can use. Which one to use depends on the nature of the specific development project. In this case study, iterative development paradigm is used.

5.1.2 Computer Vision Application System Architecture and Development Cycle

For the real world computer vision applications, the computer vision application can either be a standalone, self-contained system that performs a specific task, such as detecting a speeding vehicle; or it can be conducting vision based tasks in a bigger system, such as a vision based defect detection module in an automated assembly line.

A computer vision application consists of several processing module, including image or video data acquisition, data transmission and storage, computer vision algorithm, user interface, as well as computing, storage, and networking hardwares that constitutes a complete software execution environment. In addition, whether the computer vision algorithms are appearance based or machine learning based, there are always many model parameters that need to be determined for the specific deployment environment; even the system has been developed with previously collected data, the model parameters still need to be fine-tuned during the field deployment.

There are several basic software engineering principles that almost all real life software development projects should follow to produce high quality code, including modularity, abstraction, anticipation of change, generality, iterative development, consistency programming interface and user interface. Though some of them are heuristic and rule of thumbs, they are the best know business practice software developers learned in decades of software development.

There are many software engineering development models that can be applied to the software development cycle, such as *Increment*, *Iterative*, and *Agile*; depending on the scale and nature of the software project, different models and design paradigm can be applied. The iterative development methodology is adopted in this project, as shown in figure 5.1

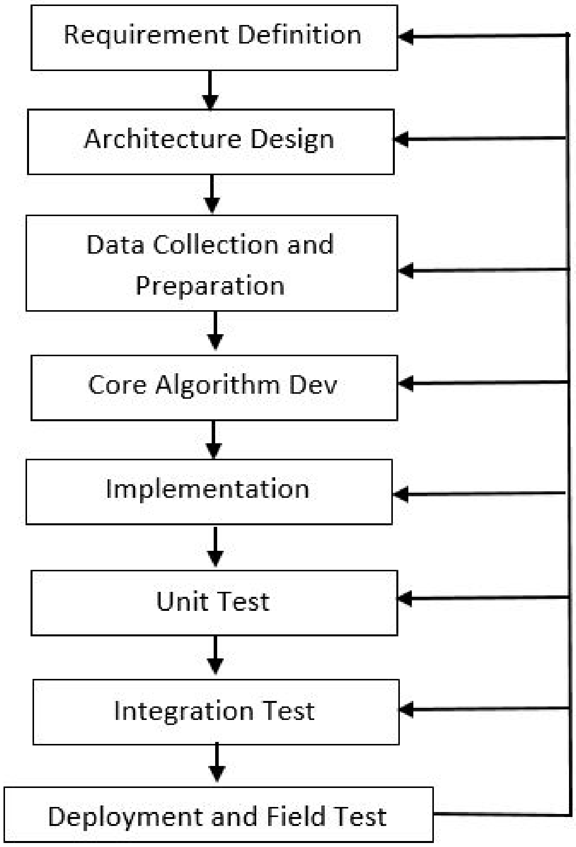


Figure 5.1: Software Engineering Approach for Water Level Monitoring System Development

The central piece of a computer vision application is the computer vision algorithm, and the core vision algorithm should be designed to be as generic as possible, so it can be extended to different scenarios.

5.2 Real Time Scale Adaptive Affine Template Matching

Even though machine learning based object detection and recognition algorithms have achieved great success, especially with deep learning algorithms such as convolutional neural network, and models derived from it, such as R-CNN, Faster-CNN, etc., they are not without shortcomings in practical use. Their main advantage is the ability to learning features that can not be easily described by simple rules, e.g., it is difficult to build a template to differentiate a dog from cat because of the intra-class variances and the change of viewing angle, lighting condition, shading... thus recognizing those objects relying on complicated feature engineering or feature extraction algorithms to obtain higher level representation of the object. However, many simple objects, such as a sphere, a rectangle, etc., can be easily detected by using template matching algorithm, or being described by geometric rules. Template matching can be a simple and powerful tool for many practice applications, especially in a controlled environment where the object itself is rigid, and designed by the algorithm developers. In this research project, the use of real time template matching is used to illustrate its effectiveness in computer based water level monitoring system for locating the gauge marks, which is used to decide the relative location of the waterline.

The same idea can also be applied in the manufacturing industry. The components on the assembly line is often rigid bodies, if a component is thin, or it has a flat facet, which are three dimensional. However, many of the components are very thin, such as integrated circuits (IC); recognizing the top side of the surface is enough to recognize the entire component. It also works for cases where the objects to be detected have no deformation other than the perspective transform.

Due to different camera view angles, the object image captured by the camera will undergo perspective transform. From a distance, perspective transform can be approximated by simpler affine transform, which reduces the transform matrix from a 4×4 matrix to a 3×3 matrix. Since in this case, the camera is far from the gauge, and each marks on the gauge occupy a relatively small part of the image, affine transform is used.

In next subsection, we first propose a generic algorithm that will detect object by fast affine template matching, and then this algorithm is customized to detect the water marks.

5.2.1 Fast Affine Template Matching Algorithm

Template matching algorithm detects the objects by matching the candidates with a predefined template. The similarity between the target and the source (template) has many ways to measure; the most commonly used one is the sum of absolute differences (SAD) distance between two images I_1 and I_2 is designated as $\Delta_T(I_1, I_2)$, which is defined in 5.1:

$$\Delta_T(I_1, I_2) = \frac{1}{n_1^2} \sum_{p \in I_1} |I_1(p) - I_2(T(p))] \tag{5.1}$$

However, even for 2D rigid shapes that has little intra-class and inter-class variations, because of the difference in scaling, view angle, brightness, shading, lighting color, etc., the SAD value between the template and target can be very large even for the same object, thus the template matching does not work well for objects under different lighting and perspective transformation.

In order to match template under affine transformation, it has to match templates under all possible perspective transformation and scale. In this research, the Color Fast Affine Transform proposed in [64] is used to match the template under affine transformation.

5.2.2 Scale Adaptive Fast Affine Template Matching

Though already optimized for speed, the fast affine template matching still consumes considerate amount of time due to the fact that it has to be rescaled multiple times for scale invariant template matching. In many computer vision tasks, the input data source is video, which normally comes in at 30 FPS (frame per second). Also, a surveillance system normally consists of many, in some cases hundreds of video cameras, thus the vision application should be capable of processing hundreds of video channels simultaneously. For these applications, processing speed is a critical performance requirement.

In this study, we optimize the Fast Affine Template Matching Algorithm for speed to enable real time performance for multi-channel video input. To improve the process speed even more for template matching under affine transform, if the object size is already known, then the template only needs to be scaled once to the size of the object, thus reducing the computation needed by an order of magnitude. It can be used as computer vision module in environment monitoring or automated manufacturing. The basic idea behind of the enhanced algorithm is that the original Affine Template Matching Algorithm needs to be rescaled multiple times in order to match the target images that may come in with various scales, if we can estimate the size or scale of the matching target, we will only need to match it once, or a very limited times, thus greatly reduces the computational complexity.

5.2.3 Object Detection with Scale Adaptive Color Affine Transform Template Matching

To look for the template in the target image, the objective function for template matching is to minimize the normalized mean square error between the template and the matching area target image. However, thexxx Template matching under affine transform can be efficiently executed using the algorithm proposed in [65]. However, this algorithm is for gray scale images, and there are many situation where pixels have similar grayscale value but different color, thus making it necessary to enhance this algorithm in color space. To maximize similarity measure between the template and target, the template matching in color space is used. Because the distances between the gauge camera and the camera view angles might be different at each installation, to use template matching algorithm to locate the red strips, we need to perform it scale invariant affine template matching on the blobs segmented against a set of templates of red strips. [64]

Object Detection with Scale Adaptive Template Matching is a special case of C-FAST fast color template matching algorithm under affine transform [64]. In the C-FAST algorithm, the template is rescaled multiple times for scale invariant. However, if the object size is known or can be estimated, the template only needs to be resized to few very limited scales close to the object size, thus largely reduces the computational time, to less than 10% of the C-FAST algorithm. Since this algorithm adapts to the object size, we name it Scale Adaptive Affine Transform Template Matching.

5.3 Case Study: Develop and Deploy a Computer Vision Based Water Level Monitoring System

As a use case of computer vision application, in this section, we demonstrate the development of a multi-channel cloud hosted and computer vision based water level monitor system to automate the water level monitoring and analysis. It leverages the existing surveillance infrastructure and public cloud computing platform and enables automated data collection and analysis, so it is significantly cheaper to set up and operate than the current mechanic or ultrasound based water level monitoring systems, and only minimum level of human effort are required to operate and maintain, and it stores historic water images and data for off line analysis as well. It can monitor water levels for still or flowing water body for a wide area to provide holistic and real time water level information and trigger alert for predefined events. This system can also be provided as a service to third parties; the new users can link their IP cameras to the system and the water level can be monitored for them with this cloud system. To the best of our knowledge, this is the only cloud based water level monitoring system with computer vision technology in practical use in China as in Fall, 2017.

Water level monitoring is essential for flood prevention/control and water resource management. Automated water level monitoring can provide accurate water information in real time for multiple locations without human intervention for water resource and disaster management authorities. Currently in China, as well as in most of the world, water level measurement is still mostly collected by people visually reading various types of water gauges, and the most widely used automated water gauges use ultrasonic based technology that determines the water level by measuring the round trip time of ultrasound wave from the sound generator to the water surface in a pipe, which are expensive to manufacture and install; also the pipes can be easily jammed by mud and trash in water, thus regular maintenance is required. This prompts the need to develop a cheaper and networked solution, and the advancement of computer vision enables this request. There are already some proposals such as [66] using computer vision technologies for water level monitoring. However due to various difficulties such as high deployment cost or the algorithm not robust to negative factors such as poor lighting and weather conditions, there is no widely used computer vision based water level systems in practical deployment. Thus we want to develop a robust, low deployment, and versatile computer vision system for water level monitoring using the existing video surveillance infrastructure and capable of reading water level from many different types of newly designed as well as existing water gauges.

5.3.1 System Architecture Process Pipeline

To reduce equipment cost and expedite deployment, our system design aims to maximally leverage the existing surveillance and telecommunication infrastructure, and reduce hardware dependency by implementing functionalities in software as much as possible, thus minimizing the initial set up and ongoing operation costs. The system hardware consists of easy to install water gauges, IP cameras, and a cloud based server to host computer vision software, web server, and to store data. The software can be scaled up to monitor and analyze hundreds of channels of water gauge videos simultaneously, and the computing power can be easily scaled up by renting more cloud servers.

The overall system design is illustrated as in 5.2; it performs end to end processing, from input image capturing, to image processing and visual information extraction, to back-end data processing and the interface to users and other enterprise applications.

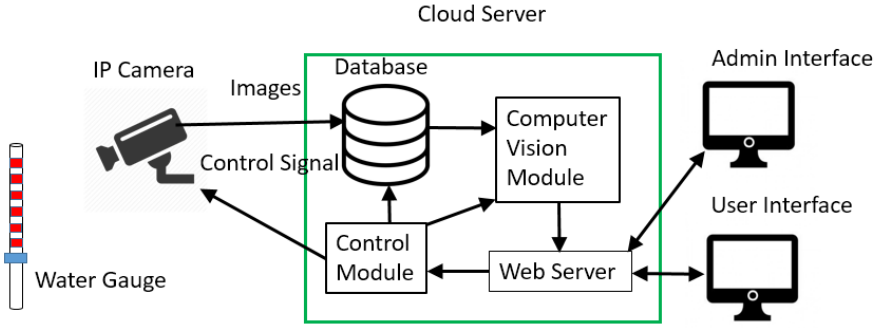


Figure 5.2: Intelligent Water Gauge System Design

5.3.2 Water Level Reading Process Pipeline

The common tasks for reading water level follow the steps in 5.3:

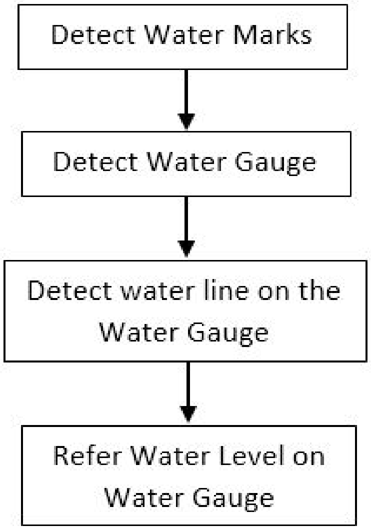

Detect Water Marks
Detect Water Gauge
Detect water line on the Water Gauge
Refer Water Level on Water Gauge

Figure 5.3: Water Level Reading Process Pipeline

Object Segmentation Module: to identify the ROI that contains the foreground objects, in this case water gauge or painted marks; the ROI location and size are also used for estimating the object locations and sizes;

Object Classification Module: to determine if the object of interested is truly contained in the ROI; for water gauge, this module determines if the water mark is truly within the ROI.

Object Locating Module: inside the ROI, pin-point the object location, often involving finding the edges of the components for measurement; in our case, the location of each water gauge mark is pin-pointed.

Measurement Inference Module: the locations of object, such as painted marks, can be used as reference to identify the location of object of interest, such as water line.

5.3.3 Water Level Staff Gauge Design

The water gauges are designed in such as way that it is both easy for human eyes to recognize, and easy for computer vision algorithm to process. Since there are many existing gauges in use today, our system should be able to recognize them as well. The following Figure 5.4 are the main types of gauges that are tested.

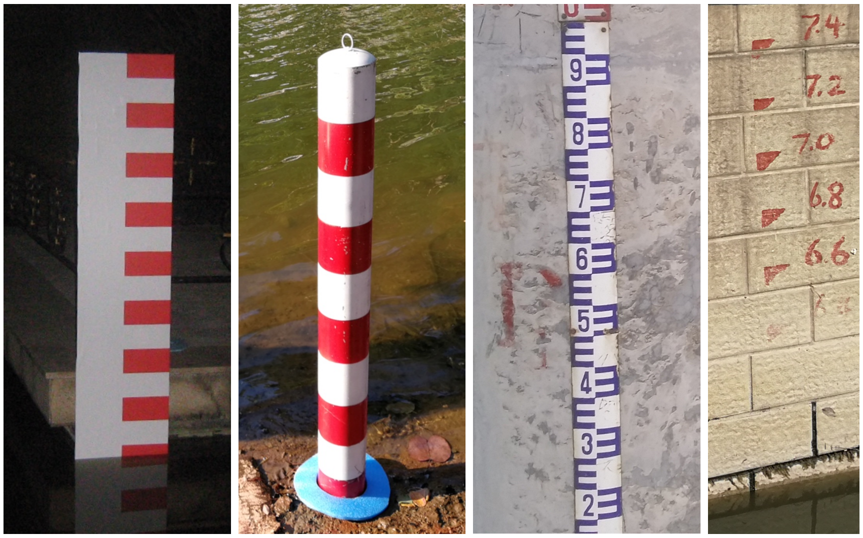


Figure 5.4: Different Water Gauge Design

In this dissertation, the flat panel water gauge is used as the example to illustrate the image, at the segmented image. The algorithm will have minor modification for different gauge designs.

5.3.4 Algorithm

The computer vision algorithm in this application is to extract water level reading from the gauge image. There are two major steps for water gauge detection, waterline detection and gauge mark detection.

It contains 3 major steps, first, the gauge marks are segmented and detected, second, the waterline on the staff gauge is detected, and finally, the water level reading on the gauge is determined. Each step is elaborated in the following sections.

Detecting Staff Gauge Marks

For field installation, the lighting condition is not consistent through out the day and in different weather condition; to mitigate the impact of lighting condition change, we can transform the color space from RGB to HSV, and use the hue value to segment the foreground objects from the background. In these scenarios, the system set up is in a controlled environment where the background and be set up or selected without the conflict of the color of foreground objects, thus HSV color based segmentation is sufficient for our specific use cases.

To detect the marks in the gauge, the first step is to use the known color to segment out all areas of the same color. To ensure the subtle color difference caused by shading, light change, paint variations, DBSCAN clustering algoirhtm is first used to "blend" the colors into homogeneous. Then the color mask is used to segment the marks; the process pipeline is as in Figure 5.5.

To use scale adaptive fast affine template matching, the scale of the object to be detected, in this case, the marks on the gauge. The process is illustrated in Figure 5.3.4

The blobs that are segmented will be used for scale adaptive color template matching. The locate of each water mark is then used to calibrate the water gauge.

Waterline Detection

Due to different view angles of the camera, the water line may not necessarily be horizontal. However, in this design, only the water line over the water gauge is considered, which greatly reduces the processing difficulty.

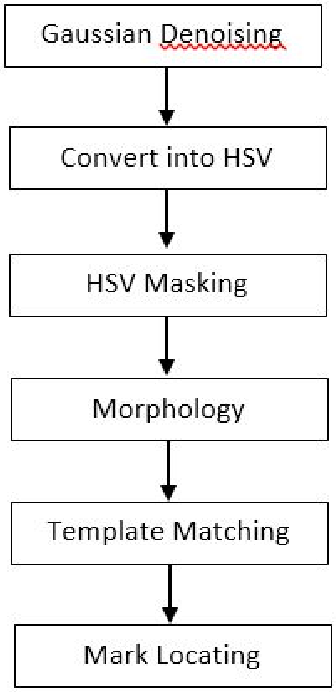


Figure 5.5: Process Pipeline for Detecting Water Gauge Marks

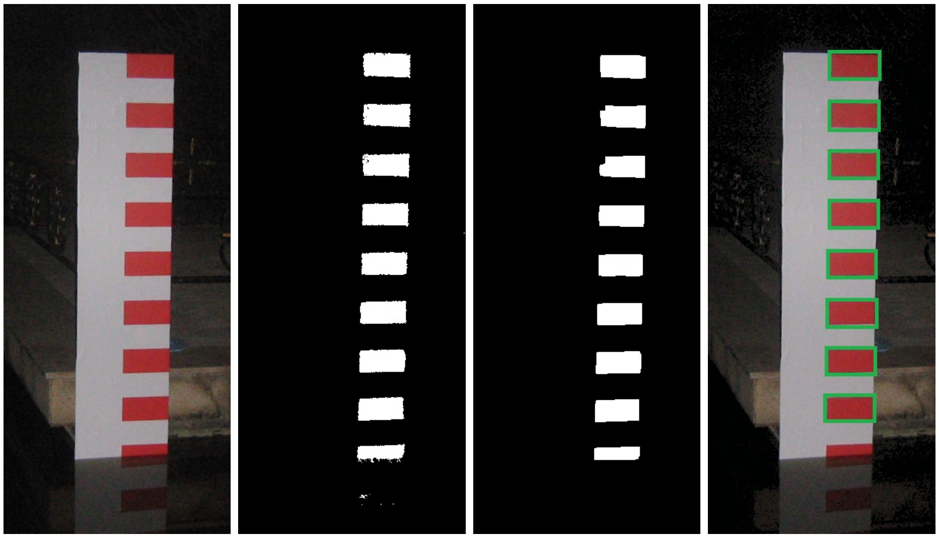


Figure 5.6: Detect watermarks from the panel, and then use Scale Adaptive Color FAST Affine Template matching to detect each water mark

The first step of waterline detection is looking for the water panel. The waterline is obtained by using Hough Transform to detect a straight line.

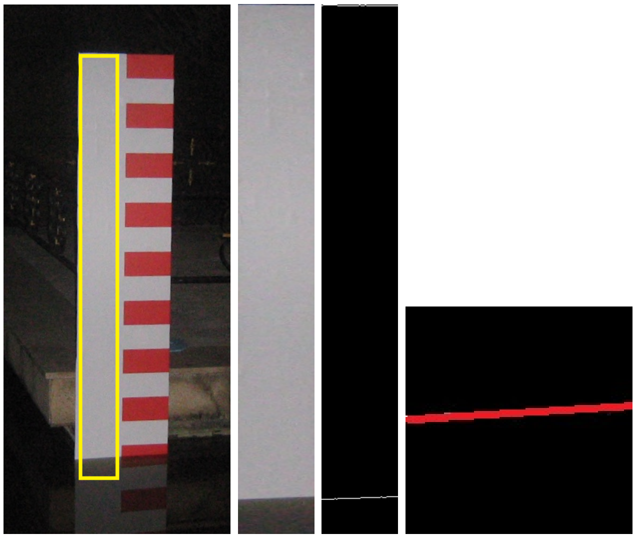


Figure 5.7: Detect waterline from the panel: first segment the area next to the marks, then perform edge detection, then use Hough Transform to fit a straight line

Water Level Inference

After the water marks are detected, the water level relative to the top of the gauge will be inferred. The gauge mark is spaced 10cm each; so the relative water level is measured at the 10cm scale is measure by the staff gauge mark above it, and the water level at centimeter scale is approximated by using linear extrapolation from the marks above.

5.4 Results and Future work

The accuracy of computer vision based water level monitoring system is measured by comparing the human vision reading and the result of the software system. For the 100 test cases, the average difference between the human reading and the computer vision system results is about 0.85cm, and are considered to be accurate enough as the actual water level reading.

Since the water level did not have a drastic change during the period of test, the water

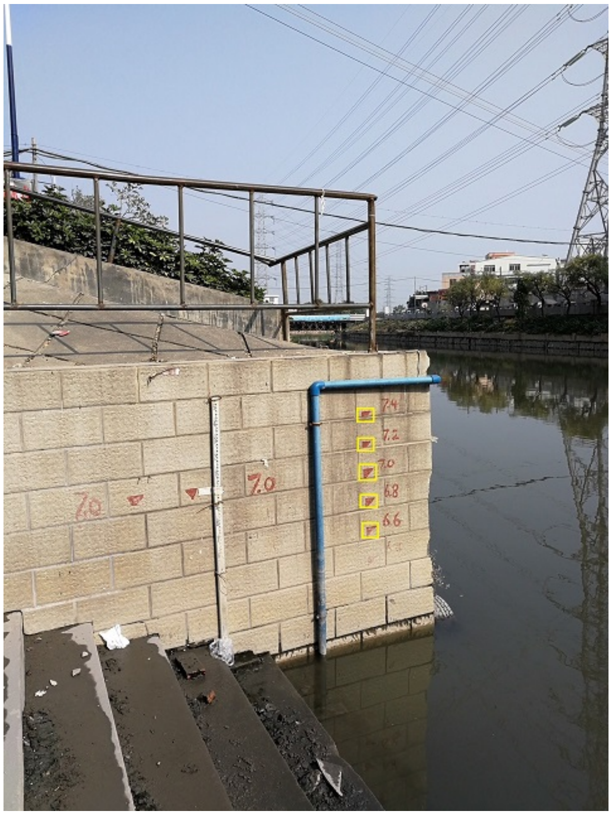


Figure 5.8: Hand Painted Marks

gauge was moved up and down to different depth to simulate the water level change.

The experiment results are shown as Table 5.1:

Test number	Ave. Difference
100	0.85 cm

Table 5.1: Manual Visual Reading vs. Computer Vision reading: In 100 readings, the difference is less than one centimeter.

The hand painted signs can also be recognized using this framework; an example is shown in figure 5.8.

Contrary to the popular thinking that computer vision software is difficult to develop for real life application and development life cycle it long, to solve many practical problems in the real world, by modularizing the problem, commercial computer vision software can be developed by a small team in a relatively short amount of time, such as locating specific parts on an assembly line, counting and sorting components, defect detection on product surfaces, counting cells, semantic based video compression, environmental monitoring, etc. The prototype of Scale Adaptive Color Fast Affine Transform demonstrated its viability in the field test for water level monitoring, and will be adopted for more future environment monitoring and manufacturing automation applications.

Chapter 6

CONCLUSION, CURRENT DEVELOPMENT AND FUTURE WORKS

This PhD research spanned about 10 years; during this time, computer vision technology has experienced their most rapid growth since its inception in 1950s, especially with digital cameras and smart phones becoming ubiquitous and the development of computer vision frameworks greatly reduced the level of complexity of developing computer vision applications. Computer vision, as a special area of the more general *Artificial Intelligence* research, has been in its ”Cambrian explosion” since 2012, mainly because of the breakthrough of using multi-layer artificial neural network, usually referred as the *deep learning* revolution. During this time, the computer vision algorithms have been gradually shifting from the traditional rule based or hand feature engineering to the automated feature extraction with deep learning based technologies, especially with different variations of convolutional neural networks. Nowadays deep learning based algorithms have exceeding humans capability in common object recognition tasks. In many areas that requires expert level of image classification, such as medical image diagnosis, computer vision algorithms have shown great potential, including the research work presented in this dissertation.

6.1 Summary of Contribution

A.I. (Artificial Intelligence) has become a hot word in scientific community as well as for the general public. For the general public, AI is about machine learning, machine learning is about deep learning, and deep learning is about supervised learning. From mathematics point of view, all it can do is to map one data manifold X into another manifold Y, assuming the existence of a learnable continuous transform from X to Y, and the availability of a dense sampling of $X : Y$ to use as training data.

In this research, we studied the system architecture, software framework, and the algorithm development for computer vision applications. We applied to two areas which have direct impact to people's lives, the intelligent surveillance and medical image analysis, and present the system architecture and innovative computer vision algorithms we develop for these applications, and demonstrate the system design and software architecture of several real life use cases and their successful field deployment.

In the intelligent surveillance area, we developed an efficient algorithm for pedestrian and vehicle detection, and a convolutional neural network based action recognition algorithm, and a system and algorithm design for intelligent water level monitoring system.

In the medical diagnosis area, we developed an innovative sparse feature selection algorithm to diagnose stomach disease with endoscopic images, and an algorithm for semantic based compression of gastroscopic videos, and a convolutional neural network based tissue level follicular lymphoma grading algorithm.

Though deep learning, especially convolutional neural network, has shown its great performance gain over the traditional feature engineering classification algorithms, it has its own limitation. It requires large amount of labeled dataset, often by experts, which may not be easily available. The only real success of deep learning so far has been the ability to map space X to space Y using a continuous geometric transform, given large amounts of human-annotated data. Currently we can only learn programs that belong to a very narrow and specific subset of all possible programs.

For object detection in real world applications, the biggest learning from this research is that incorporating the prior knowledge of the object can greatly reduce the computational complexity and improve performance, especially this knowledge of the objects are visually salient, such as the brightness, distinctive color, specific moving pattern, etc. Modal modal learning is especially efficient when data from other types of sensors are available, such as pedestrian and vehicle detection by incorporating the LiDAR or ultrasonic sensor data.

6.2 Current Trend in Computer Vision

Computer vision, which is considered as part of the Artificial Intelligence, is one of the hottest research areas in both theoretical research and commercial development in recent years (since 2012-2013), and has become a focal point for technology industry, when AlexNet beat The technology breakthrough largely attributes to the breakthrough in deep learning, where optimal features are learned automatically from the raw image, and the learned features outperform hand engineered features.

6.2.1 Advancement in Deep Learning Frameworks

The most noticeable research advancement since 2012 is the so called *Deep Learning*, which loosely referring to the algorithms using multi-layer neural network. Though originally proposed in 1980, due to the hardware restriction

The currently development in the computer focuses on automated feature learning, mainly by employing deep neural network. Their application in the real world application is largely fueled by the pre-trained models that have been trained on large image dataset such as ImageNet.

In addition to discriminant tasks such as classification, GAN (Generative Adversarial Network) can be used to generate new images based on the training images; for example, it can be used to imitate oil paintings of Van Gogh, or synthesize real life like photos based on those in similar scenes.

Region of Interest Proposal

For object detection, not only the presence of the object interested needed to be determined, the location or bounding box of the object will also be decided. Because of the large number of possible sliding windows, many novel approaches have been proposed. For using convolutional neural network as the object classifier, the latest development involves several new object proposal algorithms. The first one is R-CNN (Region-based Convolutional Neural Network), which uses Selective Search, to generate about 2,000 ROI proposals. The number of ROIs in this case are much smaller than the sliding window exhaustive search, and each ROI has much higher possibility to contain the object of interest, thus it is a much faster algorithm. Then Fast CNN was proposed, which perform feature extraction before region proposal, and uses Softmax instead of SVM. Another improvement was made on Fast CNN, which results in an algorithm call Faster R-CNN. In this algorithm, selective search algorithm was replaced by a fast neural network in inserting a Region Proposal Network (RPN) to predict proposals from features. This RPN

Semantic and Instance segmentation

In object detection, instead of obtaining bounding boxes for objects, objects can be segmented at pixel level. This is especially useful in situations where the objects are cluttered or occluded, such as in the microscopic cell images. The current state of the art algorithm for instance segmentation Mask-CNN and FCN (Fully Convolutional Network). The original sliding windows suffers from the enormous amount of possible shapes and sizes. A more effective solution is to treat the window as an initial guess, then the class and boundary box are predicted from the current sliding window. Mask-RCNN is Faster-RCNN with a mask for pixel level object detection and segmentation. It is especially effective for medical images because the cells and tissues often are touching, cluttered, or occluded.

Object Detection

The current the state-of-the-art object detection algorithm is Faster R-CNN, which improves detection and recognition speed by basically sharing computation of the convolution layers between different object proposals and swapping the order of generating region proposals and running the CNN. In this model, the image is first fed through a ConvNet, features of the region proposals are obtained from the last feature map of the ConvNet, and lastly we have fully connected layers as well as our regression and classification heads. For computation efficiency, single shot detectors such as YOLO (You Only Look Once) or SSD (Single Shot Dectors) are developed; those methods generate object proposal windows and obtain class scores for these windows simultaneously.

6.2.2 Specialized Hardware

Specialized integrated circuits (IC) have been developed by major technology companies as well as start-ups to implement machine learning algorithms; with the general purpose ICs (CPU, Graphical Processing Unit, FPGA) reaching their performance potential, ASIC (Application Specific Integrated Circuit) is a natural evolution that implement common machine learning functions, such as back-propagation, stochastic gradient descent (SGD) algorithms.

6.2.3 Generative Model

Though Deep Neural Networks traditionally are mostly used for discriminant tasks such as classification, in recent years, new architecture and frameworks are developed more and more for generative tasks, such as generative new artworks which resembles an artist. Generative Adversarial Network (GAN), proposed by IAN, Goodfellow in 2014, is one of the most noticeable algorithms in recent years.

6.3 Future Research Works

The application of computer vision technology is rapidly spreading to many different areas in our lives and other academic subjects. In the last few years, the deep learning based algorithms and frameworks demonstrated their performance advantage over the traditional machine learning models with hand feature engineering for tasks such as image classification, object detection, natural language processing, etc. With the frameworks such as TensorFlow and PyTorch become more popular, in the foreseeable future, deep learning based algorithm will continue to dominant and they are currently the mainstream implementation mechanism for machine learning and computer vision related tasks.

On the other hand, simple rule based methods such as template matching continue to provide basic functions for simple computer vision applications, especially in controlled environments such as industry automation. These simple to develop and simple to deploy computer vision applications are gaining popularity in areas such as manufacturing automation and environment monitoring. A generic framework to detect and measure the 2D affine transform template matching application is under development and will be open sourced.

The next step of my research will mainly focus on two areas, application of machine learning techniques in medical research and using machine learning algorithms to improve semiconductor fabrication production. The new techniques are being developed to count the white blood cells in bone marrow using semantic segmentation methods such as Fully Convolutional Network; and in addition to analyzing medical images, we will use machine learning algorithms to optimize the treatment plan for follicular lymphoma patients. This tool will classify the major types and sub-types of WBCs and will be able to diagnose not only if Leukimia is positive or negetive, but also the severity of the disease, thus help pathologists to diagnose the disease more accurately and efficiently, and help physicians to treat the patients.

REFERENCES

[1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[2] Tobias Senst, Volker Eiselein, and Thomas Sikora. Robust local optical flow for feature tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(9):1377, 2012.

[3] Jun Cao, Yilin Wang, and Baoxin Li. Real-time vehicle back-up warning system with a single camera. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 2275–2279. IEEE, 2015.

[4] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[5] Mark Everingham, L Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge 2007 (voc 2007) results (2007), 2008.

[6] Massimo Bertozzi, Emanuele Binelli, Alberto Broggi, and MD Rose. Stereo vision-based approaches for pedestrian detection. In *Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on*, pages 16–16. IEEE, 2005.

[7] Christoph H. Lampert, M.B. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[8] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.

[9] David Crandall, Pedro Felzenszwalb, and Daniel Huttenlocher. Spatial priors for part-based recognition using statistical models. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 10–17. IEEE, 2005.

[10] Yali Amit and Alain Trouvé. Pop: Patchwork of parts models for object recognition. *International Journal of Computer Vision*, 75(2):267–282, 2007.

[11] Zhigang Tu, Jun Cao, Yikang Li, and Baoxin Li. Msr-cnn: applying motion salient region based descriptors for action recognition. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 3524–3529. IEEE, 2016.

[12] Guilhem Chéron, Ivan Laptev, and Cordelia Schmid. P-cnn: Pose-based cnn features for action recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 3218–3226, 2015.

[13] Andrea Vedaldi and Karel Lenc. Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 689–692. ACM, 2015.

[14] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *European conference on computer vision*, pages 25–36. Springer, 2004.

[15] Zhigang Tu, Nico Van Der Aa, Coert Van Gemeren, and Remco C Veltkamp. A combined post-filtering method to improve accuracy of variational optical flow estimation. *Pattern Recognition*, 47(5):1926–1940, 2014.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[17] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014.

[18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. Ieee, 2009.

[19] Georgia Gkioxari and Jitendra Malik. Finding action tubes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 759–768, 2015.

[20] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[21] Aurélie Bugeau and Patrick Pérez. Detection and segmentation of moving objects in complex scenes. *Computer Vision and Image Understanding*, 113(4):459–476, 2009.

[22] Zhi Gao, Loong-Fah Cheong, and Yu-Xiang Wang. Block-sparse rpca for salient motion detection. *IEEE transactions on pattern analysis and machine intelligence*, 36(10):1975–1987, 2014.

[23] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

[24] Zhouchen Lin, Minming Chen, and Yi Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.

[25] L Wixson. Detecting salient motion by accumulating directionally-consistent flow. *IEEE transactions on pattern analysis and machine intelligence*, 22(8):774–780, 2000.

[26] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013.

[27] Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Learning to track for spatio-temporal action localization. In *Proceedings of the IEEE international conference on computer vision*, pages 3164–3172, 2015.

[28] Mikel D Rodriguez, Javed Ahmed, and Mubarak Shah. Action mach a spatio-temporal maximum average correlation height filter for action recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[29] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 3192–3199, 2013.

[30] Kunio Doi. Computer-aided diagnosis in medical imaging: historical review, current status and future potential. *Computerized medical imaging and graphics*, 31(4-5):198–211, 2007.

[31] Krishna Mohanta and V Khanaa. An efficient contrast enhancement of medical x-ray images-adaptive region growing approach. *International Journal of Engineering and Computer Science*, 2(02), 2013.

[32] Rasool Fakoor, Faisal Ladhak, Azade Nazi, and Manfred Huber. Using deep learning to enhance cancer diagnosis and classification. In *Proceedings of the International Conference on Machine Learning*, volume 28. ACM New York, USA, 2013.

[33] Ridhi Jindal and Sonia Vatta. Sift: scale invariant feature transform. 2010.

[34] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.

[35] David L Donoho. For most large underdetermined systems of linear equations the minimal 1-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(6):797–829, 2006.

[36] Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[37] HE Diff-Quick. Gastritis/gastropathy: Perspective of the pathologist. 2014.

[38] Stephane Lavallee and Philippe Cinquin. Computer assisted medical interventions. In *3D imaging in medicine*, pages 301–312. Springer, 1990.

[39] Baopu Li and Max Q-H Meng. Computer-aided detection of bleeding regions for capsule endoscopy images. *IEEE Transactions on biomedical engineering*, 56(4):1032–1039, 2009.

[40] Yang Cong, Shuai Wang, Ji Liu, Jun Cao, Yunsheng Yang, and Jiebo Luo. Deep sparse feature selection for computer aided endoscopy diagnosis. *Pattern Recognition*, 48(3):907–917, 2015.

[41] Baopu Li and Max Q-H Meng. Tumor recognition in wireless capsule endoscopy images using textural features and svm-based feature selection. *IEEE Transactions on Information Technology in Biomedicine*, 16(3):323–329, 2012.

[42] C-R Huang, B-S Sheu, P-C Chung, and H-B Yang. Computerized diagnosis of helicobacter pylori infection and associated gastric inflammation from endoscopic images by refined feature selection using a neural network. *Endoscopy*, 36(07):601–608, 2004.

[43] Chun-Rong Huang, Pau-Choo Chung, Bor-Shyang Sheu, Hsiu-Jui Kuo, et al. Helicobacter pylori-related gastric histology classification using support-vector-machine-based feature selection. *IEEE Transactions on Information Technology in Biomedicine*, 12(4):523–531, 2008.

[44] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, Sabine Süsstrunk, et al. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.

[45] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[46] Mingkui Tan, Li Wang, and Ivor W Tsang. Learning sparse svm for feature selection on very high dimensional datasets. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 1047–1054, 2010.

[47] Shuai Wang, Yang Cong, Jun Cao, Yunsheng Yang, Yandong Tang, Huaici Zhao, and Haibin Yu. Scalable gastroscopic video summarization via similar-inhibition dictionary selection. *Artificial intelligence in medicine*, 66:1–13, 2016.

[48] Meng Yang, Lei Zhang, Xiangchu Feng, and David Zhang. Fisher discrimination dictionary learning for sparse representation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 543–550. IEEE, 2011.

[49] Thomas Brox and Jitendra Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE transactions on pattern analysis and machine intelligence*, 33(3):500–513, 2011.

[50] Naveed Ejaz, Irfan Mehmood, and Sung Wook Baik. Efficient visual attention based framework for extracting key frames from videos. *Signal Processing: Image Communication*, 28(1):34–44, 2013.

[51] Qing Xu, Yu Liu, Xiu Li, Zhen Yang, Jie Wang, Mateu Sbert, and Riccardo Scopigno. Browsing and exploration of video sequences: A new scheme for key frame extraction and 3d visualization using entropy based jensen divergence. *Information Sciences*, 278:736–756, 2014.

[52] Jiebo Luo, Christophe Papin, and Kathleen Costello. Towards extracting semantically meaningful key frames from personal video clips: from humans to computers. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(2):289–301, 2009.

[53] Richard M Jiang, Abdul H Sadka, and Danny Crookes. Hierarchical video summarization in reference subspace. *IEEE Transactions on Consumer Electronics*, 55(3), 2009.

[54] Youssef Hadi, Fedwa Essannouni, and Rachid Oulad Haj Thami. Video summarization by k-medoid clustering. In *Proceedings of the 2006 ACM symposium on Applied computing*, pages 1400–1401. ACM, 2006.

[55] Yang Cong, Junsong Yuan, and Jiebo Luo. Towards scalable summarization of consumer videos via sparse dictionary selection. *IEEE Transactions on Multimedia*, 14(1):66–75, 2012.

[56] Dinggang Shen, Guorong Wu, and Heung-Il Suk. Deep learning in medical image analysis. *Annual review of biomedical engineering*, 19:221–248, 2017.

[57] Andre Esteva, Brett Kuprel, Roberto A Novoa, Justin Ko, Susan M Swetter, Helen M Blau, and Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 542(7639):115, 2017.

[58] Xin Yi, Ekta Walia, and Paul Babyn. Generative adversarial network in medical imaging: A review. *arXiv preprint arXiv:1809.07294*, 2018.

[59] Emmanouil Michail, Evgenios N Kornaropoulos, Kosmas Dimitropoulos, Nikos Grammalidis, Triantafyllia Koletsa, and Ioannis Kostopoulos. Detection of centroblasts in h&e stained images of follicular lymphoma. In *Signal Processing and Communications Applications Conference (SIU), 2014 22nd*, pages 2319–2322. IEEE, 2014.

[60] Elaine S Jaffe. The 2008 who classification of lymphomas: implications for clinical practice and translational research. *ASH Education Program Book*, 2009(1):523–531, 2009.

[61] Alican Bozkurt, Alexander Suhre, and A Enis Cetin. Multi-scale directional-filtering-based method for follicular lymphoma grading. *Signal, Image and Video Processing*, 8(1):63–70, 2014.

[62] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, and Clara I Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017.

[63] Olcay Sertel, Jun Kong, Umit V Catalyurek, Gerard Lozanski, Joel H Saltz, and Metin N Gurcan. Histopathological image analysis using model-based intermediate representations and color texture: Follicular lymphoma grading. *Journal of Signal Processing Systems*, 55(1-3):169, 2009.

[64] Di Jia, Jun Cao, Wei-dong Song, Xiao-liang Tang, and Hong Zhu. Colour fast (cfast) match: fast affine template matching for colour images. *Electronics Letters*, 52(14):1220–1221, 2016.

[65] Simon Korman, Daniel Reichman, Gilad Tsur, and Shai Avidan. Fast-match: Fast affine template matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2331–2338, 2013.

[66] AA Royem, CK Mui, DR Fuka, and MT Walter. Proposing a low-tech, affordable, accurate stream stage monitoring system. *Transactions of the ASABE*, 55(6):2237–2242, 2012.

APPENDIX A

RELATED PUBLICATIONS IN DISSERTATION

- Real-time Vehicle Back-up Warning System with a Single Camera Cao, Jun, Yilin Wang, and Baoxin Li. ”Real-time vehicle back-up warning system with a single camera.” Image Processing (ICIP), 2015 IEEE International Conference on. IEEE, 2015.
- Variational method for joint optical flow estimation and edge-aware image restoration, Tu, Z., Xie, W., Cao, J., Van Gemeren, C., Poppe, R., Veltkamp, R. C. (2017). Variational method for joint optical flow estimation and edge-aware image restoration. Pattern Recognition, 65, 11-25.
- MSR-CNN: Applying Motion Salient Region Based Descriptors for Action Recognition, Tu, Zhigang, Jun Cao, Yikang Li, and Baoxin Li. ”MSR-CNN: applying motion salient region based descriptors for action recognition.” In Pattern Recognition (ICPR), 2016 23rd International Conference on, pp. 3524-3529. IEEE, 2016.
- Scalable gastroscopic video summarization via similar-inhibition dictionary selection, Wang, Shuai, Yang Cong, Jun Cao, Yunsheng Yang, Yandong Tang, Huaici Zhao, and Haibin Yu. ”Scalable gastroscopic video summarization via similar-inhibition dictionary selection.” Artificial intelligence in medicine 66 (2016): 1-13.
- Deep sparse feature selection for computer aided endoscopy diagnosis, Cong, Yang, Shuai Wang, Ji Liu, Jun Cao, Yunsheng Yang, and Jiebo Luo. ”Deep sparse feature selection for computer aided endoscopy diagnosis.” Pattern Recognition 48, no. 3 (2015): 907-917.
- Jia, Di, Jun Cao, Wei-dong Song, Xiao-liang Tang, and Hong Zhu. ”Colour FAST (CFAST) match: fast affine template matching for colour images.” Electronics Letters 52, no. 14 (2016): 1220-1221.
- Deep Learning Based Tissue Level Grading of Follicular Lymphoma, Presented at 2018 The United States and Canadian Academy of Pathology annual conference in Vancouver, Canada.