Plant Seed Identification

A Thesis Submitted to the
College of Graduate Studies and Research
in Partial Fulfillment of the Requirements
for the degree of Doctor of Philosophy
in the Department of Computer Science
University of Saskatchewan
Saskatoon

By
Xin Yi

Permission to Use

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan
Canada
S7N 5C9

Abstract

Plant seed identification is routinely performed for seed certification in seed trade, phytosanitary certification for the import and export of agricultural commodities, and regulatory monitoring, surveillance, and enforcement. Current identification is performed manually by seed analysts with limited aiding tools. Extensive expertise and time is required, especially for small, morphologically similar seeds. Computers are, however, especially good at recognizing subtle differences that humans find difficult to perceive. In this thesis, a 2D, image-based computer-assisted approach is proposed.

The size of plant seeds is extremely small compared with daily objects. The microscopic images of plant seeds are usually degraded by defocus blur due to the high magnification of the imaging equipment. It is necessary and beneficial to differentiate the in-focus and blurred regions given that only sharp regions carry distinctive information usually for identification. If the object of interest, the plant seed in this case, is in-focus under a single image frame, the amount of defocus blur can be employed as a cue to separate the object and the cluttered background. If the defocus blur is too strong to obscure the object itself, sharp regions of multiple image frames acquired at different focal distance can be merged together to make an all-in-focus image. This thesis describes a novel non-reference sharpness metric which exploits the distribution difference of uniform LBP patterns in blurred and non-blurred image regions. It runs in realtime on a single core cpu and responses much better on low contrast sharp regions than the competitor metrics. Its benefits are shown both in defocus segmentation and focal stacking.

With the obtained all-in-focus seed image, a scale-wise pooling method is proposed to construct its feature representation. Since the imaging settings in lab testing are well constrained, the seed objects in the acquired image can be assumed to have measureable scale and controllable scale variance. The proposed method utilizes real pixel scale information and allows for accurate comparison of seeds across scales. By cross-validation on our high quality seed image dataset, better identification rate (95%) was achieved compared with pre-trained convolutional-neural-network-based models (93.6%). It offers an alternative method for image based identification with all-in-focus object images of limited scale variance.

The very first digital seed identification tool of its kind was built and deployed for test in the seed laboratory of Canadian food inspection agency (CFIA). The proposed focal stacking algorithm was employed to create all-in-focus images, whereas scale-wise pooling feature representation was used as the image signature. Throughput, workload, and identification rate were evaluated and seed analysts reported significantly lower mental demand ($p = 0.00245$) when using the provided tool compared with manual identification. Although the identification rate in practical test is only around 50%, I have demonstrated common mistakes that have been made in the imaging process and possible ways to deploy the tool to improve the recognition rate.

Acknowledgements

Undertaking this PhD in image processing and computer vision has been a truly life-changing and challenging experience for me and it would not have been possible to achieve it without the support and guidance that I received from many people.

First and foremost I want to express my special appreciation and thanks to my supervisor, professor Mark G. Eramian. Mark has always been the source of encouragement and inspiration. I am very grateful for his patience and willingness to explain every detail of algorithms and procedures. He has taught me how to think like a scientist, how to act like a scientist, and how to work with scientists. Interacting with him has been a great pleasure and it will certainly be missed and cherished.

I would like to thank Dr. Ruojing Wang who brought in this wonderful project and always being so supportive of my work. I also want to acknowledge the effort and constructive comments of my other committee members, Dr. Eric Neufeld, Dr. Michael C. Horsch, Dr. Francis M. Bui, Dr. Tony Kusalik.

Special thanks also goes to Canadian National Seed Herbarium for providing seed specimens, Jennifer Neudorf for taxonomy advice, Jo Jones for the images used in this study, Angela Salzl for her assistance in the user study. Thank those six seed analysts who generously volunteered to participate in the validation study. Without their cooperative work, this thesis would not be possible to finish.

The image processing group has been a second home for me. I am especially grateful for the group members who stuck it out in grad school with me, Jianning Chi, Rafizul Haque, Ekta Walia, Abdullah Chisti. I will clearly miss the time discussing nonsensical ideas and ranting about random things with them.

I am indebted to all my friends and those who opened their homes to me during my time at Saskatoon and were always so helpful in numerous ways. Special thanks to Tate Cao, Leon and Jenny Stein, Stewart Fehr.

I would like to thank my parents Ruiyu Li and Xianming Yi. They are always there encouraging me to follow my dreams. Without their unwavering support, unconditional love and trust I would not have been able to accomplish anything, let alone research.

The best outcome from these past four years is finding my best friend, soul-mate, wife – Rui Fang. I am grateful to have her on my side, living every single minute of it even when I was irritable and depressed.

This thesis is dedicated to my beloved wife and my parents.

Contents

List of Tables

List of Figures

List of Abbreviations

CFIA	Canadian Food Inspection Agency
LBP	Local Binary Pattern
PSF	Point Spread Function
CRF	Conditional Random Field
CRF	Camera Response Function
SIFT	Scale Invariant Feature Transform
ILSVRC	The ImageNet Large Scale Visual Recognition Challenge
SVM	Support Vector Machine
BoW	Bag of Words
LIOP	Point Spread Function
GLOH	Gradient Location Orientation Histogram
SURF	Speeded Up Robust Features
GMM	Gaussian Mixture Model
HoG	Histogram of Gradients
POOF	Part-based One-vs-One Features
CNN	Convolutional Neural Network
ReLU	Rectified hyperbolic tangent
DoG	Difference of Gaussian
DSIFT	Dense SIFT
SVD	Singular Value Decomposition
LDA	Linear Discriminant Analysis
TLX	Task Load Index
LTP	Local Ternary Pattern
NRLBP	Noise-Resistant LBP
RNN	Recurrent Neural Network
VRLM	Virtual Reflected-light Microscopy
NLM	None Local Mean

Chapter 1

Introduction

1.1 Plant Seed Identification: An Important and Challenging Field

Invasion of plants into a new area, either local or across continents, are mainly accomplished by the dispersal of plant seeds. The frequent commercial trade nowadays and other human activities substantially facilitate this process, with a consequence of changing the distribution of non-native species in many regions [30]. Early detection of the seeds of noxious weeds and invasive plants that contaminate agricultural products during trade activities is the most cost-effective measures for weed and invasive plant control [72, 4]. In addition to that, successful identification of seed could also provide valuable information in forensic science [20], food science [157], archaeology [9], and ecology [189].

As a specialized area of botany, seed identification has a history of over a century [124]. The difficulty of the identification varies and is strongly dependent on the specificity of the task. Normal people without any training would have no problem differentiating a corn seed and sunflower seed. But down to species level, there are many cases where seeds of one species may closely resemble the other [111]. Problem arises when one of these may be a crop plant and the other an undesirable noxious weed. Inability to screen weed seeds out could result in crop yield reduction because weeds can compete with desirable crop plants for water, light, and nutrients. For example, green foxtail (*Setaria italica viridis*) is considered as a regional noxious weed in British Columbia. Giant foxtail (*Setaria faberi*) is another noxious weed reduces crop yields by 13–14% on average plant distributions [60]. Foxtail millet (*Setaria italica italica*) is a food crop and is mainly consumed in Northern China [195]. These three seed species share very similar morphological features with example images shown in Figure 1.1. The trained seed analyst must be able to analyze and evaluate the morphologically similar seed structures of such seeds and make decisions with limited evidence provided by the seed alone [124]. Despite the importance of accurately identifying invasive or noxious weed seeds, it can be very challenging to identify morphologically similar species especially when they have a typical size of only a grain of salt.

Therefore, in this thesis we study the problem of identification of morphological similar seeds.

(a) Giant foxtail (*Setaria faberi*) (b) Green foxtail (*Setaria italica viridis*)(c) Foxtail millet (*Setaria italica italica*)

Figure 1.1: Morphologically similar seed examples shown in the left corner. Giant foxtail and green foxtail are considered as noxious weeds whereas foxtail millet is a critical food crop. Figure (a) by Kropsoq is licensed under CC BY-SA 3.0. Figure (b) by bastus917 is licensed under CC BY-SA 2.0. Figure (c) by STRONGlk7 is licensed under CC BY-SA 3.0.

1.2 Botanical Nomenclature

In order to communicate among people form different regions of the world without involving language and cultural difference, scientists have agreed upon a naming convention based primarily on Latin [3]. In this thesis, all seeds will be referred with their scientific name (or the abbreviations) to prevent any confusion. In the next paragraph I review some basic rules for the composition of scientific names.

"The Latin portion of the scientific name is italicized or underlined in print and underlined when handwritten; the first letter of the genus name is always capitalized and all letters of the specific epithet are lowercase" [3]. Genera names are monomials (e.g. *Setaria*), species names are binomials (a combination of the genus plus a specific epithet, such as *Setaria faberi*), subspecies and botanical varieties are trinomials (e.g. *Setaria italica viridis*). For detailed principles, rules and recommendations regarding scientific names, readers are recommended to go to the International Code of Botanical Nomenclature [59] for more information.

1.3 Motivation for a Computerized Solution

Protection of the plant production base and plant health is the commission of The Canadian Food Inspection Agency (CFIA). As a critical diagnostic test of agricultural products within CFIA, identifying seed especially noxious weeds, is conducted routinely for seed trade and phytosanitary certification in both domestic and international trade. Therefore, the capacity of accurately and rapidly identifying weed seeds directly affects the monitoring, surveillance and enforcement of plant health related regulations and policies, such as Weed Seed Order, Seed Act and Seed Regulations, and policies on Canada regulated plant species.

Currently the identification is performed by trained seed analysts through manual inspection of morphological features under a low magnification microscope. The inspection is mainly centred on assessment of qualitative characteristics, including shape, colour and surface texture. The seed size is the only quantitative

data that has been used for diagnosis. Texture patterns of seed are often complex and can be attributed to many different reasons. For example, in the process of harvesting and cleaning, seeds can be deceased, shriveled, and lost some of its accessory parts or even been damaged. Different origins, varieties, and even the degree of maturity could also have altered its general appearance [124]. All these variations pose huge identification challenges for seed analysts. Furthermore, if taking extensive interregional and international movement of seeds into consideration, seed analysts are expect to have experiences on both the local and worldwide seed species [124]. While the manual identification process works, it is usually time consuming and depends on considerable worker proficiency. A certified seed analysts usually requires at least 1500 hours of training to be eligible for real-world testing.

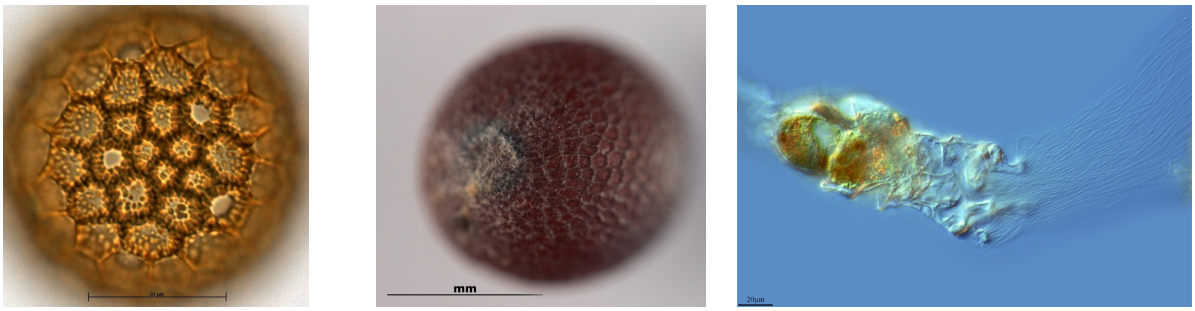
There are references that the seed analyst could resort to for further assistance, including known seed specimens, written descriptions, taxonomic identification keys, and reference books. One literature worths mentioning here is written by Jensen et al. [77]. It is a comprehensive review on seed morphology, covering handbooks, monographs, and articles, and is considered to be very useful for seed identification. Whenever a possible answer has been achieved, the test seed need to be compared against a known specimen for confirmation. If the final determination exceeds the seed analyst's level of confidence, the best practice is to forward the specimen to a person with more diagnostic expertise [3].

As can be concluded from above, the problem facing plant seed identification is two-fold: classification is labour intensive, and a huge amount of taxonomic work must be performed on a routine basis. Since plant seed identification is of such importance to society, solutions should be explored to help overcome the issues that CFIA faces today.

Developing machines to identify plant species from their DNA, also called DNA "barcoding" has been proposed as an approach to conquer this problem [127]. Although the initiatives have caught the public's attention, the generality and reliability of this technology is still waiting to be further confirmed. In this thesis, an alternative approach using 2D colour images is explored.

1.4 Image Based Identification

Image-based object class identification is a subroutine of object recognition. It is basically a multi-class classification problem and serves as the basis for higher-level computer vision tasks, e.g. automatic image captioning [179], autonomous driving [69]. While there have been many progresses in this field recently, it still remains one of the most challenging problems in computer vision because of the innumerous combinations of variations (clutter, occlusion, pose and scale changes, etc) that could possibly occur in a single image [201]. The seed identification problem falls into the scope of fine-grained object identification, which aims at distinguishing among subordinate categories of the same generic object class. Similar problems are identifying specific types of birds, motorcycles, airplanes which are only recognizable by people with certain amount of domain expertise.



(a) Pollen [1] **(b)** Plant seed [148] **(c)** Microorganism [143]

Figure 1.2: Microscopy images of minute objects. Figure (a) by Australasian Pollen and Spore Atlas is licensed under CC BY-SA 3.0. Figure (c) by Proyecto Agua is licensed under CC BY-SA 2.0.

Over the last decade, object identification has undergone rapid changes and progresses, with the advances largely concentrated on distinguishing between basic-level objects that are easy for humans to recognize, e.g. car, boat, chair, plane, etc. Challenges are hosted every year for evaluating object identification algorithms proposed by researchers, such as the PASCAL VOC challenge (20 classes) and the ImageNet Large Scale Visual Recognition Challenge (ILSVRC, 1000 classes). Based on the recent ILSVRC results, fine-grained identification is still the bottleneck of current identification methods [149].

Despite the recent high volume of research trying to solve this problem, the progress achieved has mostly been for in-focus natural images (at least the object to be recognized is in-focus) [17, 18, 178, 107]. Major sources of such images are point-and-shoot cameras and cell phones where large depth-of-field can be obtained by focusing at the hyperfocal distance. In seed testing, however, samples are too small to be captured by these portable devices. Dedicated equipment such as a microscope with high magnification is required to get clear texture representation of the seed surface.

In fact, many other similar areas suffer from the same problem, such as pollen studies, environmental monitoring, and microfossil identification in biostratigraphy, to name just a few. Some of them also have an identification requirement on a research or work basis. For example, in pollen studies, the utility and structure of pollen grains need to be analyzed to determine the plant relationships; in biostratigraphy, microfossil samples are key to providing vital information in understanding prehistoric climate [54].

Figure 1.2 gives an example of images of these small objects. It can be clearly seen that such microscopy images are always suffering from huge amount of defocus blur due to the optical limits. Another common characteristic shared by these images is they all have the scale bar which tells the viewer the actual size of the specimen. With these two distinctive image characteristics, it would be intuitive to ask:

1. ***How to robustly and reliably separate the in- and out-of-focus regions of the image given that only in-focus regions carry image details that are useful for identification?***

2. ***Can these scales be easily incorporated into the identification model and give a better feature represenation?***

The first question is related to the fundamental image acquisition and the second question involves the extension of the current identification model. Successful separation of in- and out-of-focus regions can possibly lead to two approaches to manipulate the underlying image. One would be directly using a single image frame if the seed sample is in-focus but use the defocus blur as a cue to separate the seed from the potentially cluttered background. The other would be to use the in-focus regions of multiple image frames acquired at different focal distance and fuse them together as the input for the identification.

1.5 Overview of Techniques and Contributions

The overall objective of this research is to find the software solutions to address these questions and apply them to the plant seed identification problem. The contributions of the thesis can be summarized as follows:

1.5.1 Chapter 3: Defocus Blur Segmentation

In this chapter, I proposed a sharpness metric based on local binary patterns (LBP) and a robust segmentation algorithm to separate in- and out-of-focus image regions. The proposed sharpness metric exploits the observation that most local image patches in blurry regions have significantly fewer of certain local binary patterns compared to those in sharp regions. It runs in realtime on a single core cpu and responses much better on low contrast sharp regions. Moreover, it can not only be used for defocus blur segmentation, but also can be used for online focal stacking to creating all-in-focus images. A defocus segmentation algorithm is proposed based on this sharpness metric together with image matting and multi-scale inference. Hundreds of partially blurred images are used to evaluate the proposed segmentation algorithm and five state-of-the-art comparator methods. The results show that this algorithm achieves a higher precision at high levels of recall than the comparators.

This novel metric has also been integrated into the proposed online focal stacking algorithm, which does not require stacks of images been captured before hand. It has achieved comparable results with the state-of-the-art under low noise condition but with less computation complexity.

The defocus blur segmentation method has already been published in IEEE Transaction on Image Processing and the code can be downloaded in the project page [1].

1.5.2 Chapter 4: A New Mid-level Feature for Textured Objects of Known Scale

A scale-wise pooling representation was proposed as the extension of the currently popular spacial pyramid matching scheme in the scale dimension by utilizing real pixel scale information. With representative specimens, the proposed representation described herein can achieve a high recognition rate of 95% using only texture features (no colour- or shape-based features) which is superior compared with the standard ob-

[1] https://www.cs.usask.ca/faculty/eramian/defocusseg/

ject recognition pipeline and pre-trained convolutional-neural-network-based models. It offers an alternative method for image based identification with in-focus object images of limited variance in scale.

A part of this chapter was submitted to Machine Vision and Application and major revision was requested.

1.5.3 Chapter 5: User Study

This chapter focuses on the evaluation of the effectiveness of the above proposed techniques in practical seed identification. The very first digital seed identification tool of its kind was built for plant seed identification based on realtime focal stacking and scale-wise pooling representation mentioned in Chapter 3 and 4. This tool was deployed for testing in a seed testing laboratory located in Saskatoon. Currently, seed analysts in this lab recognize large amount of plant seeds on a daily basis manually with limited assistive tools. A user study (ethics approval certificate #BEH-15-293) was conducted here to evaluate the impact of the aiding tool in practice. Throughput, recognition rate, workload was evaluated. Participants reported significantly lower mental demand by using this tool compared with conventional manual operations.

1.5.4 Chapter 6: Conclusions and Future Works

In the final chapter I conclude the thesis and discuss possible topics for future research.

Chapter 2

Literature review

Given the long lasting problem of seed identification, there have been limited attempts trying to solve it through computer vision. Granitto et al. [56, 57] conducted a series of work on seed identification using image analysis and automatic identification. Their database contains 236 different weed species. Using 12 features that consisting of morphological, colour and texture information, and using a ANN (Artificial Neural Network) classifier, their test image were assigned to the correct class at a rate of $92.5 \pm 0.4\%$. Their 12 morphological features included measurements such as seed area, compactness, and moments of planar mass distribution. These were found to be nearly optimal for their data set using the performance of a Naive Bayes classifier as the feature selection criterion. They also stated that morphological features have the largest discriminating power, that colour is not particularly good because many species are light to dark brownish or black, and that texture characteristics are even less reliable as classification parameters.

However, texture has been shown to be much more promising than colour and shape for classifying the morphological similar seeds according to the results of our pilot study [197]. In that study, the same set of features as in [56, 57] were extracted and evaluated on a subset of images presented in this thesis (that is all we have back then). Classification results demonstrate that their proposed features are not effective on differentiating the morphological similar seeds, which implicitly shows the difficultly of our task and suggests more discriminative texture descriptor has to be sort. As for the other seed identification works, they either focused on one particular morphological pattern, e.g. position of the umbilical, or were tested on data sets with very large inter-species variance [63, 113, 31, 204]. Therefore, in this thesis we only use surface textures for the identification as similar to the other object recognition systems [17, 18, 178, 107]. Another observation from the preliminary work is that, by only using texture feature, the confusion is only happened among seeds that share similar morphological features. This finding motivate us to use the real scale information to build more precise texture feature representation as will be discussed in Chapter 4.

Due to the restricted volume of work explicitly related to image based seed identification, the related works reviewed in this chapter are mostly centred on general purpose techniques for object identification and only techniques using image texture are considered. The following chapter is divided into three major sections. Section 2.1 is an introduction to basic-level object identification. Section 2.2 discusses the traditional fine-grained object identification. Finally, section 2.3 describes the recent deep neural network based approaches where image representation and classifier are both learned in the training process.

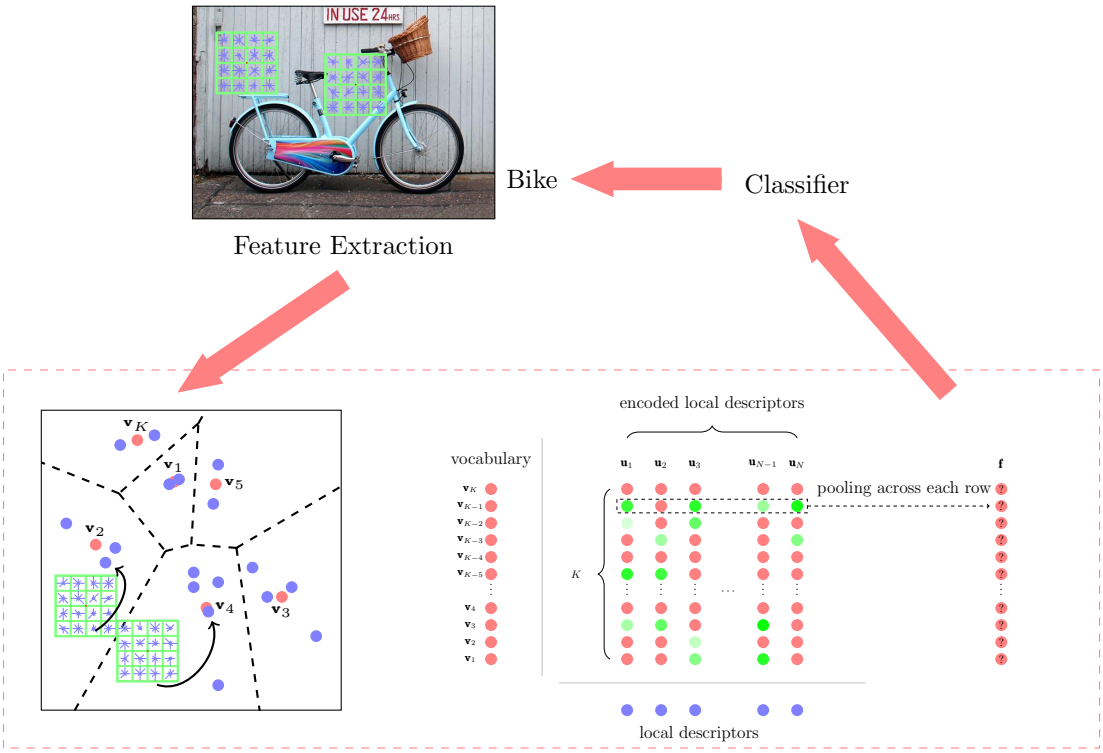


Figure 2.1: An example of a conventional visual identification model. Bike image is from ImageNet dataset [150]. This Figure is a reproduction of a Figure by Chatfield et al. [32].

2.1 Basic-level Object Identification

The Bag-of-Words (BoW) model [37] that was borrowed from natural language processing [80, 171, 110, 36], is commonly used in traditional object class identification systems. Although many variations of this model exist, the fundamental structure remains the same which can be summarized by Figure 2.1.

1. **Feature Extraction**

   First, local image descriptors are extracted from images equally chosen from each object class. These descriptors can be either built on a dense spacial grid [100, 190] or sparsely on keypoints detected by various kinds of detectors, e.g. Harris detector (corners) [61], Hessian detector (blobs) [121, 112], or even randomly chosen [129]. An example is shown in Figure 2.2 where local descriptors are extracted.

   These descriptors can either based on gradients, e.g. SIFT [112], GLOH [122], or wavelet coefficients, e.g. SURF [13], or intensity orders, e.g. LIOP [187]. Among all these descriptors, SIFT is still the most commonly used because of its balance between distinctiveness and computational efficiency. These low-level descriptors transforms the raw pixel intensities into a representation, to some extent, invariant to image variations, i.e. rotation, scale change, etc. If compared with the convolutional neural network model as will be reviewed in section 2.3, these descriptors can be treated as hand-crafted stages in the

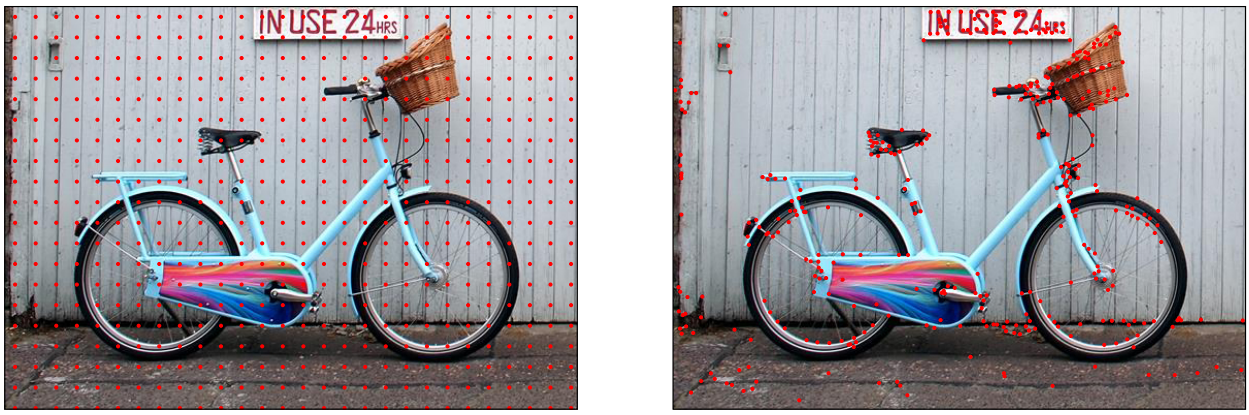


Figure 2.2: Local features are extracted in the local neighbourhood of these highlighted red points. In the left figure, points are aligned on a dense grid whereas in the right figure, points are extracted by Harris corner detector and scattered sparsely and irregularly.

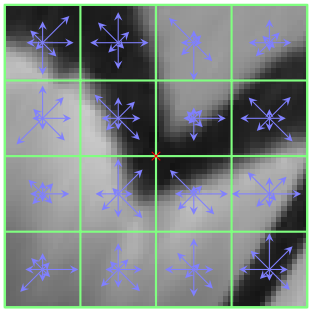


Figure 2.3: An illustration on how SIFT is built, a reproduction of Figure 7 of Lowe et al. [112]. The neighbourhood of the keypoint is divided into $4 \times 4 = 16$ spatial bins. The size of the spacial bin is proportional to the scale of the keypoint. Inside each subregion is the histogram of gradient orientations quantized into 8 bins. The final descriptor is the concatenation of histograms of each subregion.

feed-forward architecture [95].

2. **Vocabulary Building**

   Next, a visual vocabulary, also known as a dictionary/codebook, is learned through one of several clustering methods, e.g. K-means or Gaussian Mixture Model (GMM) [147]. Each cluster centre is referred as a visual word/code. By making an analogy to text document classification, each local descriptor simply plays the role of a text word if we treat image as a text document.

3. **Feature Encoding**

   Feature coding is then performed by encoding each local descriptor with the learned vocabulary into a so-called mid-level representation (since it lies in the middle of low-level feature (e.g. SIFT) and the final representation (sent to classifier)). In the basic model described in [37], each local descriptor is quantized to the nearest word. In more recent works, in order to decrease the quantization errors, many other encoding methods are proposed, such as super-vector encoding [207], Fisher encoding [138,

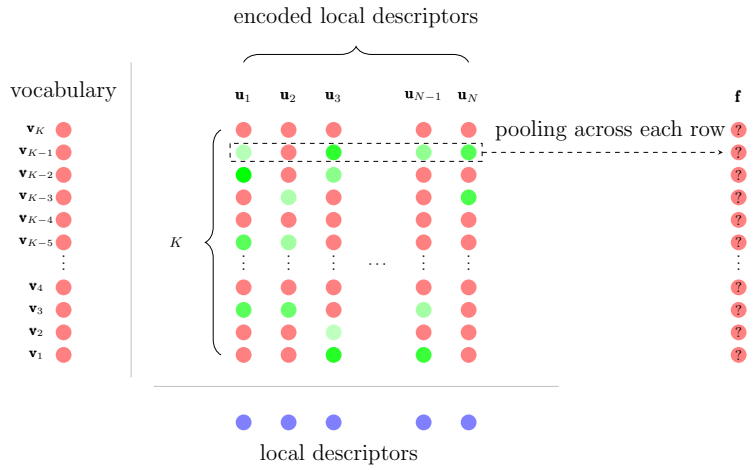


Figure 2.4: Demonstration of encoding and pooling. Each word in the vocabulary has a dimension size D. Green dots are words used to reconstruct the descriptor which are designated by encoding methods. Red dots represent words that are not used. The brightness of green dot demonstrates the weight of each corresponding word. Pooling methods decide how all encoded descriptors are aggregated together. It is pooled on a row basis such that each row of the descriptors is pooled into a single value. Note that in this figure I used sparse encoding as an example for the sake of visualization simplicity since the dimension of $\mathbf{f}$ in this case is the same as K. The actual dimension of $\mathbf{f}$ varies from method to method and does not necessarily need to be K (Fisher encoding produce a $\mathbf{f}$ with dimension of $2DK$).

76, 152], sparse coding [194], and locality-constrained linear encoding [185]. Based on experiments in [33, 87, 186, 73], the best mid-level feature for basic-level object identification is Fisher coding since the GMM used in Fisher coding is more robust given the learned density distribution. Moreover, Fisher coding preserves much more information, e.g. the mean and the variance of clusters. These claims are supported by its excellent performance in the ImageNet classification and localization challenge of 2012 [149].

4. **Feature Pooling**

   A pooling step is carried out to aggregate mid-level features from an image into a final representation with a fixed length vector as shown in Figure 2.4. Boureau et al. [25] have conducted a theoretical analysis on average pooling and max pooling. The results indicated that max pooling is better fitted for sparse features than average pooling. Furthermore, in [87], the author compared other more complex pooling strategies, such as MaxExp, Gamma, AxMin, and ExaPro and showed that they improved the performance over the baseline Max-pooling scheme but need more computations.

It can be noticed from the above summation that BoW model treats images as collections of independent patches thus discards spatial arrangement information. To recover the lost spatial information between patches, Lazebnik et al. proposed to pool across image subregions [97] which is known as pyramid matching, whereas Russakovsky et al. proposed to pool in an object-centric way [151]. In the latter case, the encoded features for object-of-interest and background are pooled separately and the final representation is the concatenation of features for these two different subregions.

The BoW model can not only be used for differentiating object classes but also for differentiating different semantic attributes (attributes now serve as object classes in this case) which is critical for recognition of subordinate categories of generic objects as will be seen in the next section.

2.2 Fine-grained Object Identification

Before fine-grained identifaction was recognized as a distinct problem from conventional basic-level identification, many researchers already adopted the above conventional basic-level approach to solve fine-grained recognition problems. For example, Larios et al. used three different region detectors and SIFT descriptor to recognize stonefly larvae [96]. Nilsback et al. used bag-of-SIFT to describe the texture, bag-of-histogram-of-gradients (HoG) [38] to describe shape of the boundary, bag-of-colour in HSV colour space to describe colour, and a multi-kernel support vector machine (SVM) on top to recognize flower species [128]. But since the difference among fine-grained objects is subtle, detecting and describing object attributes and parts have become increasingly important. In the following, I will review the traditional approaches for identification of fined-grained objects. They can be roughly categorized into the following four groups.

1. **Incorporate Humans into the Loop**

   These kinds of systems are semi-automatic methods which require humans to provide extra information to narrow down the possible answer space just like the classic 20 questions game but in a visualized fashion [17, 182, 26]. For example, when classifying an image of a bird, the human might provide the beak's location via clicking, or providing the pattern of the wing via a binary question: "Is the wing pattern striped?" [182]. An example workflow of bird species identification is shown in Figure 2.5. Such systems do not require experts, e.g. ornithologists, to perform the task since an average human being is capable of detecting and broadly categorizing objects or describing colour and shape, even if he/she does not recognize the object's identity.

2. **Attribute-based Approaches**

   An attribute is, in general, a semantic connotation that can be shared among object categories, instances, and parts, e.g. the greenness of a leaf or the sharpness of an edge. By characterizing objects with attributes, we can focus on descriptive properties of objects rather than their compositional and local traits [178]. Several authors have investigated attribute-based recognition [44, 91, 95]. They learned discriminative models from suitable attribute-labeled training data as shown in Figure 2.6 and subsequently applied the learned models to the test image to estimate the presented visual attributes. Class labels are inferred by combining the predictions of many attributes via Bayes approaches. Since visual attributes are human interpretable, successful detection of attributes would in the mean time enable other interesting applications [178], such as automatic image descriptions generation [45] or content-based image searching [16]. However, the richly-annotated data required for training is not

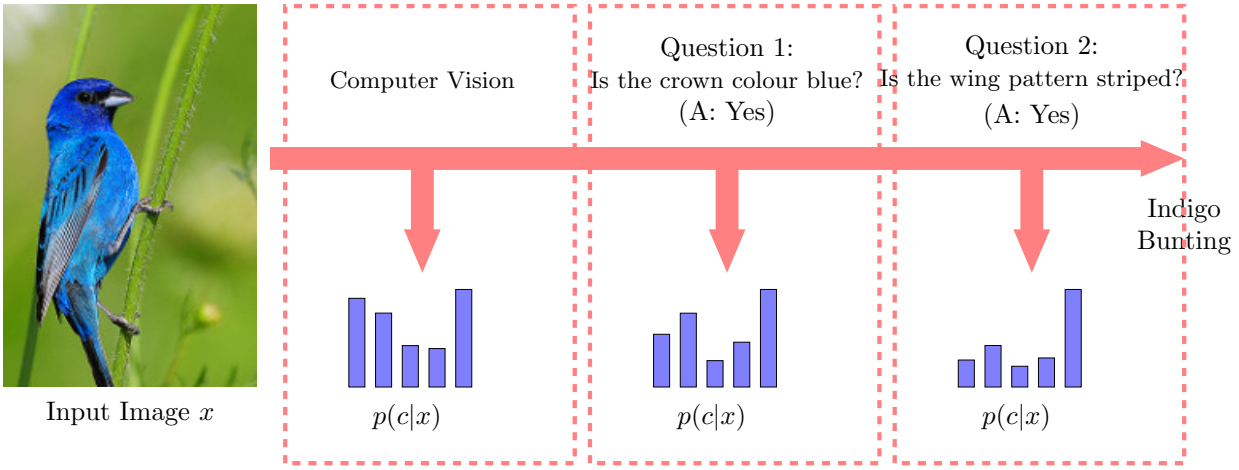


Figure 2.5: One example of the workflow of human involved fine-grained recognition system, a reproduction of Figure 10 of Branson et al. [26]. $p(c|x)$ is the possibility of image x assigned to class c. Input image is from dataset Caltech-UCSD Birds-200-2011 [183].

Figure 2.6: Exemplar images annotated in detail for training attribute detectors. Image is from dataset Caltech-UCSD Birds-200-2011 [183].

always available.

3. **Parts and Poses-based Approaches**

Distinctive features for fine-grained objects sometimes come from object parts. Pictorial structure [52, 47], constellation models [49] and discriminatively trained deformable part models [46] are examples of the many methods that detect discriminative parts. With parts detected, articulated objects can be aligned so that corresponding parts can be compared. Parkhi et al. used a face detector to detect the face of a cat/dog and then use a head + body layout as the final image representation [107, 134]. Asma et al. [155] detected landmark regions of plants (petal, sepal, labellum) and only built descriptors around these vantage parts.

Beyond object parts, a particular part of the object pose under a given viewpoint can be detected

by poselet detectors [24, 202]. The output of such detectors can also be thought of as a mid-level feature, on top of which one can run a layer of classification or regression. Instead of directly detecting individual parts of the object, Gavves et al. showed that roughly aligning the objects as a whole also allows for successful recognition of fine-grained objects [53].

4. **Learning based Approaches**

   Bangpeng et al. [196] proposed a vocabulary and annotation-free method in which image representation is acquired by high-throughput template-matching, with each template being randomly sampled on the images. Berg et al. argued that the conventional ways of constructing mid-level representations out of the standard low-level features are unlikely to be optimal for any particular problem. The best approach should be varied from task to task, i.e. the approach of constructing mid-level representations for recognizing birds should be different from recognizing cars. Therefore they proposed a framework to learn mid-level level features which called Part-based One-vs-One Features (POOFs) from a large richly-annotated dataset [17].

2.3 Deep Neural Network based Methods

Recently, deep learning has been shown to exhibit superior performance on many standard recognition benchmarks, both in speech [58] and visual recognition [145]. The breakthrough is mainly due to the large public image repositories and high-performance computing systems, such as GPUs or large-scale distributed clusters [39] or specialized hardware [93]. Convolutional neural networks (CNN) [99] has been adopted in many visual recognition systems nowadays but the original concept can be traced back to 1980s. It was inspired by the finding that cells in the visual cortex are sensitive to different size of receptive fields which are essentially a two-dimensional subregion in visual space [74]. The major characteristic of this architecture is the local connectivity and shared weights among neighbouring neurons. Features with hierarchical levels of abstraction can be learned directly from the training process with a minimum amount of domain-knowledge. CNNs have been adopted to solve many other vision problems such as non-reference image quality assessment [82], depth map estimation [43, 42, 106], visual saliency detection [103], and edge detection [19].

Figure 2.7 demonstrates a typical architecture of CNN that is composed of two stages [99], with each stage composed of three layers: one convolution layer, one nonlinearity layer and one pooling layer.

Convolution Layer: the input is a 3D array with n_3 2D feature maps of size $n_1 \times n_2$ (e.g. for the very first layer, input is a colour image with 3 channels R, G, B, thus n_1 and n_2 are the image width and height and $n_3 = 3$). Each component in the array is denoted x_{ijk}, and each feature map is denoted $\mathbf{x}_k$, where $k \in [0, n_3]$. The output is also a 3D array $\mathbf{y}$ which is composed of m_3 feature maps of size $m_1 \times m_2$. The mapping of the input feature map $\mathbf{x}_k$ to output feature map $\mathbf{y}_k$ is accomplished by a trainable filter (kernel) a with the relation being expressed as

$$\mathbf{y}_k = \mathbf{a} \otimes \mathbf{x}_k + \mathbf{b}_k$$

where $\otimes$ is the 2D discrete convolution operator and b_k is a trainable bias vector as illustrated in Figure 2.8. Note that, in practice, this convolution can span more than one feature map.

Nonlinearity Layer: A nonlinear activation function is then applied to each component (x_{ijk}), e.g. most commonly the rectified hyperbolic tangent (ReLU) [125]:

$$f(x) = max(0, x),$$

to impose sparsity and reduce the likelihood of a vanishing gradient.

Feature Pooling Layer: The term "pooling" here has exactly the same meaning as in BoW model. Features in the local spatial neighbourhood around each component are pooled to a single value, which results into a series of reduced-resolution feature maps. Doing this not only makes the feature robust to small spatial translations but also makes the computation tractable. The most common used pooling methods are average pooling and max pooling because of their simplicity. Traditionally, pooling is performed on each feature map separately, however, recently, pooling has also been been done across feature maps [84].

Practical models can be much deeper and more complicated than simply stacking these primitive layers together. For example, winner of ILSVRC-2014 employs a 19-layer model [160] and the residual network that won the 2015 ImageNet classification task has a depth of 152 and introduced shortcut connections between layers for residual learning [66]. Furthermore, there are other additional layers that can be inserted in-between for efficient training. To name a few, batch normalization layer [75] is proposed to force the activations to take on a unit gaussian distribution. Dropout layer [162] is proposed to only keep a neuron activate at a certain probability during the training and served as another regularization on the network. The parameters can be trained via simple stochastic gradient descent with sufficient labeled training data (ILSVRC has roughly 1.2 million labeled training images with the help of Amazon's Mechanical Turk crowd-sourcing tool). Recently, more sophisticated learning methods like Adam [86], Adagrad [41], AdaDelta [199], RMSprop [168] have been proposed and shown to have a faster convergence.

Training a deep network with millions of parameters from scratch requires a huge amount of labelled data and computational resources. For labeled datasets that is fairly small (on the order of thousands), which are most commonly seen in the medical imaging domain, fine tuning a pre-trained network tends to work reasonably well [165]. Moreover, recent researches [71, 14, 118, 70, 15] have also shown that unsupervised learning can be used to train each stage one after the other using only unlabelled data for a better initialization of the network parameters. But the small size of seed dataset prohibits effective training of a deep network, no matter it is for a full training, fine tuning, or layer-wise pre-training. Fortunately, it is found that a pre-trained CNN on ImageNet can be used directly as a feature extractor or as a baseline for transfer learning[1] [145, 135, 10, 172]. As such, in Chapter 4, a pre-trained CNN will be employed to serve as a baseline for the performance evaluation of the proposed seed identification method.

[1]Improvement of learning in one task by leveraging related knowledge learned from another task

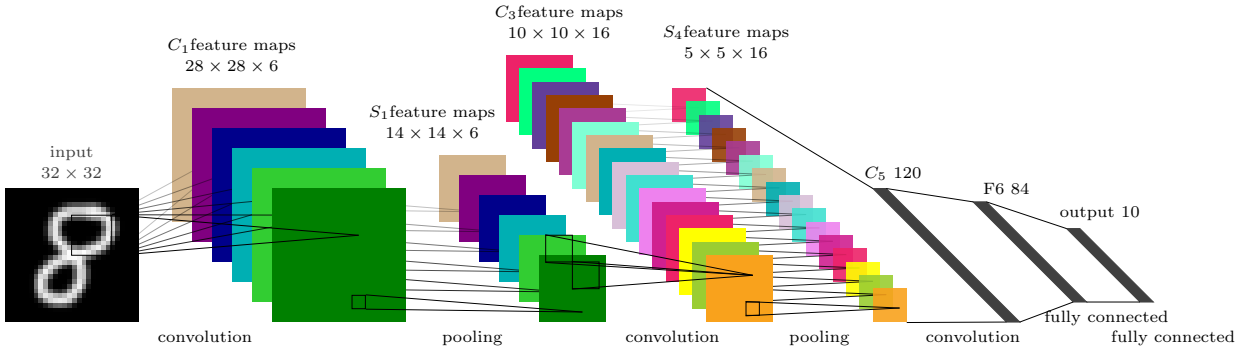


Figure 2.7: LeNet used for digital character recognition, a reproduction of Figure 2 of LeCun et al. [98]. It is a typical CNN architecture with two feature extraction stages. Nonlinear operation is applied right after convolution thus is not shown in this diagram.

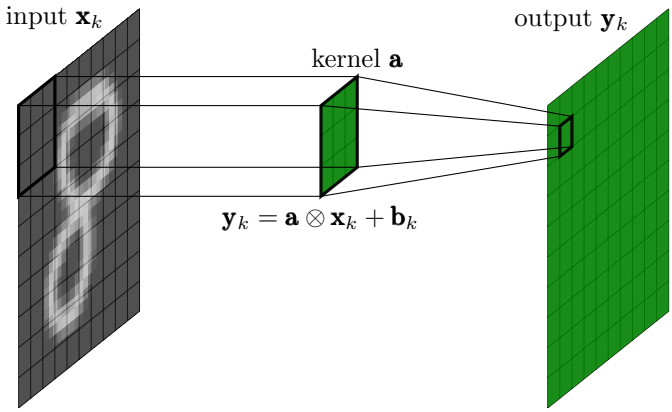


Figure 2.8: Image convolution with kernel. For simplicity, the depth of the kernel is set to 1. In practice, the depth of both the feature map and the kernel is almost always larger than one. Thus the convolution is performed between two 3 dimensional tensors. A nonlinear function is instantly applied element-wisely on the convolved results.

Chapter 3

Defocus blur segmentation

The defocus blur segmentation method has already been published in IEEE Transaction on Image Processing (TIP) with Xin Yi as the lead author.

Copyright Notice

3.1 Introduction

Defocus blur in an image is the result of an out-of-focus optical imaging system. In the image formation process, light radiating from points on the focus plane are mapped to a point on the sensor, but light from a point outside the focus plane illuminates a non-point region on the sensor known as a circle of confusion. Defocus blur occurs when this circle becomes large enough to be perceived by humans.

In digital photography, defocus blur is employed to blur background and "pop out" the main subject using large-aperture lenses. However, this inhibits computational image understanding since blurring of the background suppresses details beneficial to global scene interpretation. In microscopic imaging of opaque 3D specimens, e.g. plant seeds, this effect could have both good and bad influences. One one hand, if the seed sample is in-focus within a single image frame, the defocus blur can be served as a cue to separate the seed from the potentially cluttered background. On the other hand if the seed is under high magnification where the depth-of-field is so narrow that only a small portion can be in focus as already shown in Figure 1.2, multiple image frames acquired at different focal distance would be required for focal stacking to create an all-in-focus image. The reason is that the blurring of large portion of the seeds will make certain species

indistinguishable since usually similar seed species would look the same under large amount of blur. Encoding features from the blurred areas would degrade the image descriptors. Both cases would require efficient and accurate detection of blurred or non-blurred regions.

Moreover, several other contexts could also benefit from accurate blur detection including: 1) in avoiding expensive post-processing of non-blurred regions (e.g. deconvolution); 2) in computational photography to identify a blurred background and further blur it to achieve the artistic bokeh effect [11, 159], particularly for high-depth-of-field cellular phone cameras.

Herein, I treated the defocus blur detection problem as a binary segmentation problem where 1 denotes the sharp region and 0 denotes the blur region. I proposed a novel sharpness metric based on Local Binary Patterns (LBP) that is able to run in real-time and can be adopted not only for defocus segmentation but also for focal stacking.

3.2 Related works

The most common approach to defocus segmentation is local sharpness measurement. There are many works in this area in the past two decades and most of them can be found in the image quality assessment field where images are rated by a single sharpness score that should conform to the human visual perception. These applications only require a single sharpness value to be reported for a single image, thus most of the measures only rely on sharpness around local edges [50, 126, 119] or some distinctive image structures determined in the complex wavelet transform domain [64]. Similarly, the line spread profile has been adopted for edge blurriness measurement in image recapture detection [167]. Since most of these metrics are measured around edges, they cannot readily characterize sharpness of any given local image content unless using interpolation as was done in [11, 210].

Measures such as higher order statistics [85], variance of wavelet coefficients [184], and local variance image field [191] have been used directly in segmentation of objects of interest in low-depth-of-field images. These local sharpness metrics are based on local image energy which means that the measures will not only decrease if the energy of the point spread function (PSF) decreases (becomes more blurry), but also decreases if the energy of the image content drops. Thus, a blurry, high-contrast edge region could have a higher sharpness score than an in-focus, low-contrast one. These metrics are suitable for relative sharpness measures, e.g. in focal stacking, but do not behave very well for local sharpness measure across various image contents. This deficiency has already been pointed out in [208].

Recently, the authors of [159, 108] proposed a set of novel local sharpness features, e.g. gradient histogram span, kurtosis, for training of a naïve Bayes classifier for blur classification of local image regions. The sharpness is interpreted as the likelihood of being classified as sharp patch. Su et al. used singular value decomposition (SVD) of image features to characterize blur and simple thresholding for blurred region detection [163]. Vu et al. used local power spectrum slope and local total variation to measure sharpness in

both the spectral and spatial domains. The final sharpness is the geometric mean of the two measures [181].

Instead of measuring sharpness only based on local information, Shi et al. proposed to learn a sparse dictionary based on a large external set of defocus images and then use it to build a sparse representation of the test image patch. The final measure was the number of non-zero elements of the corresponding words [79].

Depth map estimation is another approach that can also be used for defocus blur segmentation. Zhuo et al. used edge width as a reference for depth measurement under the assumption that edges in blurred regions are wider than those in sharp regions [210]. They obtained a continuous defocus map by propagating the sharpness measures at edges to the rest of the image using image matting [101]. Bae and Durand's work is similar, but they computed edge width differently by finding the distance of second derivative extrema of opposite sign in the gradient direction [11]. These methods tend to highlight edges in places where the blur measure is actually smooth.

Zhu et al. tried to explicitly estimate the space-variant PSF by analyzing the localized frequency spectrum of the gradient field [209]. The defocus blur kernel is parameterized as a function of a single variable (e.g. radius for a disc kernel or variance for Gaussian kernel) and is estimated via MAP_k estimation [102]. Similar work can be found in [29] but the blur kernel is restricted to a finite number of candidates. Khosro et al. estimate the blur kernel locally using blind image deconvolution by assuming the kernel is invariant inside the local block. But instead of fitting the estimated kernel to a parameterized model, they quantified the sharpness through reblurring [12]. Florent et al. treat the blur kernel estimation as a multi-label energy minimization problem by combining learned local blur evidence with global smoothness constraints [35]. These methods are inherently slow because of their iterative nature.

Unlike [11, 209, 210], I do not intend to construct a depth map. My goal is only to separate in-focus regions from regions of defocus blur. Also, unlike [79], I do not rely on external defocus images; in this respect my work is most similar to [163, 108, 159, 181] but with better runtime and segmentation performance. I postulate that local-based defocus blur segmentation methods to date have been limited by the quality of the sharpness measures which they employ.

Local metrics of image sharpness that have been recently introduced for the segmentation of blurred regions are now reviewed in the following section. Generally, they fall into one of three categories: gradient domain metrics, intensity domain metrics, and frequency domain metrics.

3.3 Commonly used Sharpness Metrics

3.3.1 Gradient Domain Metrics

1. **Gradient Histogram Span** [203, 163]

   The gradient magnitude of sharp images exhibits a heavy-tailed distribution [48, 156, 102, 88] and can

be modelled with a two-component Gaussian mixture model (GMM):

$$G = a_1 e^{-\frac{(g-\mu_1)^2}{\sigma_1}} + a_2 e^{-\frac{(g-\mu_2)^2}{\sigma_2}}, \tag{3.1}$$

where means $\mu_1 = \mu_2 = 0$, variance $\sigma_1 > \sigma_2$, g is the gradient magnitude, and G is the gradient magnitude distribution in a local region. The component with larger variance is believed to be responsible for the heavy-tailed property. Thus the local sharpness metric is:

$$m_{GHS} = \sigma_1. \tag{3.2}$$

2. **Kurtosis** [159]

Kurtosis, which captures the "peakedness" of a distribution, also characterizes the gradient magnitude distribution difference. It is defined as:

$$K = \frac{E[(g-\mu)^4]}{E^2[(g-\mu)^2]} - 3, \tag{3.3}$$

where the first term is the fourth moment around the mean divided by the square of the second moment around the mean. The offset of 3 is to cause the peakedness measure of a normal distribution to be 0. The derived local sharpness metric is:

$$m_K = \min(\ln(K(g_x) + 3), \ln(K(g_y) + 3)), \tag{3.4}$$

where g_x, g_y are gradient magnitudes along x and y axis respectively.

3.3.2 Intensity Domain Metrics

1. **Singular Value Decomposition (SVD)** [108]

An image patch $\mathbf{P}$ can be decomposed by SVD:

$$\mathbf{P} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T = \sum_{i=1}^{n} \lambda_i \mathbf{u}_i \mathbf{v}_i^T, \tag{3.5}$$

where $\mathbf{U}, \mathbf{V}$ are orthogonal matrices, $\mathbf{\Lambda}$ is a diagonal matrix whose diagonal entries are singular values arranged in descending order, $\mathbf{u}_i$ and $\mathbf{v}_i$ are the column vectors of $\mathbf{U}$ and $\mathbf{V}$ respectively, and λ_i are the singular values of $\mathbf{\Lambda}$. It is claimed that large singular values correspond to the rough shape of the patch whereas small singular values correspond to details. The sharpness metric is:

$$m_{SVD}(k) = 1 - \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{n} \lambda_i}, \tag{3.6}$$

where the numerator is the sum of the k largest singular values.

2. **Linear Discriminant Analysis (LDA)** [159]

By sampling a set of blurred and non-blurred patches, this method finds a transform $\mathbf{W}$ that maximizes the ratio of the between-class variance S_b to the within-class variance S_w of the projected data with each variance:

$$\begin{aligned}\mathbf{S}_b &= \sum_{j=1}^{2}(\boldsymbol{\mu}_j - \boldsymbol{\mu})^T(\boldsymbol{\mu}_j - \boldsymbol{\mu}),\\ \mathbf{S}_w &= \sum_{j=1}^{2}\sum_{i=1}^{N_j}(\mathbf{x}_j^i - \boldsymbol{\mu}_j)^T(\mathbf{x}_j^i - \boldsymbol{\mu}_j),\end{aligned} \tag{3.7}$$

where $j = 1$ represents the blurred class, $j = 2$ represents the sharp class, $\mathbf{x}_j^i$ is the vectorized pixel intensity of the i-th sample of class j, $\boldsymbol{\mu}_j$ is the mean of image intensity in class j, $\boldsymbol{\mu}$ is the mean across all classes and N_j is the number of samples in the corresponding class (see also Section 2.3 of [159]). This is solved by maximizing the ratio $\frac{det|\mathbf{S}_b|}{det|\mathbf{S}_w|}$ and the resulting column vectors of the projection matrix $\mathbf{W}$ are the eigenvectors of $\mathbf{S}_w^{-1}\mathbf{S}_b$. The final metric can be expressed as:

$$m_{LDA}(i) = \mathbf{w}_i^T\mathbf{P}, \tag{3.8}$$

where $\mathbf{w}_i$ is the i-th column vector of matrix $\mathbf{W}$, and $\mathbf{P}$ is the vectorized patch intensity.

3. **Sparsity** [79]

This measure is based on sparse representation. Each patch is decomposed according to a learned over-complete dictionary which expressed as

$$\underset{\mathbf{u}}{\operatorname{argmin}}\,||\mathbf{P} - \mathbf{D}\mathbf{u}||^2 + \lambda||\mathbf{u}||_1 \tag{3.9}$$

where $\mathbf{D}$ is the learned dictionary on a set of blurred image patches. $\mathbf{P}$ is the vectorized patch intensity and $\mathbf{u}$ is the coeficients vector, each item of which is the weight used for the reconstruction. The reconstruction of a sharp patch requires more words than blurred patches. Thus the sharpness measure is defined as the number of non-zero elements in $\mathbf{u}$, i.e., the L_0 norm of $\mathbf{u}$.

$$m_S = ||\mathbf{u}||_0 \tag{3.10}$$

4. **Total variation** [181]

This metric is defined as

$$\begin{aligned}m_{TV} &= \frac{1}{4}\max_{\xi\in P} TV(\xi)\\ \text{with } TV(\xi) &= \frac{1}{255}\sum_{i,j}|x_i - x_j|\end{aligned} \tag{3.11}$$

which is the maximum of the total variation of smaller blocks ξ (set as 2×2 in the original paper) inside the local patch P. The coefficient $\frac{1}{4}$ is a normalization factor since the largest TV of a 2×2 block is 4. The author argued that a non-probabilistic application of TV can be used as a measure of local sharpness due to its ability to take into account the degree of local contrast.

3.3.3 Frequency Domain Metrics

1. **Power Spectrum** [108, 159, 181]

   The average of the power spectrum for frequency ω of an image patch is:

   $$J(\omega) = \frac{1}{n}\sum_{\theta} J(\omega, \theta) \simeq \frac{A}{\omega^{\alpha}} \tag{3.12}$$

   where $J(\omega, \theta)$ is the squared magnitude of the discrete Fourier transform of the image patch in the polar coordinate system, n is the number of quantizations of θ, and A is an amplitude scaling factor. It was shown that $\alpha = 2$ for sharp, natural images [176, 51, 28]. Since blurred images contain less energy in the high frequency components, the magnitude of their power spectra tend to fall off much faster with increasing ω, and the value of α is larger for such images. Rather than fitting a linear model to obtain α, the average of the power spectrum can be used instead as an indicator since the power spectra of blurred regions tend to have a steeper slope than for sharp regions, thus have a smaller average power. The metric is:

   $$m_{APS} = \frac{1}{n}\sum_{\omega}\sum_{\theta} J(\omega, \theta). \tag{3.13}$$

   In [181, 108], the authors directly use the fitted spectrum slope α as the measure. However, the author in [159] claimed that the average power spectrum is more robust to outliers and overfitting, thus I only evaluate m_{APS}.

3.4 Drawbacks of current sharpness metrics

Given the sharpness metrics reviewed in section 3.3, I conducted a preliminary study to observe how they respond to different local image textures to see if they are limiting progress in blur detection as previously postulated. Since the proposed work is centred on local sharpness measures, this experiment excludes measures that rely on external information, e.g. m_{LDA} and m_S.

Following the same methodology in [34], I assumed there are four common types of textures that appear in natural scenes, a random texture such as grass, a man-made texture, a smooth texture such as sky or fruit surface, and an almost smooth texture such as areas on the road sign (its texture is of low contrast and has more detail than pure smooth regions). Four such exemplar textures are shown in Figure 3.1. Gaussian blur of varying severity ($\sigma \in [0.1, 10.0]$) was applied to these image patches and each metric was computed for each texture and blur level. For the SVD-based metric, I tested with $k = 6$, that is, $m_{SVD}(6)$, but the response is similar for most values of k. The size of image patches were 21×21 pixels for all metrics.

Figure 3.2 shows the response of each metric to each of the four exemplar textures in Figure 3.1 over the tested range of σ. In addition, by evaluating 8000 sharp patches covering different scenes, an aggregate performance of these measures is also shown in Figure 3.2. The thick red curve shows the mean response over the 8000 patches and the dashed red curves show higher and lower quartiles (75th and 25th percentile). It

man-made texture

smooth texture

random texture

almost smooth texture

Figure 3.1: Four commonly appeared textures in natural scenes.

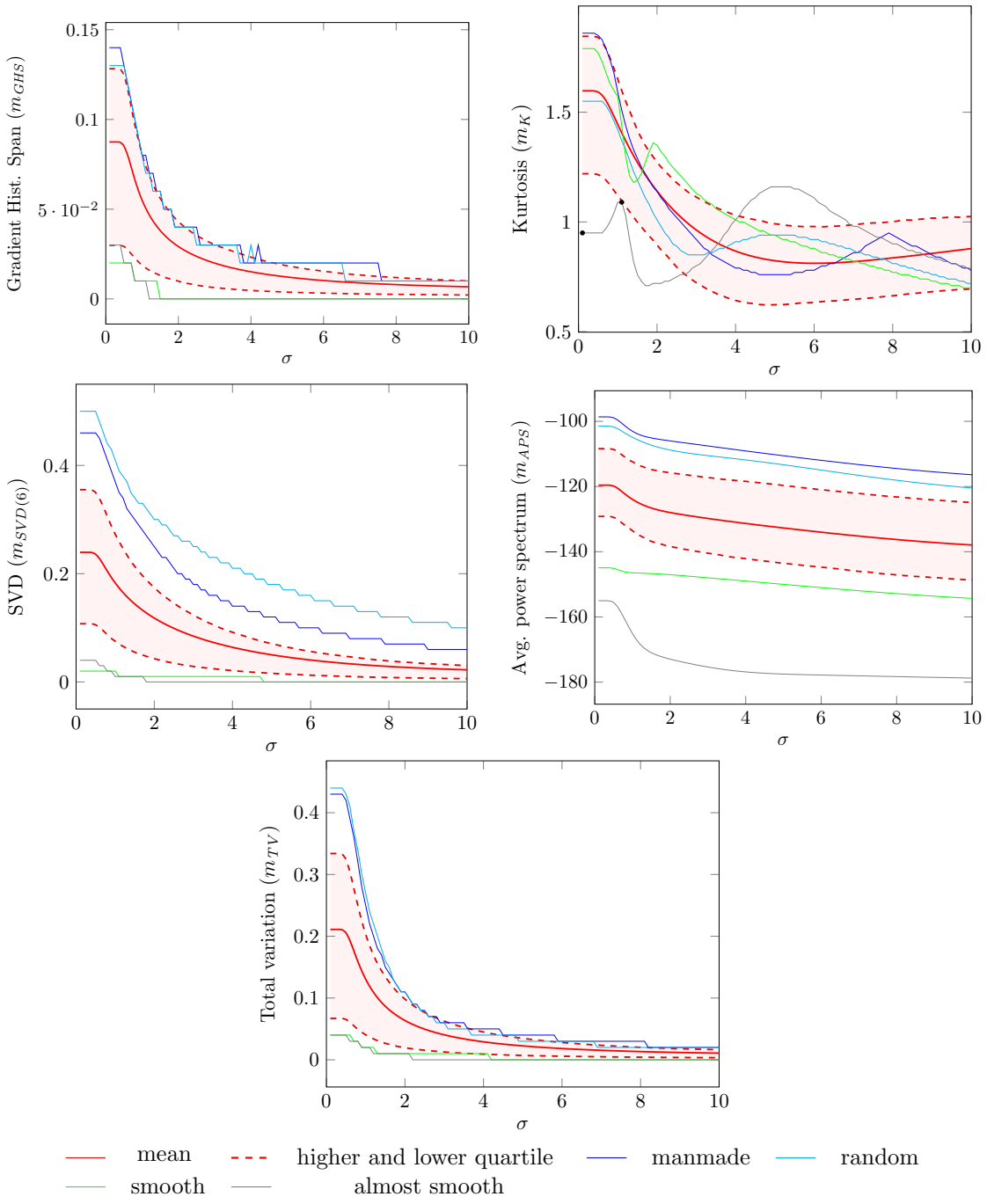


Figure 3.2: Responses of different measures. The thick red curve shows the mean performance over 8000 patches and the dashed red line shows the higher and lower quartile. The responses to 4 exemplar patches are shown in blur, cyan, green, grey curves respectively.

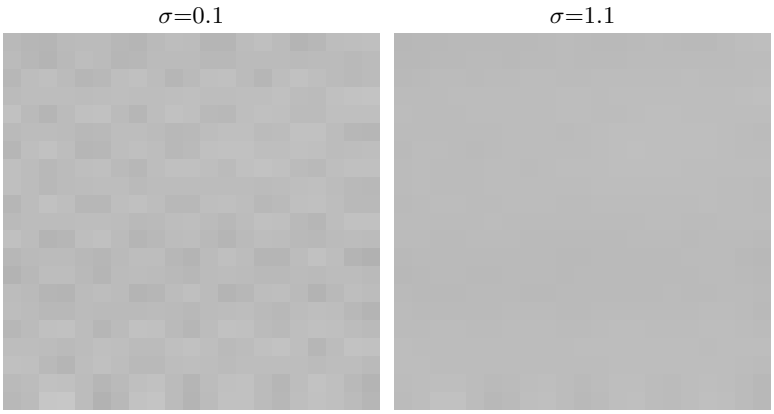


Figure 3.3: An example of the non-monotonicity of the sharpness measure m_K. The patches showing here are the almost smooth patch under two levels of Gaussian blur as marked by black dots in m_K response in Figure 3.2.

can be seen from this figure that, in an aggregate manner, all measures decreases when blur extent increases (one exception is that m_K shows a slight increase after σ approaches 5). However, the aggregate data hides responses that are very different from the aggregate with m_{GHS} and m_K exhibiting minor to moderate non-monotonicity on some specific textures. Two patches are shown in Figure 3.3 with two levels of blur. The one with larger σ has larger m_K.

A smooth texture should elicit a constant, yet weak response to the sharpness metrics since its appearance does not change with varying degrees of defocus blur, but the yellow curve shows big differences in responses for most of the sharpness metrics, with m_{GHS}, m_{TV} and m_{SVD} exhibiting the least variation. One would also expect that blurry regions would have smaller responses than sharp regions, but that is not the case for all metrics. At a given σ, for example 1.5, the region formed by the higher and lower quartiles has a large intersection with the quartiles for $\sigma = 0$. In this respect, m_{APS} has the worst performance. Finally, none of the metrics are well-suited for measuring low contrast sharp regions, such as the almost smooth region in the example. This is because the low contrast region has very small intensity variance which leads to low gradient and low frequency response. The green and grey curve are almost inseparable for m_{GHS}, m_{SVD} and m_{TV}. This drawback is further shown in Figure 3.11. The low contrast yellow region of the road sign does not have a correct response for all measures even if it is in focus.

In the next section I proposed a new sharpness metric based on local binary patterns which has a monotonic response to blur. The range of response values for blur patches has less intersection with sharp regions and it has a more appropriate response to low contrast region.

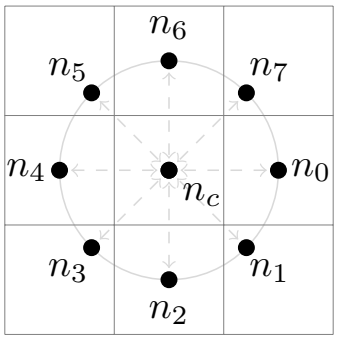


Figure 3.4: 8-bit LBP with $P = 8, R = 1$.

3.5 Proposed LBP based blur metric

Local Binary Patterns (LBP) [131] have been successful for computer vision problems such as texture segmentation [130], face recognition [8], background subtraction [68] and recognition of 3D textured surfaces [141]. The LBP code of a pixel (x_c, y_c) is defined as:

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} S(n_p - n_c) \times 2^p \text{ with } S(x) \quad = \begin{cases} 1 & |x| \geq T_{LBP} \\ 0 & |x| < T_{LBP} \end{cases} \tag{3.14}$$

where n_c is the intensity of the central pixel (x_c, y_c), n_p corresponds to the intensities of the P neighbouring pixels located on a circle of radius R centered at n_c, and $T_{LBP} > 0$ is a small, positive threshold in order to achieve robustness for flat image regions as in [68]. Figure 3.4 shows the locations of the neighbouring pixels n_p for $P = 8$ and $R = 1$. In general, the points n_p do not fall in the center of image pixels, so the intensity of n_p is obtained with bilinear interpolation.

A rotation invariant version of LBP can be achieved by performing the circular bitwise right shift that minimizes the value of the LBP code when it is interpreted as a binary number [132]. In this way, number of unique patterns is reduced to 36. Ojala et al. found that not all rotation invariant patterns sustain rotation equally well [132], and so proposed using only uniform patterns which are a subset of the rotation invariant patterns. A pattern is uniform if the circular sequence of bits contains no more than two transitions from one to zero, or zero to one. The non-uniform patterns are then all treated as one single pattern.

This further reduces the number of unique patterns to 10 (for 8-bit LBP), that is, 9 uniform patterns, and the category of non-uniform patterns. The uniform patterns are shown in Figure 3.5. In this figure, neighbouring pixels are coloured blue if their intensity difference from centre pixel is larger than T_{LBP}, and I say that it has been "triggered", otherwise, the neighbours are coloured red.

Figure 3.6 shows the normalized histogram of the nine uniform LBP patterns appearing in the blurred and non-blurred regions of 100 images randomly selected from a publicly available dataset of 704 partially blurred images [28], each of which is provided with a hand-segmented groundtruth image denoting the blurred and non-blurred regions. Bin 9 is the number of non-uniform patterns. The frequency of patterns 6, 7, 8, and 9

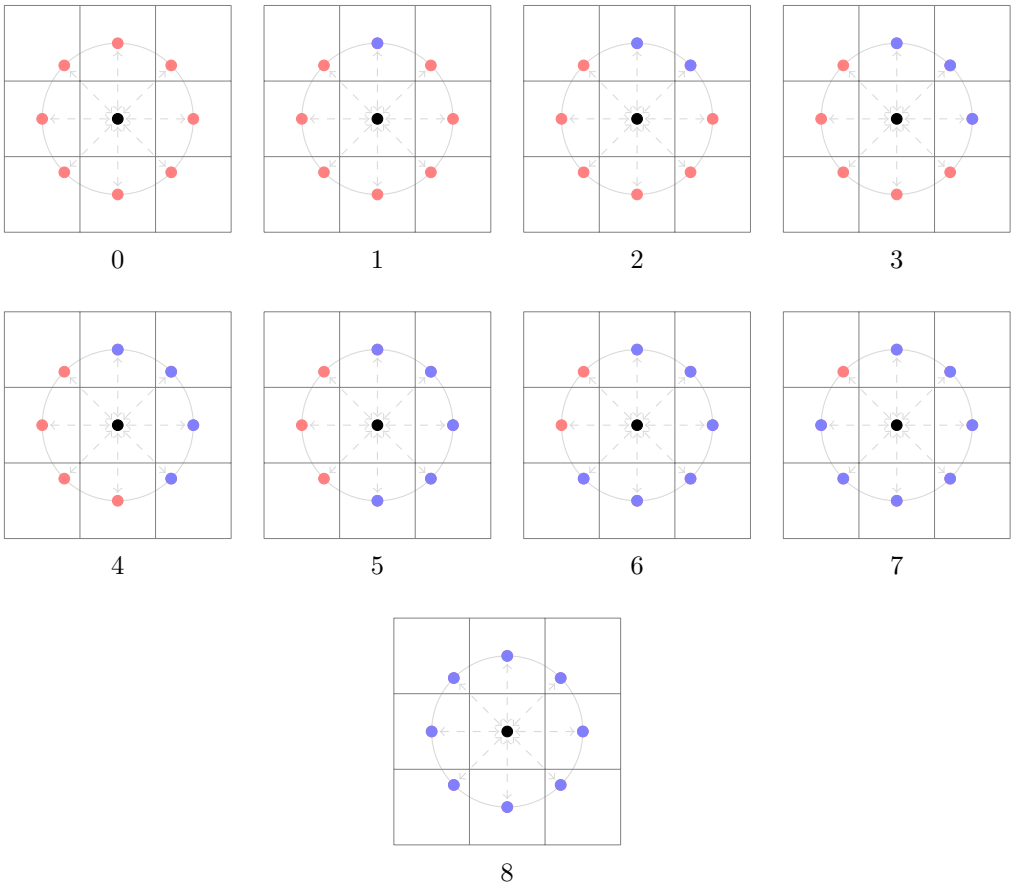


Figure 3.5: The uniform rotationally invariant LBP. Red dots represent pixels that have a intensity difference to the centre pixel less than designated threshold whereas blue dots are the opposite or here interpreted as activated.

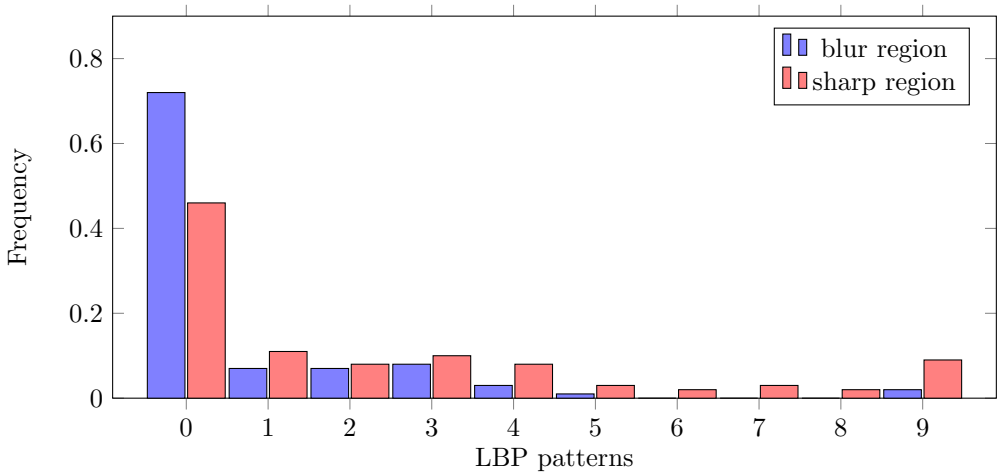


Figure 3.6: LBP code distribution in blurred and sharp regions. Bins 0–8 are the counts of the uniform patterns; bin 9 is the count of non-uniform patterns. Data is sampled from 100 partial blurred images from [158].

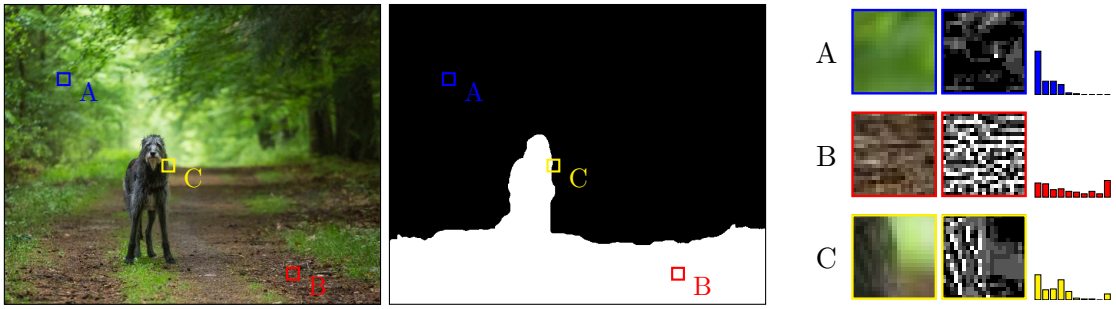


Figure 3.7: Histogram of LBP patterns in three different patches which are sampled from blurred (A), sharp (B), and transitive (C) areas respectively. In the ground truth image, white denotes the sharp region and black the blurred region.

in blurred regions is noticeably less than that for sharp regions. The intuitive explanation for this is that in smoother areas, most neighbouring pixels will be similar in intensity to n_c, and the chance of a neighbour being triggered is lower, making the lower-numbered uniform patterns with fewer triggered neighbours more likely. Examples of the LBP histograms of specific sharp and blurred patches is given in Figure 3.7 which also exhibit this expected behaviour.

My proposed sharpness metric:

$$m_{LBP} = \frac{1}{N} \sum_{i=6}^{9} n(LBP_{8,1}^{riu2} i) \tag{3.15}$$

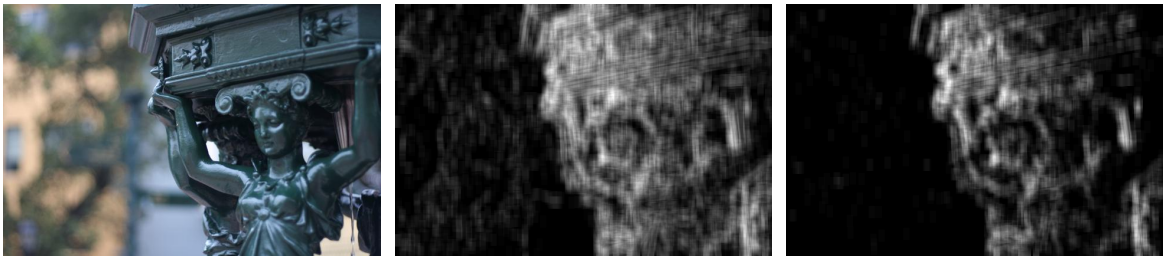
exploits these observations where $n(LBP_{8,1}^{riu2} i)$ is the number of rotation invariant uniform 8-bit LBP pattern of type i, and N is the total number of pixels in the selected local region which serves to normalize the metric so that $m_{LBP} \in [0, 1]$. One of the advantages of measuring sharpness in the LBP domain is that LBP features are robust to monotonic illumination changes which occur frequently in natural images.

The threshold T_{LBP} in Equation 3.14 controls the proposed metric's sensitivity to sharpness. As shown in Figure 3.8, by increasing T_{LBP}, the metric becomes less sensitive to sharpness. However, there is a tradeoff between sharpness sensitivity and noise robustness, as shown in Figure 3.9. In situations where high sensitivity to sharpness is needed, a discontinuity-preserving noise reduction filter such as non-local means [27] should be employed.

Figure 3.10 shows my metric's response to various levels of blur ($T_{LBP} = 0.016$). There is a sharp fall-off between $\sigma = 0.2$ and $\sigma = 1.0$ which facilitates segmentation of blurred and sharp regions by simple thresholding.

Moreover, the metric response is nearly monotonic, decreasing with increasing blur, which should allow such regions to be distinguished with greater accuracy and consistency. Figure 3.11 shows maps of the local response of my metric and comparators for a sample image. My metric has the most coherent response and responds the most consistently to the road sign with respect to the given ground truth.

Table 3.1 shows a comparison of the runtime of m_{LBP} and comparator metrics. Where available, author-



original image | $T_{LBP} = 0.004$ | $T_{LBP} = 0.012$

$T_{LBP} = 0.020$ | $T_{LBP} = 0.028$ | $T_{LBP} = 0.036$

Figure 3.8: Response of m_{LBP} (Equation 3.15) for various values of threshold T_{LBP}. T_{LBP} determines the cutoff for the magnitude of intensity change that is considered an "edge", regardless of edge sharpness.

original image | PSNR = 29.98dB | PSNR = 20.22dB

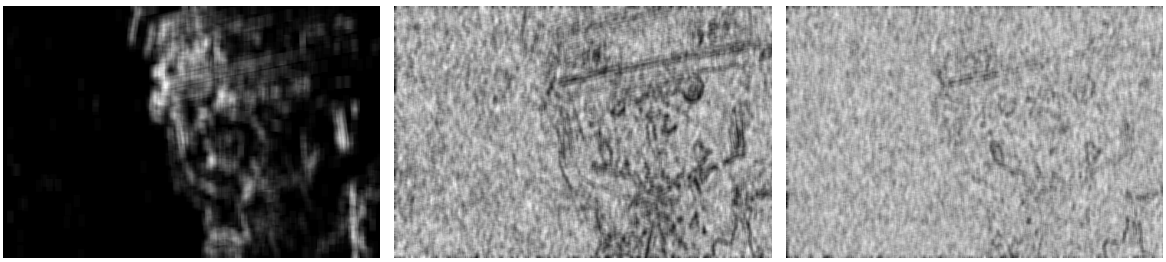


Figure 3.9: Response of m_{LBP} in the presence of noise. Top: the original image and two copies corrupted by Gaussian noise; bottom: the corresponding sharpness maps. $T_{LBP} = 0.016$.

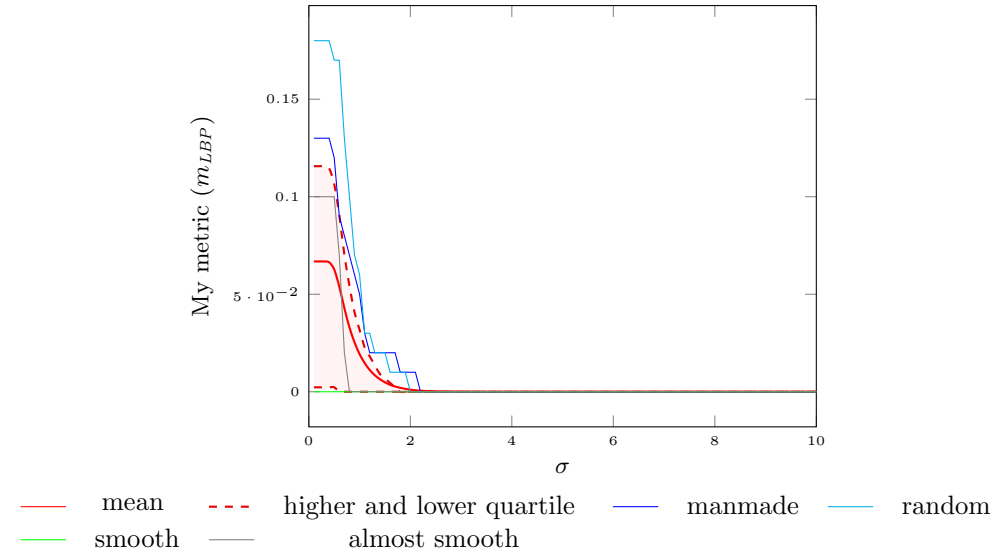


Figure 3.10: My metrics' response to the sample patches shown in Figure 3.1. As the same as in Figure 3.2, an aggregate response on 8000 sharp patches is also shown with the thick red curve showing the mean response and the dashed red curve showing the higher and lower quartile.

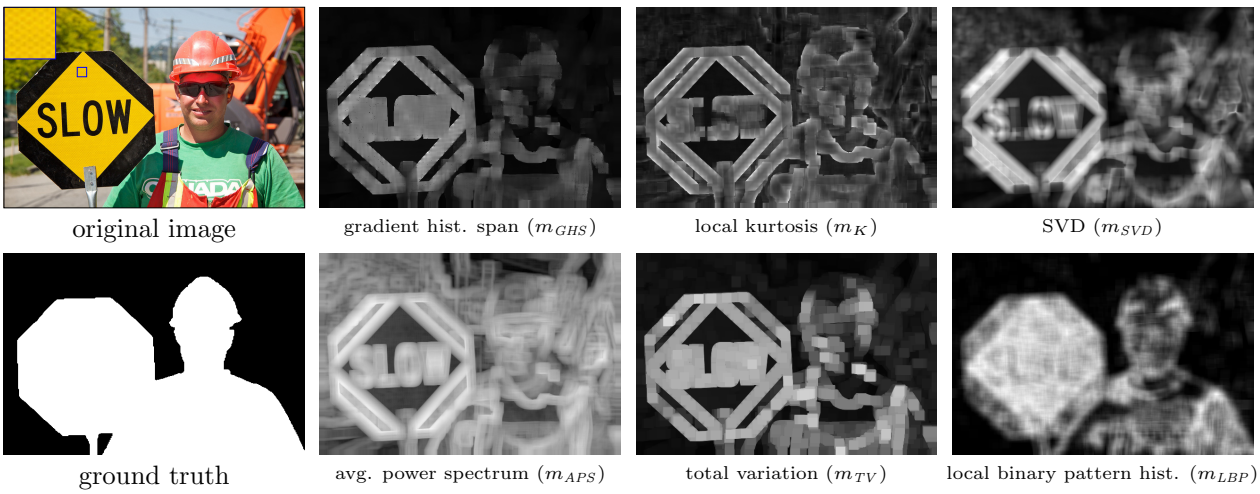


Figure 3.11: Metric responses for a sample image for different sharpness metrics. Only the proposed metric has the correct responses on the local contrast yellow road sign.

Sharpness Metric	Avg. Runtime
gradient histogram span (m_{GHS}) [159, 108]	273.19s
kurtosis (m_K) [159]	11.57s
singular value decomposition (m_{SVD}) [163]	*38.66s
total variation (m_{TV}) [181]	50.00s
average power spectrum slope (m_{APS}) [159]	22.89s
my LBP-based metric (m_{LBP})	*3.55s
my LBP-based metric (m_{LBP}, mex imp.)	*26.5ms

Table 3.1: Runtime comparison of various metrics. Note that the speed of my metric can be boosted by using integral image which makes the complexity independent of the size of local region. Those marked by * are from my own implementation. Mex implementation is a C++ implementation that is callable from MATLAB.

supplied code for calculating the metrics was used, otherwise my own implementations were used (marked with *). All implementations were in MATLAB. 10 randomly selected images with approximate size of 640×480 pixels were tested on a Mac with 2.66 GHz intel core i5 and 8 GB memory. The average runtimes are reported in Table 3.1.

The sharpness maps, response curves, and runtimes provide strong qualitative and quantitative evidence that the proposed metric is superior. In the next section I present a blur segmentation method that achieves the state-of-the-art results by employing this metric.

3.6 New Blur Segmentation Algorithm

This section presents my algorithm for segmenting blurred/sharp regions with the proposed LBP-based sharpness metric; it is summarized in Figure 3.12. The algorithm has four main steps: multi-scale sharpness map generation, alpha matting initialization, alpha map computation, and multi-scale sharpness inference.

3.6.1 Multi-scale Sharpness Map Generation

In the first step, multi-scale sharpness maps are generated using m_{LBP}. The sharpness metric is computed for a local patch about each image pixel. Sharpness maps are constructed at three scales where scale refers to local patch size. By using an integral image [180], sharpness maps may be computed in constant time per pixel for a fixed P and R[1].

[1]P is the number of neighbouring pixels used to compute the LBP code around each pixel. R is the Manhattan distance between the neighbouring pixels to the centre pixel.

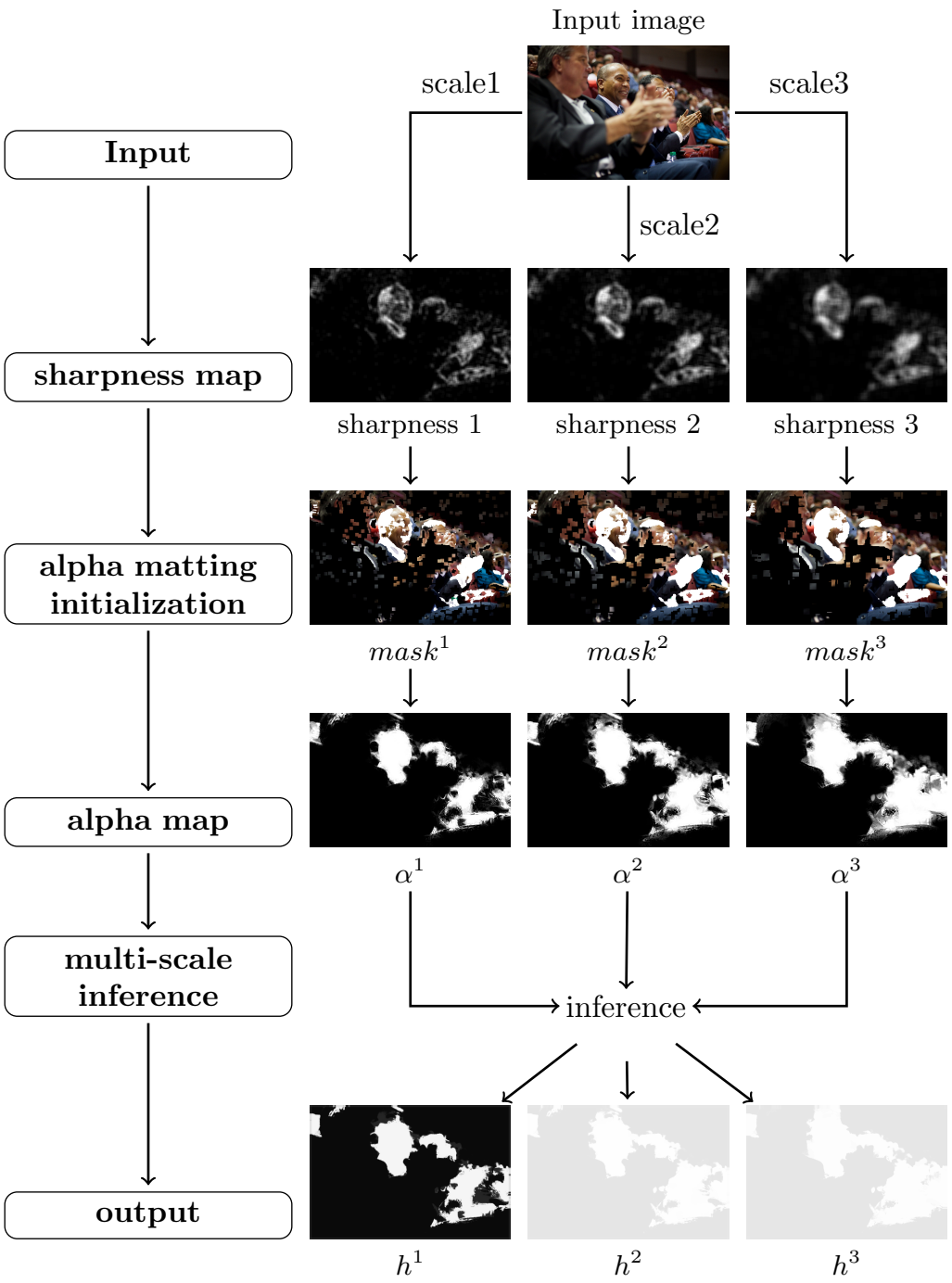


Figure 3.12: My blur segmentation algorithm. The main steps are shown on the left; the right shows each image generated and its role in the algorithm. The output of the algorithm is h^1.

3.6.2 Alpha Matting Initialization

Alpha matting is the process of decomposing an image into foreground and background. The image formation model can be expressed as:

$$I(x,y) = \alpha_{x,y}F(x,y) + (1-\alpha_{x,y})B(x,y), \tag{3.16}$$

where the *alpha matte*, $\alpha_{x,y}$, is the opacity value on pixel position (x,y) and takes a value between 0 and 1 (unlike segmentation which only takes discrete value 0 or 1). It can be interpreted as the confidence that a pixel is in the foreground. Typically, alpha matting requires a user to interactively mark known foreground and background pixels, initializing those pixels with $\alpha = 1$ and $\alpha = 0$, repectively.

Interpreting "foreground" as "sharp" and background as "blurred", I initialized the alpha matting process automatically by applying a double threshold to the sharpness maps computed using the proposed sharpness metric to produce an initial value of α for each pixel:

$$mask^s(x,y) = \begin{cases} 1, & \text{if } m_{LBP}(x,y) > T_{m_1}. \\ 0, & \text{if } m_{LBP}(x,y) < T_{m_2}. \\ I(x,y), & \text{otherwise.} \end{cases} \tag{3.17}$$

where s indexes the scale, that is, $mask^s(x,y)$ is the initial α-map at the s-th scale.

3.6.3 Alpha Map Computation

The α-map will be solved by minimizing the following cost function as proposed by Levin [101]:

$$E(\alpha) = \alpha^T \mathbf{L} \alpha + \lambda(\alpha - \hat{\alpha})^T(\alpha - \hat{\alpha}), \tag{3.18}$$

where α is the vectorized α-map, $\hat{\alpha} = mask^i(x,y)$ is one of the vectorized initialization alpha maps from the previous step, and $\mathbf{L}$ is the matting Laplacian matrix. The first term is the regularization term that ensures smoothness, and the second term is the data fitting term that encourages similarity to $\hat{\alpha}$. For more details on Equation 3.18, readers are referred to [101].

The alpha matting will be applied at each scale as shown in Figure 3.12. The final alpha map at each scale is denoted as $\alpha^s, s = 1,2,3$. The underlying assumption of alpha matting is that patches of similar colour have similar alpha value. By doing this, some smooth sharp regions that do not respond to the sharpness metric can be recovered to some extent.

3.6.4 Multi-scale Inference

The values of the neighbouring pixels are correlated in natural images thus here I chose to regularize the obtained sharpness map (alpha map in the last step) with a conditional random field (CRF) [94] as similarly performed in [159].

To be more specific, a CRF was employed. I maximize the following probability to regularize $\hat{h^s}$ returned by the alpha map:

$$p(h^s; \hat{h^s}) \propto \prod_{i,j \in N_i, s=1}^{3} \psi(\hat{h_i^s}|h_i^s)\Psi(h_i^s, h_j^s) \prod_{s=1}^{2} \Psi(h_i^s, h_i^{s+1}), \tag{3.19}$$

where $\hat{h_i^s} = \alpha_i^s$ is the alpha map for scale s at pixel location i that is computed in the previous step, and h_i^s is the sharpness to be inferred. N_i denotes the neighborhood of i, ψ is the observation model and Ψ is the neighborhood potential, each of which is defined as:

$$\begin{aligned} \psi(\hat{h_i^s}|h_i^s) &\propto \exp\left(-\frac{|\hat{h_i^s} - h_i^s|}{2\sigma_1}\right) \\ \Psi(h_i^s, h_j^s) &\propto \exp\left(-\frac{|h_i^s - h_j^s|}{2\sigma_2}\right). \end{aligned} \tag{3.20}$$

Note that σ_1 and σ_2 were set to be equal in the following. The neighborhood in this setting not only refers to nearby pixels in the same scale (spacial domain) but also across scales.

By computing the negative log likelihood of equation 3.19, maximizing the probability is equivalent to minimizing the total energy which can be expressed as:

$$E(h) = \sum_{s=1}^{3} \sum_{i} |h_i^s - \hat{h_i^s}| + \quad \beta \left(\sum_{s=1}^{3} \sum_{i} \sum_{j \in N_i^s} |h_i^s - h_j^s| + \sum_{s=1}^{2} \sum_{i} |h_i^s - h_i^{s+1}| \right). \tag{3.21}$$

In this form, the first term on the right hand side is the unary term which is the cost of assigning sharpness value h_i^s to pixel i in scale s. The second is the pairwise term which enforces smoothness in the same scale and across different scales. The weight β regulates the relative importance of these two terms. Optimization of Equation 3.21 was performed using loopy belief propagation [123].

The output of the algorithm is h^1 which is the inferred sharpness map at the smallest scale. This is a grayscale image, where higher intensity indicates greater sharpness.

3.7 Dataset

This blur segmentation algorithm was tested using a public blurred image dataset [158] consisting of 704 partially blurred images and their accompanying hand-segmented ground truth images[2]. In addition, since the algorithm is proposed to segment microscopy images such as the seed images, 11 microscopy images were collected for qualitative evaluation.

3.8 Blur Segmentation Algorithm Evaluation

Each image in the dataset was segmented into sharp and blurred regions using the process described in Section 3.6. Sharpness metric m_{LBP} was computed with $T_{LBP} = 0.016$. The sharpness map scales were

[2]The blurred images and ground truth are both from the Image & Visual Computing Lab, Chinese University of Hong Kong.

square local regions of 11×11, 15×15, and 21×21 pixels. The thresholds used in the alpha matting step were $T_{m_1} = 0.3$ and $T_{m_2} = 0.01$. Weight $\beta = 0.5$ was used in the multi-scale inferencing step.

I compared my algorithm to six comparator methods briefly mentioned in Section 3.2 of which I now remind the reader. Su et al. simply calculated a sharpness map using m_{SVD} [163]; Vu et al. combined both spectral and spatial sharpness (S_1 and S_2 in their original paper) using a geometric mean [181]. Shi et al.(14) used all of $m_{GHS}, m_K, m_{LDA}, m_{APS}$ together with a naïve Bayes classifier and multi-scale inference model [159]. Shi et al.(15) formed a sparse representation of image patches using a learned dictionary for the detection of slight perceivable blur [79]. Zhuo and Sim computed a depth map based on edge width [210]. Zhu et al. estimated the space-variant PSF by statistical modelling of the localized frequency spectrum of the gradient field [209].

All the outputs of these methods are grayscale images where greater intensity indicates greater sharpness, and all (except for Zhu et al.) use a simple threshold, T_{seg}, as a final step to produce a segmentation, as in my own algorithm. The parameters for the comparator algorithms were set to the defaults as in their original code. Since I was unable to get the original code for Zhu et al.'s algorithm [209], which belongs to Adobe Systems Inc., the results shown here were produced by my own implementation of the algorithm as described in the published paper. The depth map was normalized by a factor of 1/8 (since the coherence labels are in the range of [0, 8]) and inverted to get the sharpness map.

3.8.1 Precision and Recall

Precision and recall curves were generated for each algorithm by varying the threshold used to produce a segmentation of the final sharpness maps (i.e. similar to [159]).

$$precision = \frac{|R \cap R_g|}{|R|}, \;\; recall = \frac{|R \cap R_g|}{|R_g|} \tag{3.22}$$

where R is the set of pixels in the segmented blurred region and R_g is the set of pixels in the ground truth blurred region. Figure 3.13 shows the precision and recall curves for each method with the threshold T_{seg} sampled at every integer within the interval $[0, 255]$. My algorithm achieves higher precision than the comparator algorithms when recall is above 0.8. Moreover, the proposed sharpness metric alone achieves results comparable to Shi et al.(15).

Figure 3.14 shows the sharpness maps (prior to final thresholding) for each algorithm for a few sample images. My method is superior than the others under various background and blurs. I attribute errors mainly to the shortcomings of the sharpness metrics used by local based methods–Shi et al.(14), Vu et al., Su et al. (Section 3.2). Moreover, my detection maps contain mostly high- or low-confidence values which can be more correctly thresholded.

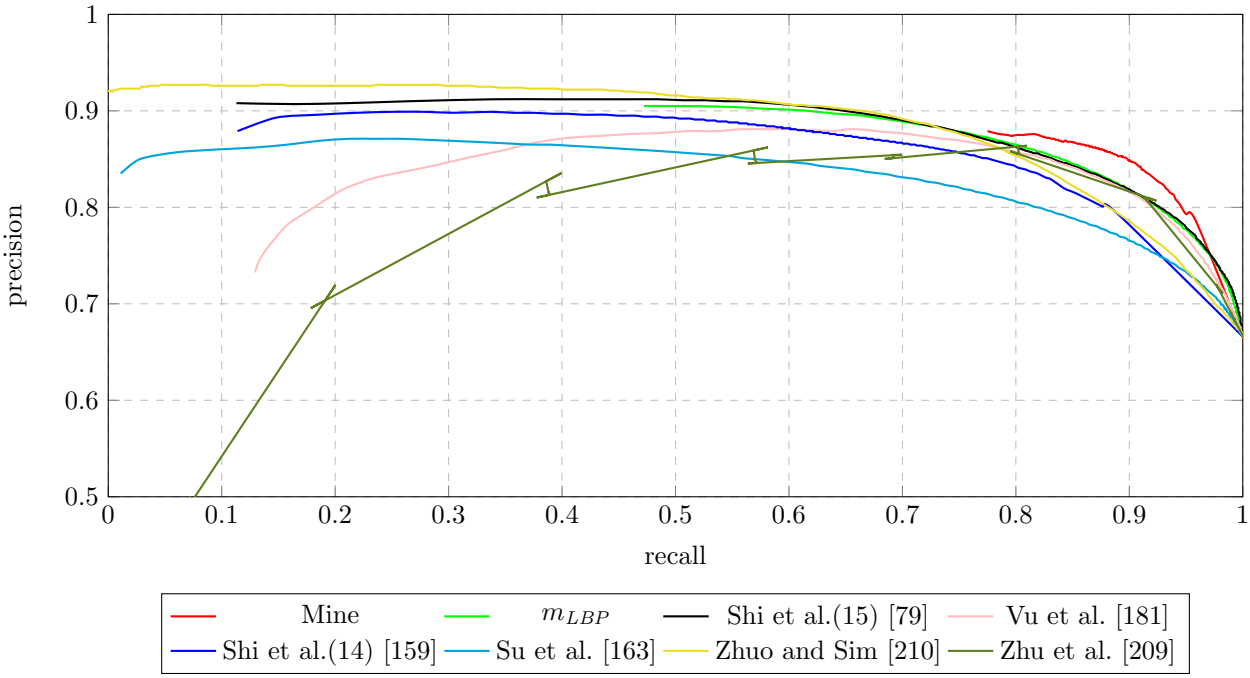


Figure 3.13: Precision and recall curves for different methods on the blur dataset. The curves were obtained by thresholding the sharpness maps with threshold varying in the range of [0, 255]. Note that our method achieves the highest precision when recall is larger than 0.8. This comparison might be unfair for Zhu et al. since their segmentation is based on graph cut rather than thresholding of the depth map. Therefore we compared their graph cut segmented binary map in section 3.8.2.

3.8.2 F-measure

In another experiment, I used an image-dependent adaptive threshold, proposed in [6], for the segmentation with the threshold defined as:

$$T_{seg} = \frac{2}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} I(x, y) \tag{3.23}$$

where, W, H are the width and height of the final sharpness map I. Then, similar to [137], the weighted harmonic mean measure of precision and recall or F-measure was computed for comparison. The definition is as follows:

$$F_\beta = \frac{(1 + \beta^2) \times precision \times recall}{\beta^2 \times precision + recall} \tag{3.24}$$

Here, β^2 was set to 0.3 as in [137, 6].

Note that, the segmentation map of Zhu et al. was produced by graph cut instead of simple thresholding of the depth map. The parameters I used were the same as suggested in their paper which are $\lambda_0 = 1000$, $\sigma_\lambda = 0.04$, $\tau = 2$. Exemplar segmentation maps of images in Figure 3.14 is shown in Figure 3.16. Because my sharpness map contains mostly high confidence values, the F-measure computed for mine was calculated with $T_{seg} = 0.3$. F-measure of all methods can be found in Figure 3.15.

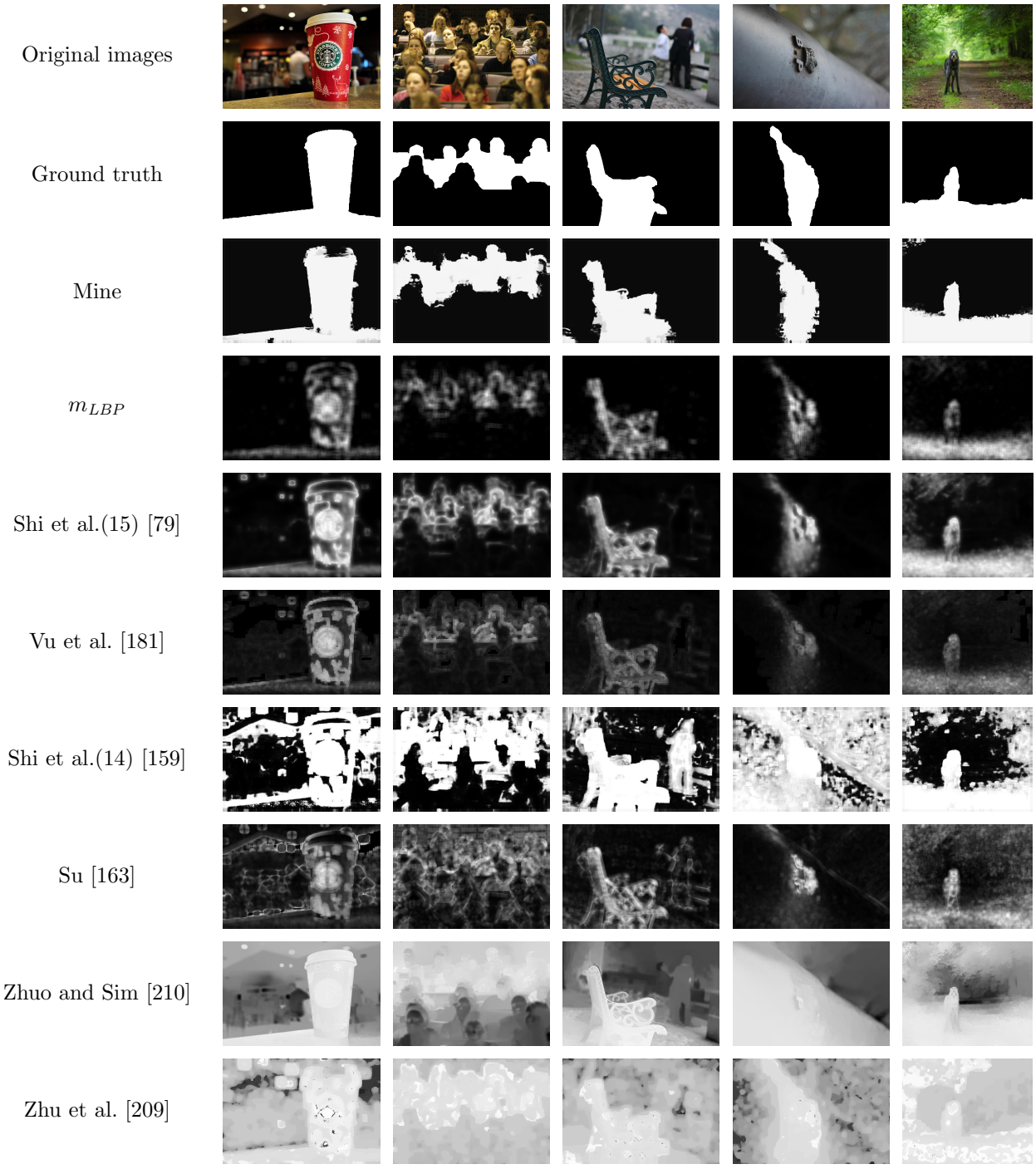


Figure 3.14: Results achieved by different blur detection methods. Final sharpness maps, prior to thresholding for segmentation, are shown.

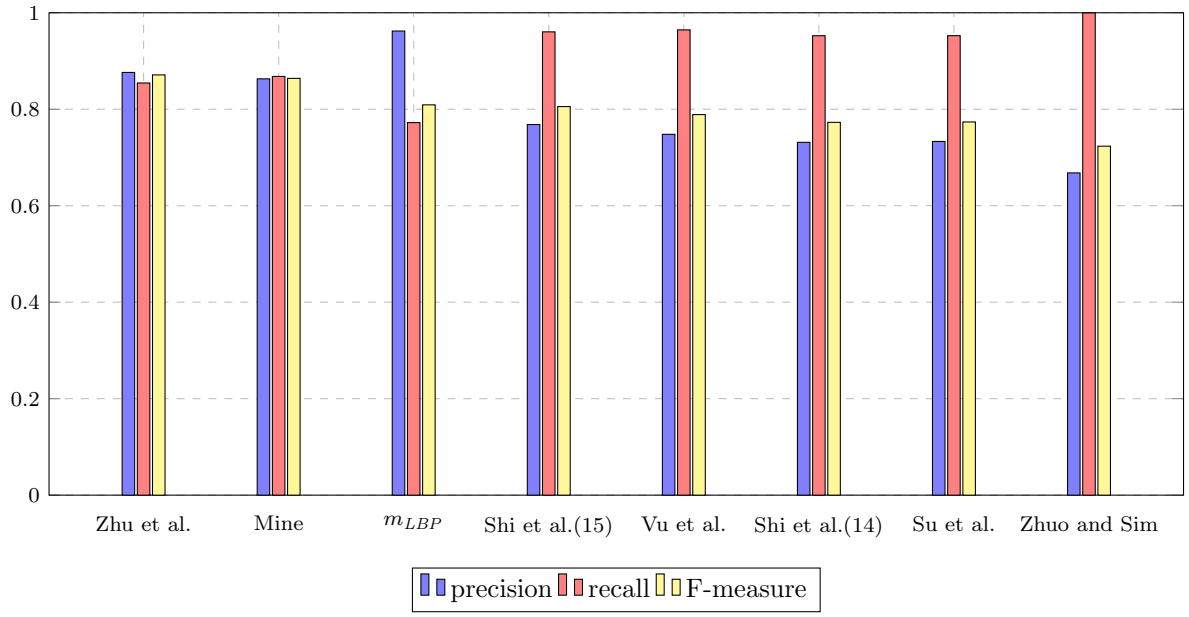


Figure 3.15: Precision, Recall and F-measure for adaptive thresholds. The result of Zhu et al. is achieved by using graph cut instead of simple thresholding as suggested in their paper. Note that because my sharpness map contains mostly high confidence values, the F-measure computed for mine was calculated with $T_{seg} = 0.3$.

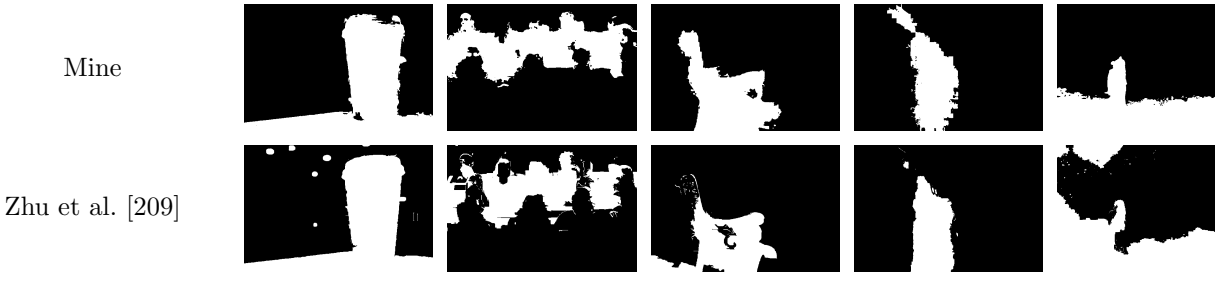
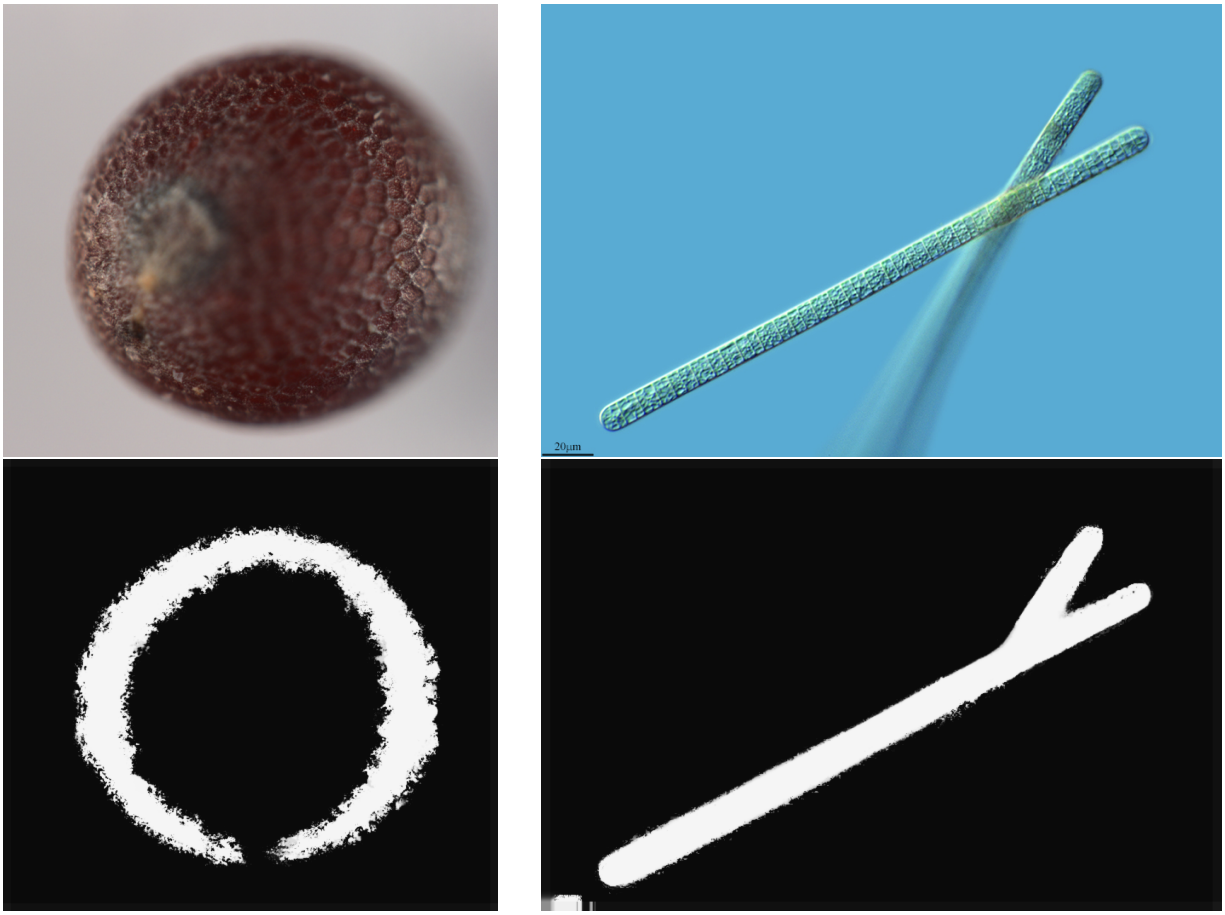


Figure 3.16: Binary segmentation map comparison with Zhu et al.

Blur segmentation	Avg. Runtime
Shi et al.(14) [159]	705.27s
Zhu et al. [209]	387.17s
Shi et al.(15) [79]	38.36s
Su et al. [163]	37s
Mine	27.75s
Zhuo and Sim [210]	20.59s
Vu et al. [181]	19.18s
m_{LBP}	40ms

Table 3.2: Run time comparison of different blur segmentation methods. The time for our method is based on a mex implementation of m_{LBP}.



(a) Plant seed **(b)** Microorganism

Figure 3.17: My algorithm applied to microscopy images. Top row: original images; bottom row: final sharpness maps.

3.8.3 Runtime

A run time comparison of the complete segmentation algorithms is shown in Table 3.2. The same setup was used for the measurement of runtime as in Table 3.1. Compared with the other comparators that has similar precision and recall performance, m_{LBP} has a significant advantage. The time for the complete segmentation algorithm proposed is mostly spent on the matting and multi-scale inference. Although it ranks the fourth among all these methods, its speed is one order of magnitude faster than that of Zhu et al., which is the only algorithm that can match its performance. Since the algorithm of Zhu et al. is implemented by myself, herein I also analyzed its time complexity as opposed to mine for a fair comparison. The worst case complexity of Zhu et al.[3] is $\mathcal{O}((r^2+a)N + MN^2|C|)$ and the time complexity of mine[4] is $\mathcal{O}(N)$. Furthermore, it also surpasses Shi et al.(15) which is my next strongest competitor.

Finally, I give some examples of my algorithm applied to images other than those in our evaluation data set. Microscopy optics often have low depth of field and form an important class of images for blur detection as was shown in Chapter 1. Figure 3.17 shows examples of my algorithm applied to such images. The first is a plant seed [148] whose roughly spherical shape results in a ring-shaped in-focus region. The other image is a microorganism [143] in fresh water. The threshold T_{LBP} for the sharpness metric was set to 0.012 and 0.04 respectively. Note how well my segmentation results conformed to the visual perception of the image sharpness. Additional results can be seen in the appendix.

3.9 Discussion

When there is a distinctive discontinuity between the foreground and background, there is a jagged boundary of my segmentation map, e.g. the cup in Figure 3.14. This is because the sharpness is measured locally. It is inevitable to incorporate regions with various extents of sharpness by using a local window, especially around edges where the depth discontinuity occurs. Therefore, the sharp area is enlarged in the alpha matting initialization step (step B). Zhu et al. solved this problem by taking smoothness and color edge information into consideration in the coherence labeling step but would also fail in cases where depth changes gradually.

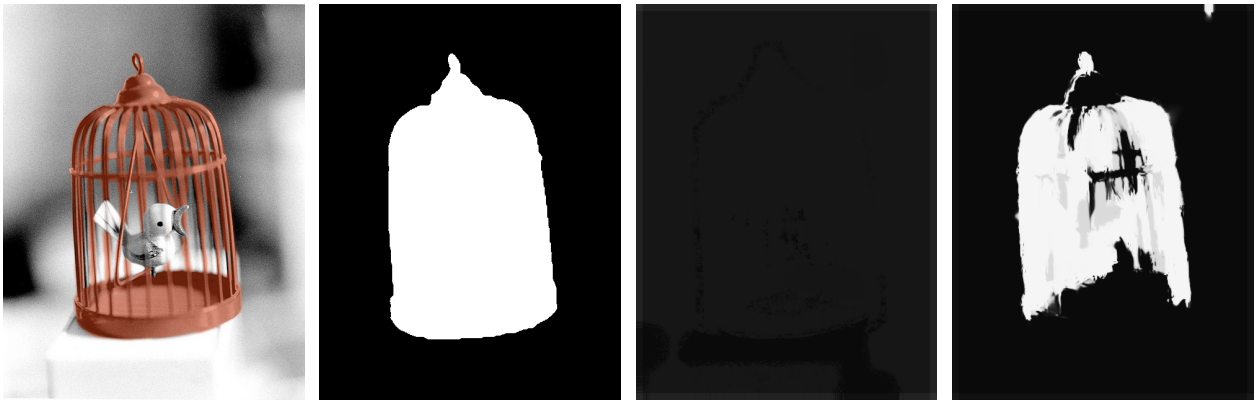
There are certain situations that can cause my method to fail. My method has difficulty differentiating an in-focus smooth region and a blurred smooth region since only a limited small size of local neighbour is considered, but this is a problem that will be inherently challenging for any algorithm. If the noise level in the image is low, this problem can be overcome to some extent by reducing the T_{LBP} threshold. In addition, for object recognition purposes, this drawback would not weaken the feature representation too much since smooth regions contain little to no useful discriminating texture. An example of this type of failure case and

[3]The local frequency analysis of Zhu et al. has a complexity of $\mathcal{O}(r^2N)$; the local probability estimation has a complexity of $\mathcal{O}(aN)$; the graphcut used in coherent labeling has a worst case complexity of $\mathcal{O}(MN^2|C|)$. a is the number of iteration and C is the cost of the minimum cut. N is the number of nodes and M is the number of edges in the formed graph.

[4]$\mathcal{O}(N)$ for sharpness metric, $\mathcal{O}(N)$ for the close form matting (solved by using the large kernel matting Laplacian matrices [65]). $\mathcal{O}(N)$ for the multi-scale inference.

Original image | Groundtruth | Our result | Lower T_{LBP}

(a) failure due to ambiguity of smooth regions



Original image | Groundtruth | Our result | Our result after noise removal with [27]

(b) failure due to noise

Figure 3.18: Blur segmentation algorithm failure cases and mitigation.

the proposed remedy can be seen in Figure 3.18(a).

Another failure case occurs due to image noise, but it can be mitigated by applying a noise reducing filter as mentioned in section 3.5. An example of this type of failure and the proposed remedy is shown in Figure 3.18(b).

The selection of T_{LBP} is essential for obtaining a satisfactory segmentation. It controls how much sharp area would appear in the final segmentation result. For a image with little to no noise, T_{LBP} 0.016 should produce a reasonable result. Lowering the value would cause the inclusion of more low contrast sharp regions. For a image corrupted by noise, a noise reduction procedure should be employed.

3.10 Application: Focal Stacking

Earlier I have proposed a novel no-reference sharpness metric which exploits the distribution difference of uniform LBP patterns in blurred and non-blurred image regions. It runs in realtime on a single core cpu and has a better response on low contrast sharp regions. A single-image-based defocus segmentation algorithm was developed on top of it and achieved state-of-the-art performance. This is beneficial when the seed under microscope is all-in-focus so that we can use the defocus blur as a cue to separate the seed and the potential cluttered background. However, if the seed is unable to be fully observed under the current image setting, then multiple image frames, each of which focuses at different focal distance is required so that sharp regions

of each frame can be merged together. This process is called focal stacking and is a common technique in macro-photography when the surface profile of the observed object is beyond the focal range. The sharpness measures commonly employed in this case are inherently different from those reviewed in Section 3.3 because the sharpness measured is with respect to the same underlying image structure. Measures as simple as variance can perform pretty well in a noise-free condition.

In this section, I applied my proposed metric to the focal stacking problem and conducted a series of experiments to prove its effectiveness. For the sake of simplicity, all the image frames used here are assumed to be perfectly aligned or in other words, there is no image shift due to parallax or magnification change.

Most non-parametric focal stacking methods (that do not model the defocus kernel) more or less follow the same scheme [7]:

1. **Stack Acquisition**

   Acquiring an image stack, $I_k(x, y)$ where k denoting the index of frame at a certain focal distance and x, y denotes the spatial coordinates.

2. **Building a Decision Map**

   A sharpness map is constructed for each frame. the one with the maximum response at each pixel location (x, y) is the focused pixel to be selected. It can be expressed in the following mathematical form:

   $$d(x, y) = \underset{k}{\operatorname{argmax}}(I_k^s(x, y)) \tag{3.25}$$

   where I_k^s is the sharpness map of the k-th image frame $I_k(x, y)$. The decision map $d(x, y)$ is generated to keep track of the frame number for each pixel so that image fusion can be performed accordingly. Tenenbaum Gradient (Tenengrad) was one of the very first focus measures that was proposed. It is defined as the sum of square of the gradient along the x, and y axes [90]. Since then, more complex measures have been introduced, such as the norm of the image gradient, norm of the image Laplacian [11], energy of the Fourier spectrum [12], and image moments [13]. An extensive review of the popular reference sharpness measures can be found in [174, 139].

   Another set of focal stacking methods are based on multi-resolution transforms. It decomposes the original image slices into several scaled and oriented sub-bands where the saliency of features are measured. Coefficients with the highest responses are selected to build the decision map. Multi-resolution transforms such as Laplacian pyramid, contrast pyramid [170], gradient pyramid, morphological pyramid [133], ratio-of-low-pass pyramid [169] and wavelet decomposition have been used. However, no one has emerged superior.

3. **Output Rendering**

   Rendering the all-in-focus image by selecting the corresponding pixels in the decision map.

This conventional scheme requires a focal stack (at least two frames) to be captured beforehand given that the focus measures it adopts are only capable of comparing in-between frames. Now that we have a sharpness measure that can efficiently detect sharp regions with a single image, the focal stacking process can be performed on the fly without referring to frames before or after it. The proposed focal stacking algorithm has a very simple mathematical form which can be expressed as:

$$S(x,y) = \sum_k (\alpha_k^\gamma * I_k(x,y)) / \sum_k \alpha_k^\gamma \tag{3.26}$$

where α_k is the sharpness value computed from k-th frame I_k at spatial location (x, y). γ is used to "increase the contrast" of the sharpness values. In practice, $\gamma = 3$ works for most cases. Unlike traditional method that select the pixel with the maximum sharpness response, proposed one does a weighted average over all image slices. The traditional scheme is a special case of equation 3.26 when $\gamma = 1$ and α_k only takes discrete values 0 and 1. Averaging multiple frames inevitably result in blurriness in the final fused image. However, in my case, this problem is not as server as the others. As already shown in Figure 3.10, one property of the proposed sharpness metric is that it falls off rapidly with increasing blurriness. A consequence of which is that the weight of the pixel in the sharpest frame approaches 1. Conversely, the weights for the pixels in the blurry frame would be near 0 which makes the impact of these frames on the final fuse image negligible. However, as has already been pointed out in section 3.5, the proposed sharpness map can be disrupted by noise. Moreover, the proposed stacking method operates "on-the-fly" which means there are no additional images from which to build a noise model at every single spacial location as was done in [140]. As such, in order to mitigate the effects of noise, the acquired images have to undergo noise suppression before fusion in high noise condition[5].

One of the seminal denoising methods is the non-local means filter (NLM) [27]. It utilizes redundant information in the image by assuming that a single image always contains patches of similar appearance, and averages these image patches across different spatial locations. A wide variety of work has since then been motivated using nonlocal self-similarity priors, such as BM3D [83], LSSC [117], and EPLL [211]. Although good results can be achieved, the computation cost is high for these methods which makes it unsuitable for use in real-time. An edge-preserving smoothing method called guided image filtering is thus adopted here for noise suppression. The noise estimation algorithm proposed in [109] is adopted to control the degree of smoothing in guided filtering. This estimation can performed in the very beginning of imaging thus can be treated as constant time. The complexity of guided filter is $\mathcal{O}(N)$ where N is the total number of image pixels.

[5]This is unlikely to happen under laboratory image settings with high-end microscope and sufficient ambient light

3.10.1 Data for Focal Stacking Evaluation

A simulated and real-data experiment were carried out to test the performance of this algorithm for both quantitive and qualitative evaluation.

- **Simulated Data**

  The simulation process is adopted from [139]. A defocused image $I_d(x, y)$ of a 2D planar scene can be simulated as a convolution of the all-in-focus image $I(x, y)$ of the scene with the blur kernel k:

  $$I_d(x, y) = I(x, y) \otimes k \tag{3.27}$$

  where k is constant across the image plane and usually referred as Point Spread Function (PSF) given that it is the shape of blur formed by a point source. For an ideal lens with circular aperture, this shape is known as the Airy disc and can be approximated by a Gaussian function [136, 164, 192] in diffraction limited optics with polychromatic incoherent illumination. The σ of the Gaussian function controls the amount of defocus. The relation of σ and the depth of the scene u was derived [142, 175] as

  $$\sigma_k = \gamma \frac{|u - u_f|}{u(u_f - f)} \text{ with } \gamma = \frac{\kappa f^2}{F} \tag{3.28}$$

  where u_f is the depth of the scene that is in-focus at current camera settings and f is the focal length; γ is a camera-dependent constant and is fully determined by the current camera settings; F here stands for the F-number which is computed as the ratio of f and lens diameter d; κ is the pixel density of the sensor. By grouping the effects of the physical parameters of the lens in a single constant γ, this model simply expresses the blur radius as a function of the target position u and the focal length f.

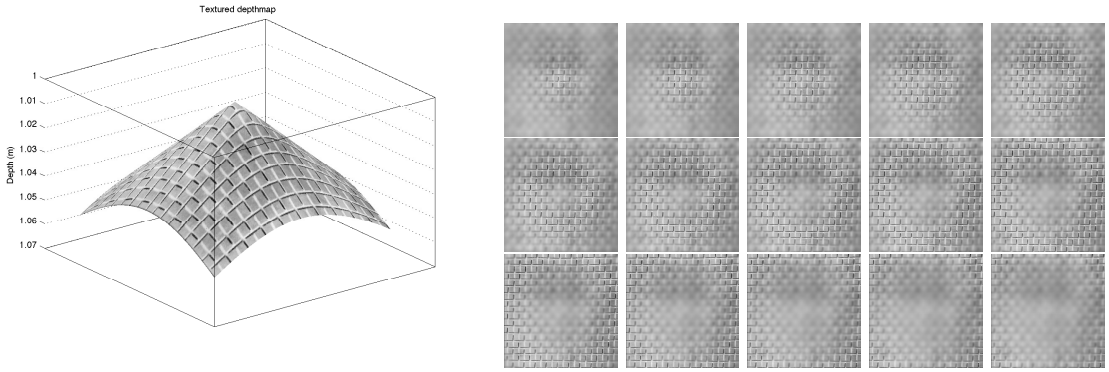
  For a 3D scene, the image formation model looks different from the one in equation (3.27) because the PSF varies in the image plane with regard to the depth of the scene. For every scene point at coordinate (x, y), the response on the sensor can be expressed as

  $$B = a \otimes k_{(x,y)} \tag{3.29}$$

  where a is the radiance of the scene point and $k_{(x,y)}$ is the depth-related PSF. The defocused image in this case thus can be obtained by adding up every point's contribution in the 3D scene.

  $$I_d(x, y) = \sum_{i=0}^{W-1} \sum_{j=0}^{H-1} B(i - x, j - y) \tag{3.30}$$

  W and H are the width and height of the imaged scene. In the implementation, the response of a particular scene point (x_0, y_0) can be simplified by only summing over neighbouring points that are $3\sigma_k$ away from x_0, y_0. Figure 3.19 shows the blur sequence simulated for a lens with $f = 50$mm, F/2.0, $\kappa = 1.6e5$ (Canon EOS 5D Mark III, full frame sensor size with 5760×3840 pixels). The simulated scene has a cone shape with the depth spanning from 1m to 1.05m. In order to test the general applicability



(a) (b)

Figure 3.19: (a) shows the simulated cone shaped object and the corresponding setting of the image equipment. (b) shows the simulated blur image sequences.

of the algorithm on a variety of scene textures, the underlying all-in-focus images, shown in Figure 3.21, were selected from Brotaz texture dataset [2].

- **Real Data**

  Real world image sequences are adopted for qualitative evaluation. All of the image sequences are from the seed dataset as will be shown in chapter 4.

3.10.2 Evaluation conditions

The following two conditions were applied to simulated data as did similarly in [140].

- **Varying Noise Level**

  A CCD camera has several primary noise sources, such as fixed pattern noise, dark current noise, shot noise, amplifier noise and quantization noise [67], and can be categorized into two groups, irradiance-dependent and irradiance-independent sources. As such, a noisy image can be modelled as

$$I(x,y)_n = f(I(x,y) + n_s + n_c) + n_q \tag{3.31}$$

  where $I(x,y)$ is the original image, $f()$ is the camera response function (CRF, the image brightness as a function of scene irradiance). n_s is the irradiance-dependent noise component, n_c is the irradiance-independent noise, and n_q is the additional quantization and amplification noise [105, 173]. Because most cameras now can achieve very low n_q, it is neglected in this noise model [105], n_s and n_c are assumed to have zero mean and variances $Var(n_s) = I\sigma_s^2$ and $Var(n_c) = \sigma_c^2$, respectively. As found in [105], $\sigma_s = 0.16$ and $\sigma_c = 0.06$ result in very high noise, so these two values are set as the maximum of the two parameters. I sampled σ_s from 0.00 to 0.16 with step size 0.016 and sampled σ_c from 0.01

to 0.06 with step size 0.006. It can be mathematically expressed as:

$$\begin{matrix}\sigma_s = 0.16/10 * \text{NLevel} \\ \sigma_c = 0.06/10 * \text{NLevel}\end{matrix}, \text{NLevel} = 0, \ldots, 10 \tag{3.32}$$

In the upper portion of Figure 3.20, I selected one natural image and added varying degrees of noise so that the reader can have a good comprehension of the noise levels differences.

- **Varying Contrast Level**

  Image contrast is another common factor that could modify the image content, which in turn can affect the performance of sharpness measures. Lowing the contrast of images will make it harder to measure the relative degree of focus because of the smoothing of edges. In order to assess the robustness of the proposed metric to reductions of image contrast, sequences of images with the same content but decreasing contrast were generated. In particular, for every image sequence, contrast was reduced by performing the following operation:

$$I_c(x, y) = \frac{\text{Clevel}}{10}(I(x, y) - 128) + 128, \text{CLevel} = 0, \ldots, 10 \tag{3.33}$$

  where $I(x, y)$ is the intensity of the original image and $I_c(x, y)$ is the generated low contrast version. The same natural image was chosen and the above contrast transform was applied to it for visual comprehension (lower portion of Figure 3.20).

3.10.3 Evaluation Metric

The quality metrics employed for the evaluation of the focal stacking quality are gray-scale structural similarity [188] (SSIM) and peak signal-to-noise ratio (PSNR). PSNR, along with its related quantity mean square error (MSE), are commonly used to objectively quantify the difference between the distorted image and the reference. SSIM, on the other hand, measures the perceived changes in structural information and is deemed to be better conform to the human visual system (HVS). The mathematical definition for each metric is:

$$PSNR = 10\log_{10}\left(\frac{peakval^2}{MSE}\right) \tag{3.34}$$

where *peakval* is the maximum value in the current range of the image datatype or can be specified by the user, and $MSE = \frac{1}{N}\sum_{x,y}(I(x, y) - \hat{I}(x, y))^2$

$$SSIM = \frac{(2\mu_I\mu_{\hat{I}} + C_1)(2\sigma_{I\hat{I}} + C_2)}{(\mu_I^2 + \mu_{\hat{I}}^2 + C_1)(\sigma_I^2 + \sigma_{\hat{I}}^2 + C_2)} \tag{3.35}$$

where μ_I, $\mu_{\hat{I}}$, σ_I, $\sigma_{\hat{I}}$, $\sigma_{I\hat{I}}$ are the means, standard deviations, and cross-covariance for images I, $\hat{I}$. In practice, the SSIM index is computed locally rather than globally to account for the spatially non-stationary property of natural images and the mean SSIM index is reported instead.

Figure 3.20: Visual examples for different noise and contrast levels. Note that Nlevel = 0 and Clevel = 0 corresponds to the original image. This flower image used here is solely for illustration. The underlying image by Johnson Cameraface is licensed under CC BY-NC-SA 2.0.

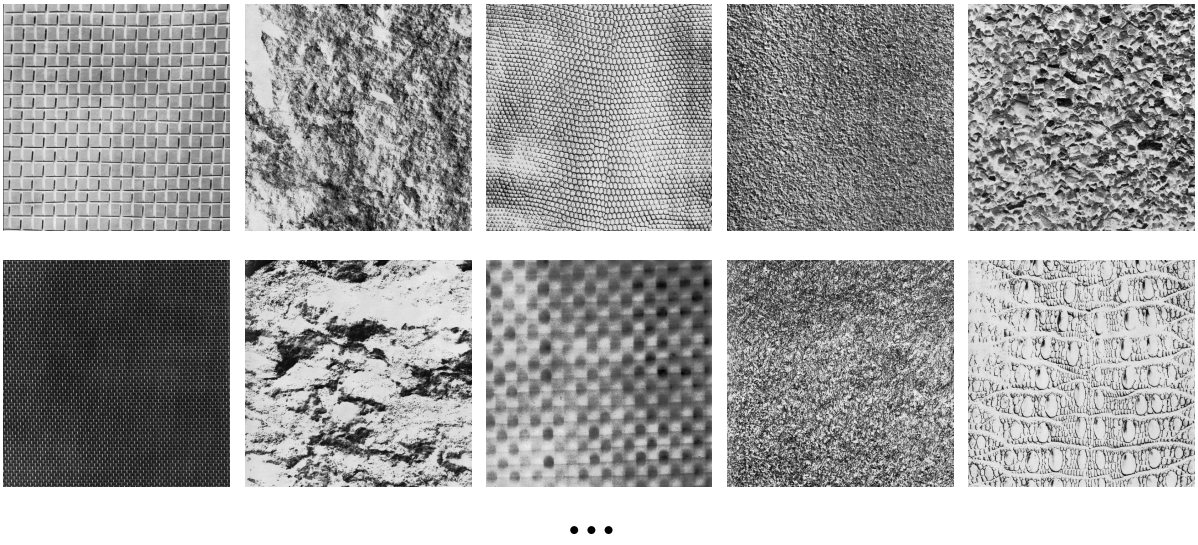


Figure 3.21: Texture variations in the Brotaz texture dataset. Only 10 images are shown. A complete view can be found in [2].

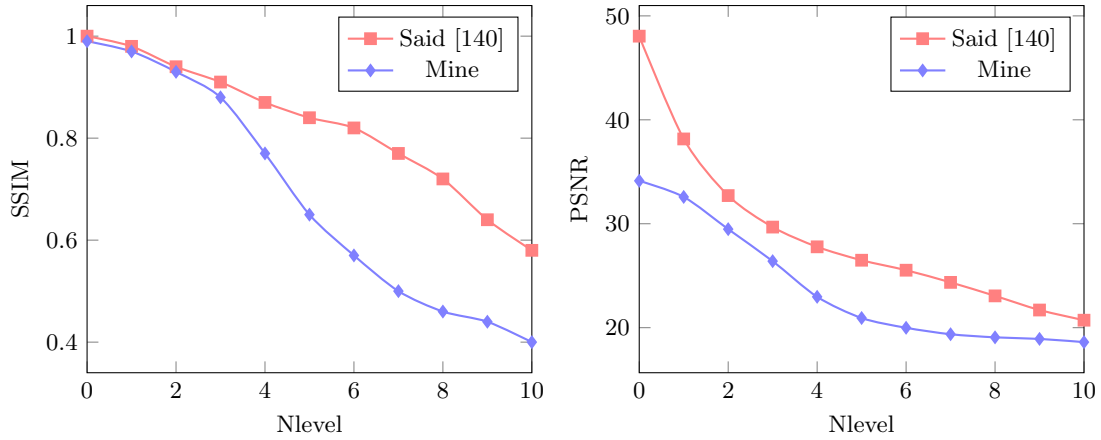


Figure 3.22: Focal stacking performance under different level of noise.

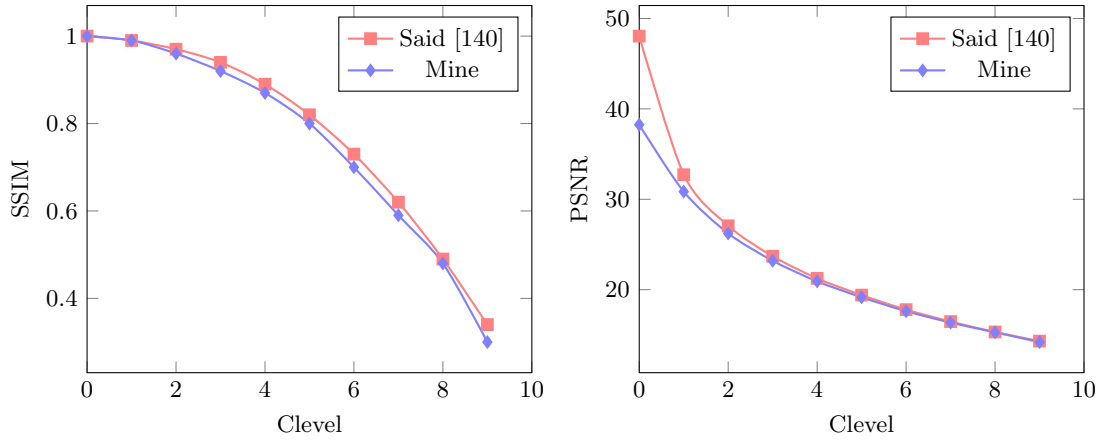


Figure 3.23: Focal stacking performance under different level of contrast. level 0 corresponding to the original contrast whereas level 9 corresponding to 90% contrast reduction

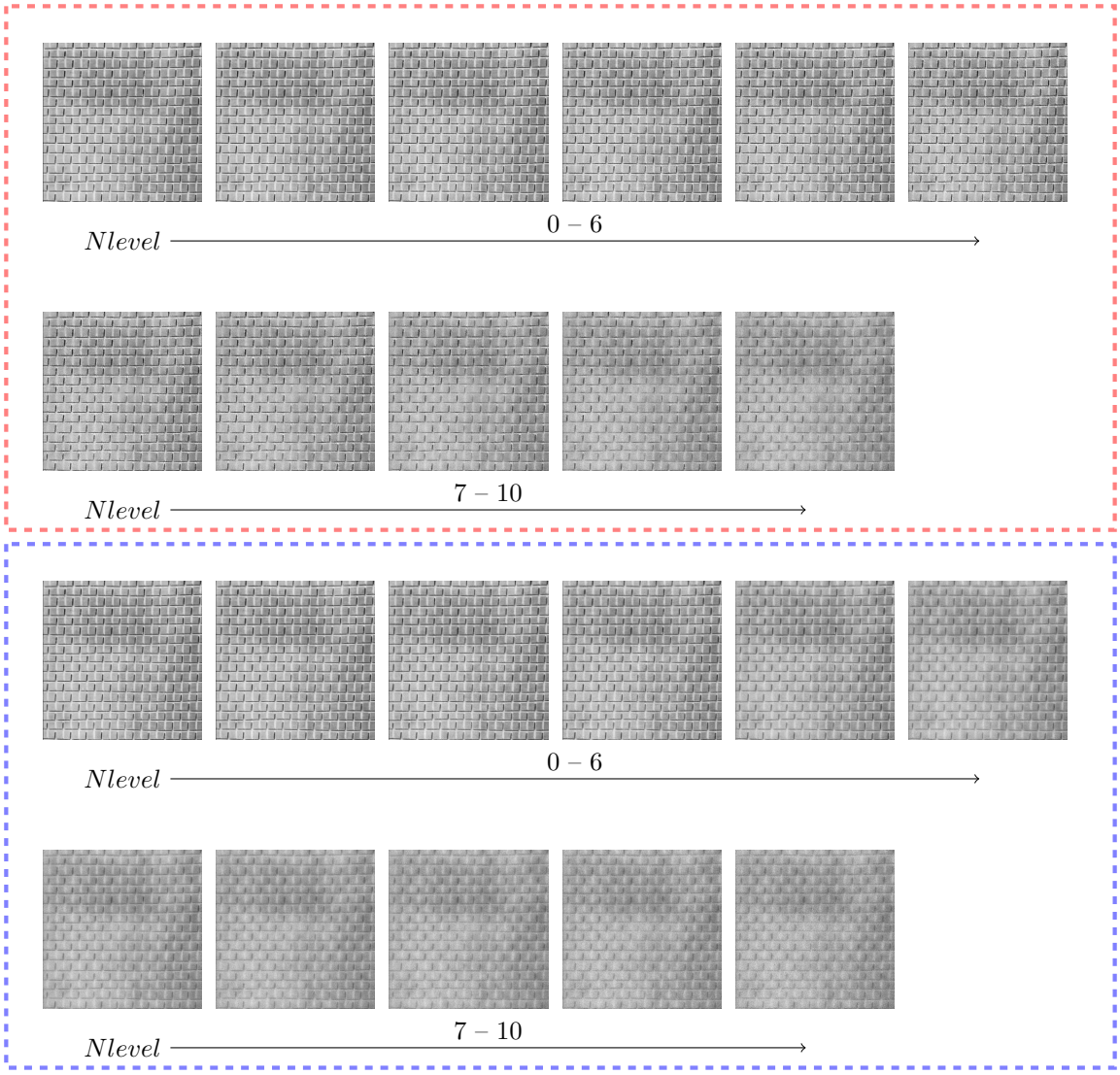


Figure 3.24: Visual results for stacking at different noise levels on simulated image sequences. Images in the red dashed box (top two rows) are created by Said et al. [140], whereas those in the blur dashed box (bottom two rows) are created by the proposed method. Zooming on digital version of this paper for better comprehension.

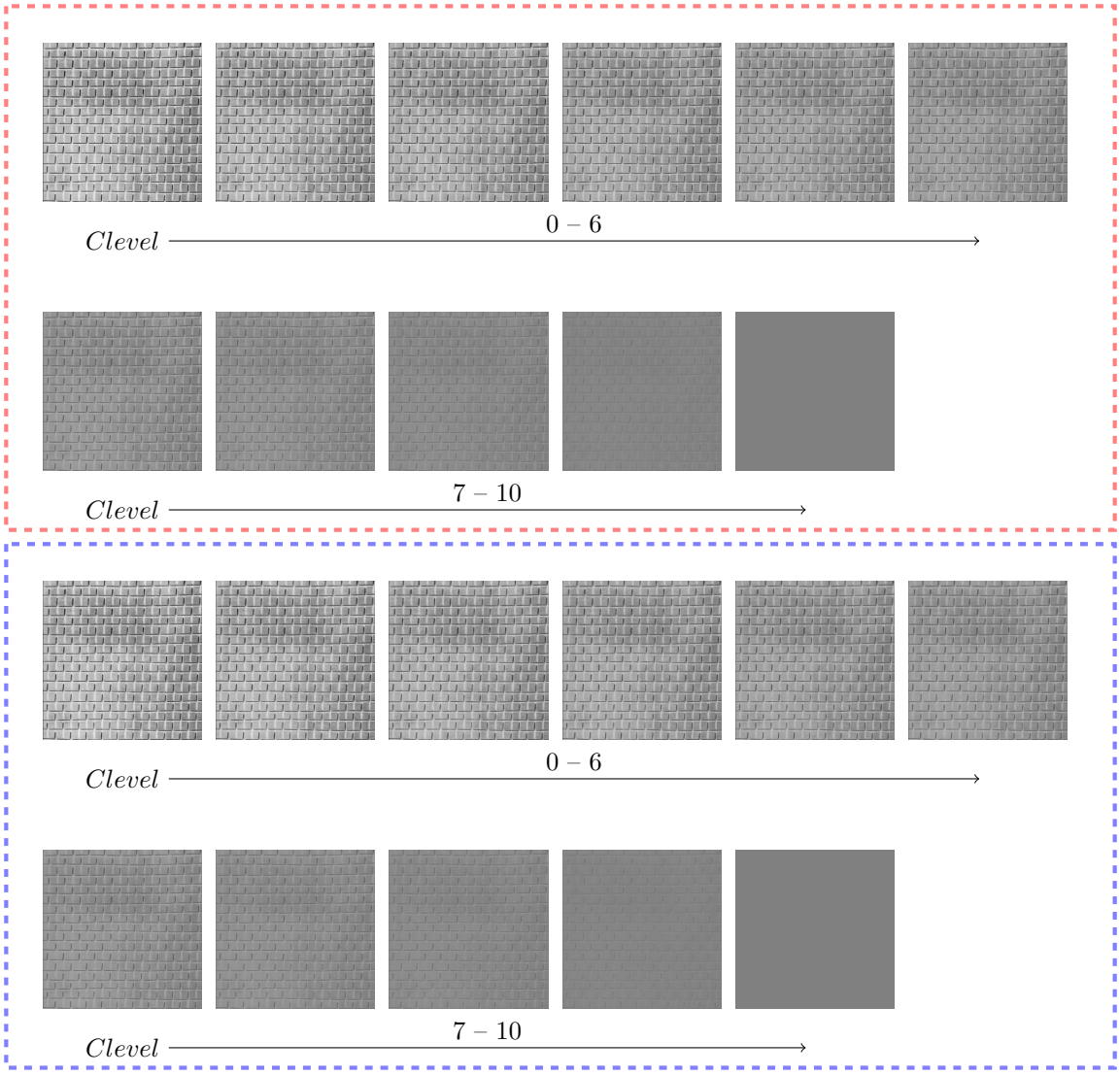


Figure 3.25: Visual results for stacking at different contrast levels on simulated image sequences. Images in the red dashed box (top two rows) are created by Said et al. [140], whereas those in the blur dashed box (bottom two rows) are created by the proposed method. Zooming on digital version of this paper for better comprehension.

3.11 Results and Discussion

As can be seen from Figure 3.22, the proposed method behaves almost the same compared with the state-of-the-art at low noise level (< 3). We see the same trends in the performance under different contrast levels shown in Figure 3.23. At high noise level ($>= 3$), the performance starts to deteriorate which is not surprising because unlike traditional method that select the pixel with the maximum sharpness response, proposed one does a weighted average over all image slices. In high noise level cases, blurry parts start to have larger weights which makes the fused image blurry. However, these high level of noises would be unlikely to occur in the laboratory settings given the high-end imaging equipment and sufficient ambient light. Figure 3.24 and 3.25 give a visual demonstration of what the stacked images look like under the specified noise and contrast level.

In addition, I also attached some visual results for qualitative evaluation on those raw seed image sequences that used to produce Figure 3.26. There is virtually no visually detectable difference on these results. The colour difference between these two groups of images is because of the white balance correction that is performed manually by the image technician given the constant shown red hue of the raw image slices. A big advantage of the proposed method is that the complexity is much lower which allows for realtime stacking. The proposed method has a complexity of $\mathcal{O}(KN)$ where Said et al. has $\mathcal{O}(KNr^2)$. K is the number of image frames; N is the number of pixels in the image and r is the radius of the window for the evaluation of the sharpness.

3.12 Conclusion

I have proposed a very simple yet effective no-reference sharpness metric of time complexity of $\mathcal{O}(N)$ that is capable of run in realtime on a single core cpu. It better measures the sharpness on low contrast sharp regions and behaves monotonically to the increased extents of defocus blur. A single-image-based defocus segmentation algorithm that is also of time complexity of $\mathcal{O}(N)$ was developed on top and achieved state-of-the-art performance. The segmentation algorithm is not only suitable for defocused microscopic images but also for complex natural scenes.

I used this proposed metric also for online focal stacking and achieved results comparable with state-of-the-art under low noise conditions. The performance is also robust to varying contrast and it behaves almost the same as that of Said et al.. It does not require a stack of images to be captured in-prior and the complexity is $\mathcal{O}(KN)$ as opposed to $\mathcal{O}(KNr^2)$ of Said et al. In the user study that will be discussed in chapter 5, I applied the proposed focal stacking method to create all-in-focus images instead of defocus segmentation because of the extreme shallow depth-of-field of the used microscope. Now that we have efficient method to produce all-in-focus seed images, we can proceed to the next chapter to discuss the proposed discriminative image representation.

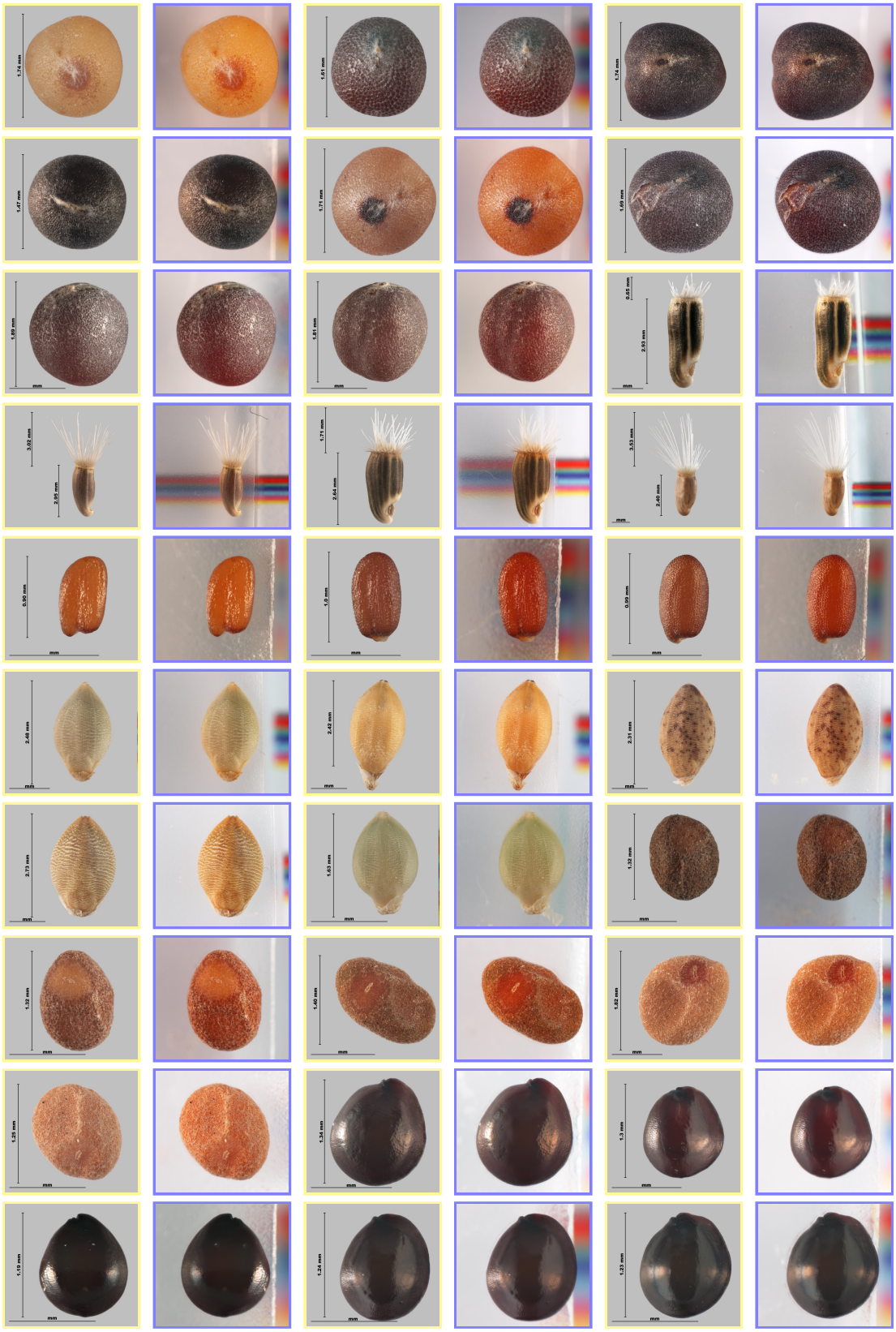


Figure 3.26: Visual comparison of focal stacked seed images. Images with yellow boundaries (odd columns) are those that will be used in Chapter 4. They are stacked with Nikon's proprietary software – NIS-Elements, whereas those with blue boundaries (even columns) are from the proposed method. Zooming on digital version of this paper for better comprehension. Note that yellow boxed images are colour corrected by the image technician.

Chapter 4

A new mid-level feature for textured objects of known scale

A part of this chapter was submitted to Machine Vision and Application (MVAP) with Xin Yi as the lead author.

4.1 Introduction

From the discussion from Chapter 2 we can see that for fine-grained identification tasks, researchers have tried to incorporate object-specific prior information into the identification model, e.g. landmark points on plants are priors only known by botanists; object parts' locations (birds' head, airplanes' propeller, etc) are priors annotated by outsourced person. This information is useful but requires extra human labor and is not available for other datasets. In this chapter, we explore using accurate scale information given by a calibrated microscope as an alternate prior.

The representations of images are categorized into two levels for the traditional BoW identification model; the low-level representations (a set of local descriptors that extract information in the pixel domain), and the mid-level representations that manipulate the low-level descriptors and produce a fixed length feature vector as the image signature. This model is usually referred as *shallow representation* because it only has two levels of abstractions. In contrast, CNN-based representation are often referred as *deep representation* because of its larger number of hidden layers. Hierarchical levels of abstraction have been shown by some visualization literatures [200, 116, 198]. The learned weights in the first layer are always image edges in various orientations and weights in deeper layers have increasingly semantic meanings such as car wheels or human eyes.

In this Chapter I have proposed an image representation for plant seeds based on the BoW model. The method used can be considered an extension of pyramid matching at the scale level. Pyramid matching is widely used in identification tasks where images share the same configuration, for example, natural scenes that are all upright. Seed images do not possess this property, but the pyramid can be formed in the scale dimension instead of the spatial dimension when images from the same class have limited scale variance. The details of the method can be found in Sect. 4.2. This model is validated experimentally on the task of discriminating 30 seed species (Sect. 4.4). It is shown that the proposed method produces better classification

results compared to pre-trained CNN-based methods.

4.2 Multi-scale Image Representation

Seed identification is expected to be conducted by seed analysts inside a laboratory, where imaging systems can be pre-calibrated to give accurate scale information (pixel size). Due to the well-constrained imaging setup, all the seed samples' surface textures can be clearly rendered. A direct outcome of this setup is that we can image samples of the same species at the same scale. In this section I describe the mid-level feature used for representation of this kind of seed images and how real pixel-scale information is incorporated.

4.2.1 Multi-scale vs. Single Scale

Multi-scale representations have been exploited in many different tasks. For example, Bertasius et al. detect edges in increasing window sizes [19]. Li et al. extract features on three nested windows with increasing size for salient region detection [103]. Zheng et al. partition images with three different rectangular grid sizes and extract features on them to represent global and regional context for the task of image retrieval [205]. Shi et al. compute a sharpness measure on three overlapped localized windows and combined them to produce a single sharpness score with a multi-scale graphical model [159]. A conclusion that can be made from these works is that analysis at multiple scales is generally superior to analysis at a single scale.

In object identification, features are usually extracted on different sizes of local windows and then encoded and pooled into a single feature vector to achieve scale invariance. It is uncommon to see representations from different spatial sizes concatenated except for cases where spatial arrangement of features is important. Spatial pyramid matching works effectively on datasets like SUN [193], and MIT Indoor [144] because scenes in these datasets have coherent "canonical composition" [97] (ground at the bottom of the image and sky on top). In this context, "multi-scale" refers to a nested pyramid of regions. The multi-scale framework introduced by Gong et al. [55] uses spatial pyramids but with CNN features. They achieved state-of-the-art results on scene classification tasks but not on the general classification task (ILSVRC 2012). The author claimed that this might be due to the underlying implementation of neural nets. I would also argue that this could also result from the large pose and scale variations of objects in the dataset which makes concatenation in the spatial domain less effective.

Since, multi-scale is beneficial for all kinds of tasks, herein I also extract features at different scales but instead of pooling them all together, I scale-normalize the feature representation by using actual pixel scale information and concatenate them to conduct a scale-wise comparison of the objects in the classification phase.

4.2.2 Fixed Scale vs. Detected Characteristic Scale of the Keypoint

One of the fundamental problems in analyzing real-world images is that objects may have different appearance depending on the scale of observation. In most cases, the scale information required to represent the image features at an appropriate scale is unknown. If so, the only reasonable approach is scale estimation. Indeed, many modern computer vision systems are now equipped with automatic scale estimation mechanisms. The most commonly adopted framework for performing scale estimation is detection of local extrema over scale through γ-normalized derivative expressions [104]. For example, SIFT detects scale via local extrema over a scale-normalized difference of Gaussian (DoG) pyramid. Unless otherwise mentioned, we use the same definition as in [13, 112, 120], which is that scale is the standard deviation of the Gaussian function and is related to the bin size of the support region that descriptor is built upon by a magnification factor. The measurement unit is pixels. Scale invariance is useful in situations where large variations in scale exist. However, Mikolajczyk reported that, "under a scale change factor of 4.4, the percentage of pixels for which a scale is detected is as little as 38% for the DoG detector and only 10.6% of the detected scales were correct" [120]. In light of this, I matched four pairs of same-species seed images using locally detected keypoints computed using the same matching scheme used by Lowe [112]. One would expect that matched keypoints would be of the same scale. However, it was found that most of the matched local structures do not have the same scale, as shown in Figure 4.1.

Probably due to this unstable scale estimation, some studies have found that the dense version of SIFT (DSIFT) is better than the sparse version in classification tasks [22, 23]. DSIFT descriptors are computed from keypoints on a regular grid with a spacing of G pixels and the scale of the points are explicitly selected instead of estimated as in Lowe's method. This approach was proven successful by the results of the OXFORD_VGG system of the 2012 ImageNet challenge [5]. It is a good fit for the seed identification problem since dense keypoints of constant scale will capture structure in texture coherently given that there are limited scale changes across the same seed species.

4.2.3 Multi-scale Concatenation

We propose a multi-scale representation of image descriptors to incorporate real image scales (measured in µm). On each densely sampled pixel, the descriptors are computed over M circular support patches with different radii (which are predetermined, as mentioned in the previous section). The final image representation is the concatenation of pooled feature vectors across these M different scales as shown in Figure 4.2.

The final image representation uses the same Fisher encoding method described in [138]. In Fisher encoding, statistics of feature descriptors are learned by Gaussian mixture model (GMM) with K components:

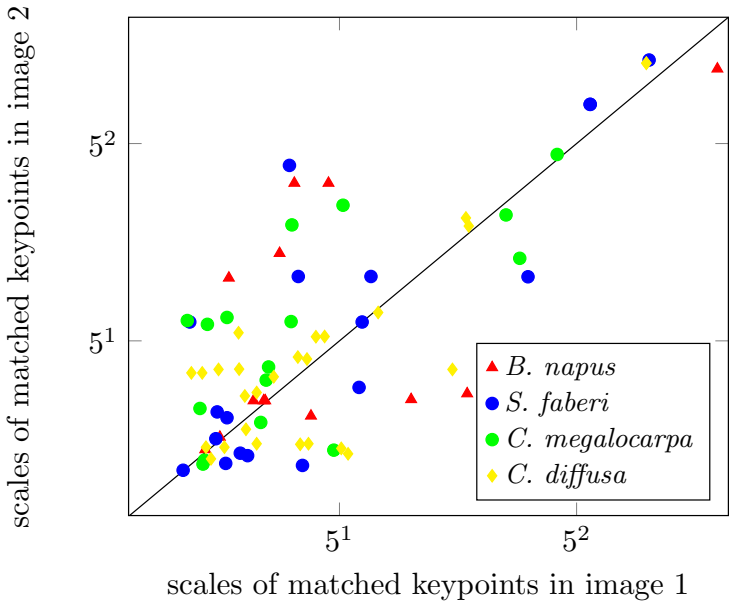


Figure 4.1: Keypoint matching of four pairs of seed images of the same species (*B. napus*, *S. faberi*, *C. megalocarpa*, and *C. diffusa*). This scatter plot of the estimated scales of the matched keypoints shows that matched keypoints often have different estimated scales. A log-scale was used on the axes to compress the dynamic range of scales.

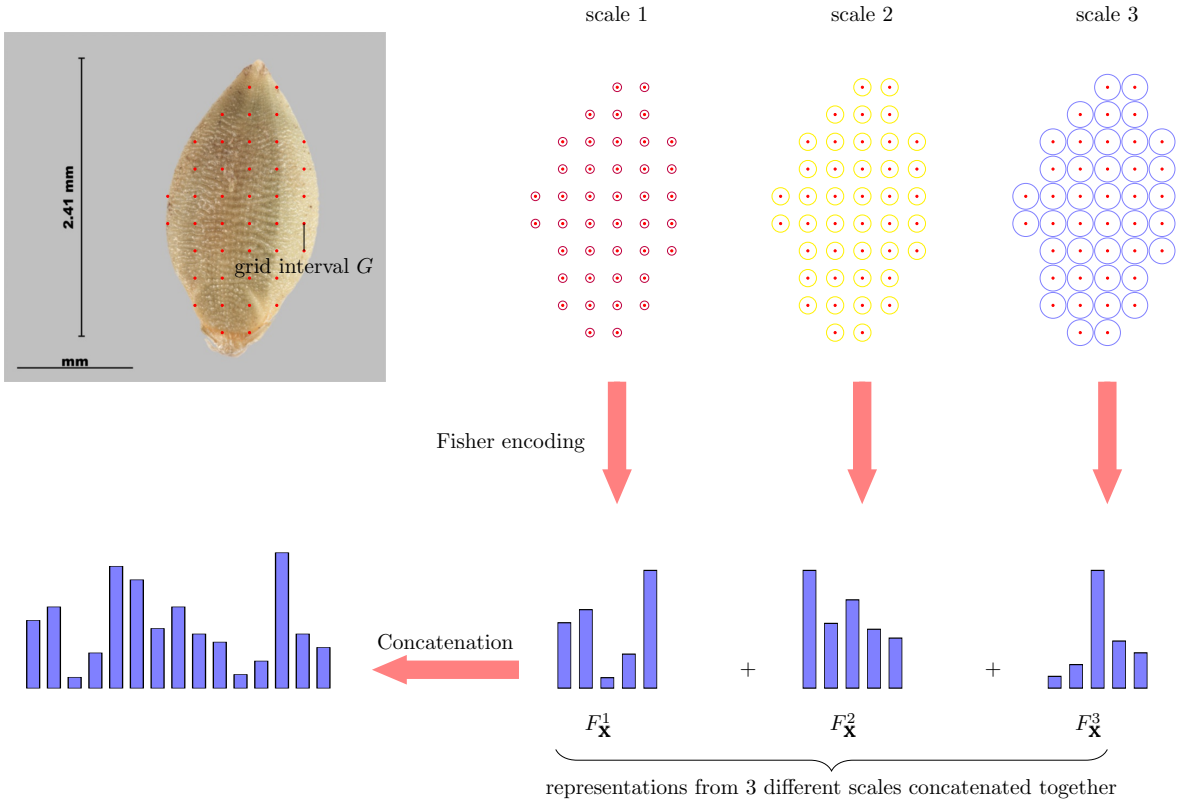


Figure 4.2: Seed representation. Dense SIFT descriptors extracted on M (here $M = 3$) different scales with grid spacing 10 pixels. For each scale, the corresponding representation is achieved by maxpooling of Fisher-encoded descriptors. Let $F^i_{\mathbf{X}}$ denote the Fisher-encoding of vector $\mathbf{X}$ at scale i.

$$
\begin{aligned}
p(\mathbf{x}|\boldsymbol{\theta}) &= \sum_{k=1}^{K} p(\mathbf{x}|\mathbf{v}_k, \boldsymbol{\Sigma}_k)\pi_k \\
\text{with } p(\mathbf{x}|\mathbf{v}_k, \boldsymbol{\Sigma}_k) &= \frac{1}{\sqrt{(2\pi)^D det(\boldsymbol{\Sigma}_k)}} e^{-\frac{1}{2}(\mathbf{x}-\mathbf{v}_k)^\top \boldsymbol{\Sigma}_k^{-1}(\mathbf{x}-\mathbf{v}_k)},
\end{aligned}
\tag{4.1}
$$

where $\boldsymbol{\theta} = (\pi_1, \mathbf{v}_1, \boldsymbol{\Sigma}_1, \ldots, \pi_K, \mathbf{v}_K, \boldsymbol{\Sigma}_K)$ is the vector of parameters of the model, and $det(\cdot)$ is the matrix determinant. To be more specific, π_k are the weights for each distribution; $\mathbf{v}_k$ is the mean of the k-th cluster and $\boldsymbol{\Sigma}_k$ is the covariance matrix of the k-th cluster. Fisher encoding computes the derivative of the log-likelihood function with respect to the various model parameters:

$$
\nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}|\boldsymbol{\theta}). \tag{4.2}
$$

To ensure that the resulting vectors can be meaningfully compared, Eq. (4.2) is whitened by multiplying with the inverse of the square root of the Fisher information matrix $\mathbf{H}$ [177]. The encoded descriptor can be expressed as:

$$
\Phi(\mathbf{x}) = \mathbf{H}^{-\frac{1}{2}} \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}|\boldsymbol{\theta}). \tag{4.3}
$$

Note that the GMM was fitted to descriptors from all scales. In my case, descriptors come from M different scales $\{s_1, s_2, \cdots, s_m\}$. Suppose descriptor $\mathbf{x}$ is computed from support region with scale s_m, in my encoded feature vector $\Phi(\mathbf{x})'$, only the part of encoded descriptors that are related to a corresponding scale are the same as in equation 4.3, and the others are set to 0:

$$
\Phi(\mathbf{x})' = [\mathbf{0}, \cdots, \mathbf{H}^{-\frac{1}{2}} \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}|\boldsymbol{\theta}), \cdots, \mathbf{0}]. \tag{4.4}
$$

The dimension of the final representation is thus $2 * N * K * M$ in which N is the dimension of the descriptor, K is the size of the vocabulary and M is the number of chosen scales.

Note that the proposed multi-scale representation differs from those commonly described, e.g. in [33], in that their representation is a pooling of all multi-scale descriptors whereas the one proposed here pools descriptors in a scale-wise manner and concatenates pooled descriptors from different scales together. Moreover, the scale used in the above representation is a real scale which measured in µm.

4.2.4 Extension of Pyramid Match Kernel

The authors of [97] note that matching images from different spatial resolutions has proven to be efficient and has been adopted by many computer vision systems in which the images have a coherent composition. However, for objects like seeds, which lack a fixed orientation, this technique can not be directly utilized. Recall that an image can have multi-scale representation created by recursive convolution with a Gaussian filter (by subsampling the filtered image successively, resulting in the pyramid representation). Since seed images were captured with a calibrated microscope, the image comparison can be conducted in a scale-wise manner. Let $\mathbf{X}$ and $\mathbf{Y}$ be two sets of vectors extracted on support regions with a sequence of M scales

that reside in a d-dimensional feature space. Let $F_{\mathbf{X}}^m$ and $F_{\mathbf{Y}}^m$ denote the Fisher-encoded vector at scale $m, m = 1, \cdots, M$. Then the distance between subsets of $\mathbf{X}$ and $\mathbf{Y}$ that have the same scale m is given by the linear kernel $\mathcal{I}$:

$$\mathcal{I}(\mathbf{X}^m, \mathbf{Y}^m) = (F_{\mathbf{X}}^m)^T F_{\mathbf{Y}}^m \tag{4.5}$$

where $(\cdot)^T$ denotes the matrix transpose. The difference here from [97] is that they used a histogram intersection kernel to measure the hard assignment encoded feature vector as the distance, whereas I used a linear kernel to measure the Fisher-encoded feature vector as the distance.

Since a larger image patch conveys different information from a small image patch centred around the same pixel (compositional information vs. fine detail), the weight associated with every scale is set equally to 1. The intuition here is that seeds from the same species should have the same appearance at arbitrary scales. Putting all the pieces together, the pyramid match kernel can be expressed as:

$$\mathcal{K}^M(\mathbf{X}, \mathbf{Y}) = \sum_{m=1}^{M} \mathcal{I}(\mathbf{X}^m, \mathbf{Y}^m) \tag{4.6}$$

It can be implemented in practice as a long vector formed by concatenating the equally weighted Fisher vectors at all scales.

4.3 Dataset and Experimental Protocol

4.3.1 Dataset

The eleven *Brassica* species and small mustards of *Brassicaceae* family (group 1 + group 2B) were selected to represent small round seeds with surface texture patterns and hilum[1] position; the four *Centaurea* species of *Asteraceae* family (Group 2A) were selected to represent longer seeds with shape variation and special feature–pappus; the five *Setaria* species of the *Poaceae* family (Group 3) were selected to represent dual sided seeds with surface texture; the five *Amaranthus* species of *Amaranthaceae* (Group 5) were selected to represent seeds that have very limited surface features to be distinguished to species level; and the five *Cuscuta* species of *Convolvulaceae* family (Group 4) were added in response to a recent regulation change requiring differentiation of species which imposes identification challenges. All species chosen for image analysis of computer vision are difficult and time consuming species in routine diagnostic testing for seed or phytosanitary certification, and it poses much more trouble to discriminate seeds inside each group than between groups. Seed examples can be seen in Figure 4.3. The seed species names and their abbreviations used in this document are given in Table 4.1.

The images were provided by the Canadian National Seed Herbarium of CFIA. The identity of seed specimens were verified by a taxonomist. To ensure sufficient representation of a species, multiple samples (10 per species) were carefully selected to represent the typical range of feature variations within a population.

[1]The scar on a seed marking the point of attachment to its seed vessel

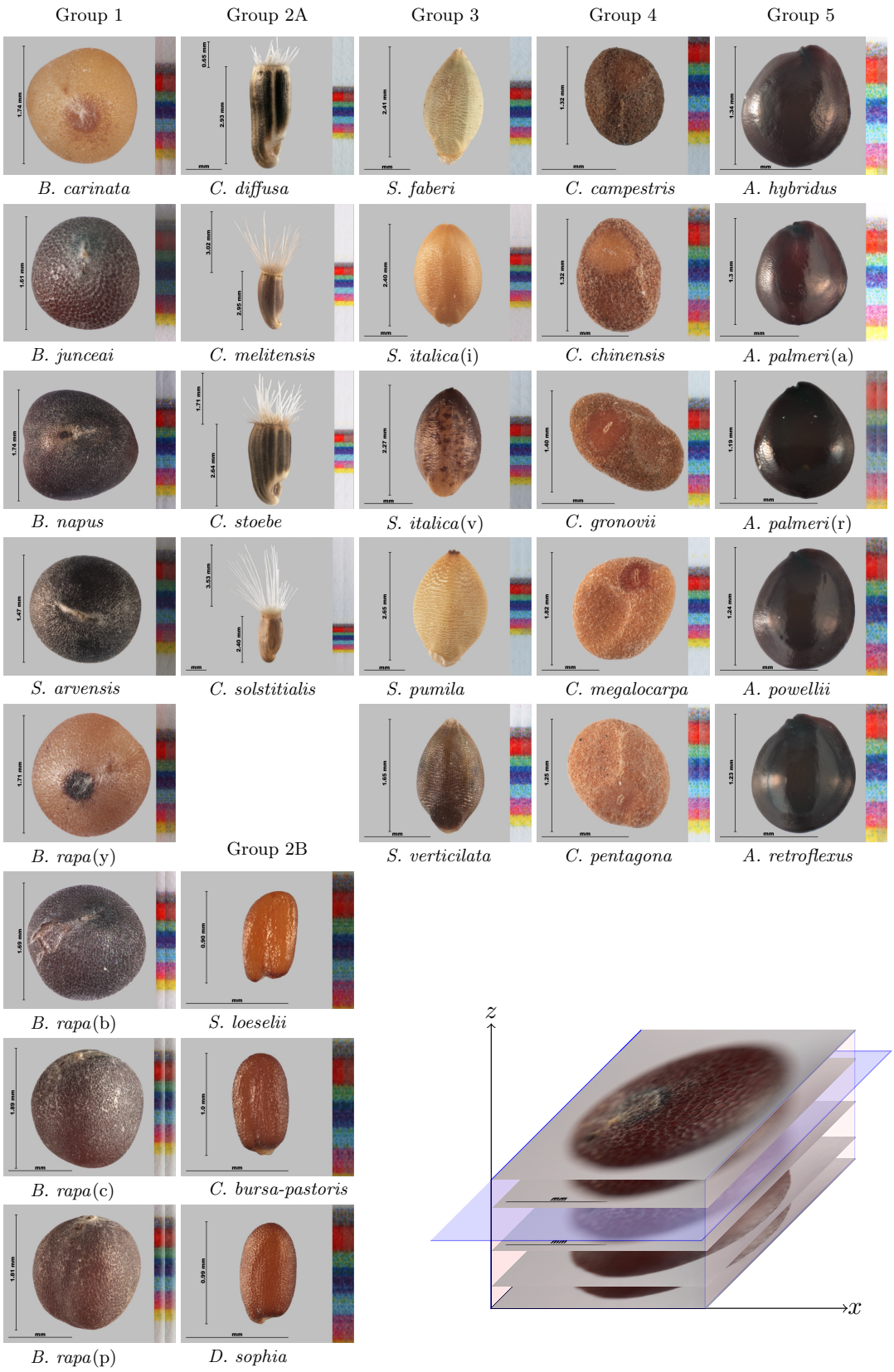


Figure 4.3: Example all-in-focus images from my seed dataset. Each one is composed from multiple image sequences as seen in the lower-right corner. Sub-captions are the short names of seed species used in this paper; the corresponding full names are in Table 4.1. Proposed feature representation of these seed images are visualized in 2D in Figure 4.7

This sample selection process differs from the one of other similar researches where seeds specimens are either unidentified or unrepresentative [63, 113, 31, 204, 56, 57]. Images with multi-focus stacking were acquired using a AZ100M motorized Multi-Purpose Zoom Microscope (Nikon, Tokyo, Japan). Seeds were placed on a glass slide next to a colour map so any colour corrections during editing can be clearly seen. After acquisition, colour correction was performed manually by a technician. Colour editing was restricted to correcting the colour balance of the seed and to making the image have a uniform background. By pre-calibration of the microscope with a stage micrometer, a measurement scale was also included. The final image therefore has the seed, two colour maps (one corrected and the other as the image was taken) and a measurement scale. Each image in the data set is a composite of 50–120 (determined by the size of the seed) image slices taken at different focus points so that the entire seed is in focus. These all-in-focus images were produced by Nikon's proprietary software. The final images have a resolution of 300ppi and a size of 1280×1600 pixels.

4.3.2 Preprocessing

All images were cropped to omit the colour map and scaled down to 640×700 pixels using bicubic interpolation. The uniform gray background allowed segmentation of the seed from the images by thresholding of the S (saturation) channel from the HSV colour model. The resulting segmentations, represented as binary images, then underwent a morphological opening to remove small connected components (resulting from noise) with less than 10,000 pixels. Texture features were extracted from the segmented region of the S channel since it gives the best visual distinction.

The size of pixel in each image was calculated from the length of scale bar. Hough transform was employed to detect the scale bar and the detected scales are listed in Table 4.1. Note that images were captured incrementally at different times. Thus only one size for each species is reported to give a basic comprehension of the seed size. In the real testing case, this information can be entered manually by the analyst who conducts the test.

Dense SIFT features involve two parameters: the grid interval G and radius of the support region R which is related to the chosen scale of the region. A third parameter associated with proposed representation is the set of scales (µm) to pool. Well-performing values of the parameters were selected via a series of experiments, described below.

4.3.3 Experiments

1. **Training and Testing Methodology**

   The following process was used for each classification test made in the course of conducting experiments 2 and 3 (described in subsequent sections). Following standard procedures, the dataset was split into 9 training images (chosen randomly) per seed species (class), and 1 for testing – disjoint from the training images. The classification process was repeated 10 times (ten-fold cross-validation). Features were extracted from each image using the parameters under investigation during a given classification test.

Groups	Seed species	Abbr.	Px. Size
Group 1	*Brassica carinata*	*B. carinata*	3.2 µm
	Brassica junceai	*B. junceai*	3.1 µm
	Brassica napus	*B. napus*	3.4 µm
	Sinapis arvensis	*S. arvensis*	3.0 µm
	Brassica rapa, yellow seed type	*B. rapa*(y)	3.1 µm
	Brassica rapa, brown seed type	*B. rapa*(b)	3.1 µm
	Brassica rapa, subsp. *chinensis*	*B. rapa*(c)	3.3 µm
	Brassica rapa, subsp. *pekinensis*	*B. rapa*(p)	3.4 µm
Group 2A	*Centaurea diffusa*	*C. diffusa*	6.5 µm
	Centaurea melitensis	*C. melitensis*	10.6 µm
	Centaurea solstitialis	*C. solstitialis*	12.4 µm
	Centaurea stoebe	*C. stoebe*	7.0 µm
Group 2B	*Sisymbrium loeselii*	*S. loeselii*	2.1 µm
	Capsella bursa pastoris	*C. bursa-pastoris*	2.1 µm
	Descurainia sophia	*D. sophia*	2.1 µm
Group 3	*Cuscuta campestris*	*C. campestris*	3.0 µm
	Cuscuta chinensis	*C. chinensis*	2.5 µm
	Cuscuta gronovii	*C. gronovii*	3.2 µm
	Cuscuta megalocarpa	*C. megalocarpa*	3.0 µm
	Cuscuta pentagona	*C. pentagona*	2.4 µm
Group 4	*Setaria faberi*	*S. faberi*	4.7 µm
	Setaria italica, subsp. *italica*	*S. italica*(i)	5.1 µm
	Setaria italica, subsp. *viridis*	*S. italica*(v)	4.5 µm
	Setaria pumila	*S. pumila*	5.3 µm
	Setaria verticilata	*S. verticilata*	3.2 µm
Group 5	*Amaranthus hybridus*	*A. hybridus*	2.5 µm
	Amaranthus palmeri amaranth	*A. palmeri*(a)	2.7 µm
	Amaranthus palmeri rennselaer	*A. palmeri*(r)	2.2 µm
	Amaranthus powellii	*A. powellii*	2.3 µm
	Amaranthus retroflexus	*A. retroflexus*	2.3 µm

Table 4.1: Seed dataset composition. 30 species categorized into 5 groups based on the visual similarity, 10 samples per species. The second column shows the binomial name of each seed species. Third column shows the corresponding abbreviation used in the paper. The fourth column shows the size of each pixel in the image. As the sample images are obtained in different time period which result in slight scale change across images. Only the scale for the first set of images are shown. Horizontal divisions separate species in different genera.

A performance score was computed as the average per-class recognition rate which is the proportion of correctly classified images for each of the classes.

2. **Baseline**

   I have compared my proposed method with two baseline methods. The first is multi-scale DSIFT-FV where mutli-scale DSIFT descriptors (pixel size of spatial bins are 4, 6, 8, 10 and grid spacing is 10) are extracted and encoded using Fisher vectors. I used two variants of this method. In the first, denoted as DSIFT-FV-noncat, representations from different scales are pooled into a single Fisher vector (as in [33]). In the second, denoted DSIFT-FV-cat, representations are concatenated. Note that in both variants, the orientations of the support regions are detected, instead of fixed, to achieve rotation invariance.

   The second baseline uses features extracted with a pre-trained deep convolutional neural network (CNN). Two variants of CNN architecture are adopted. The first one is BVLC Reference CaffeNet [89] that was originally trained on ILSVRC 2012 and the other one is VGG-19 [160]. VGG-19 has more layers and uses a smaller convolution kernel than the BVLC Reference CaffeNet. Both networks were obtained from the Caffe model zoo [78]. It might seem inappropriate to use a CNN not trained on seed images because the deeper layers of a CNN are normally domain-specific. But recent works have shown that the deep features work surprisingly well and have surpassed the traditional hand-crafted features on many recognition datasets [145]. The deep feature I use is from fc6 for both BVLC Reference CaffeNet and VGG-19. The feature representation has 4096 dimensions and is L_2-normalized before sending to the classifier. The bounding box of the seed was first found, and the image was then cropped and resized to 256×256 pixels so that it can be fed into the network.

   Linear SVM was trained on top of these feature representations for classification with parameter C (regularization-loss trade off) searched in the range of $[2^{-4}, 2^4]$ and only the best results are reported from ten-fold cross validation.

3. **Experiment 1: Scale Selection**

   The purpose of this experiment was to determine which set of scales are effective. Descriptors were extracted on a grid with a spatial interval of 10 pixels for all 10 image samples of each class. Rather than fixing the rotation of the SIFT descriptors to a constant value, I computed dominant orientations for each local patch to achieve rotation invariance. A series of scales ranging from 1 to 50 µm were examined. The selection was conducted by determining the number of descriptors extracted under each scale. We would expect that more appropriate scales should lead to larger numbers of dominant gradient orientations since multiple orientations would be helpful for describing the patches.

4. **Experiment 2: Selection of Scales to Pool**

   In the proposed representation, the scale of seeds are explicitly incorporated, a consequence of which is

that only seeds from the same species can have the same representation as the training samples of a given species no matter what set of keypoint scales are chosen. Even if some seed species happen to share the same appearance at one scale, when more scales are chosen, the chance of misclassification decreases, because the number of scales at which the appearance differs will increase. Thus, I investigated various combinations of scales in the range of 6 to 20 µm; the restriction to this range, and the specific subsets chosen were based on the outcome of experiment 1. Each subset of scales was tested using the training and testing methodology from Sect. 4.3.3.

5. **Experiment 3: Grid Spacing Selection**

   Using the most promising combination of scales from experiment 2, I tested grid spacings of 5, 10, and 15 pixels. Each test was performed using the training and testing methodology from Sect. 1.

4.4 Results and Discussion

4.4.1 Experiment 1: Scale Selection

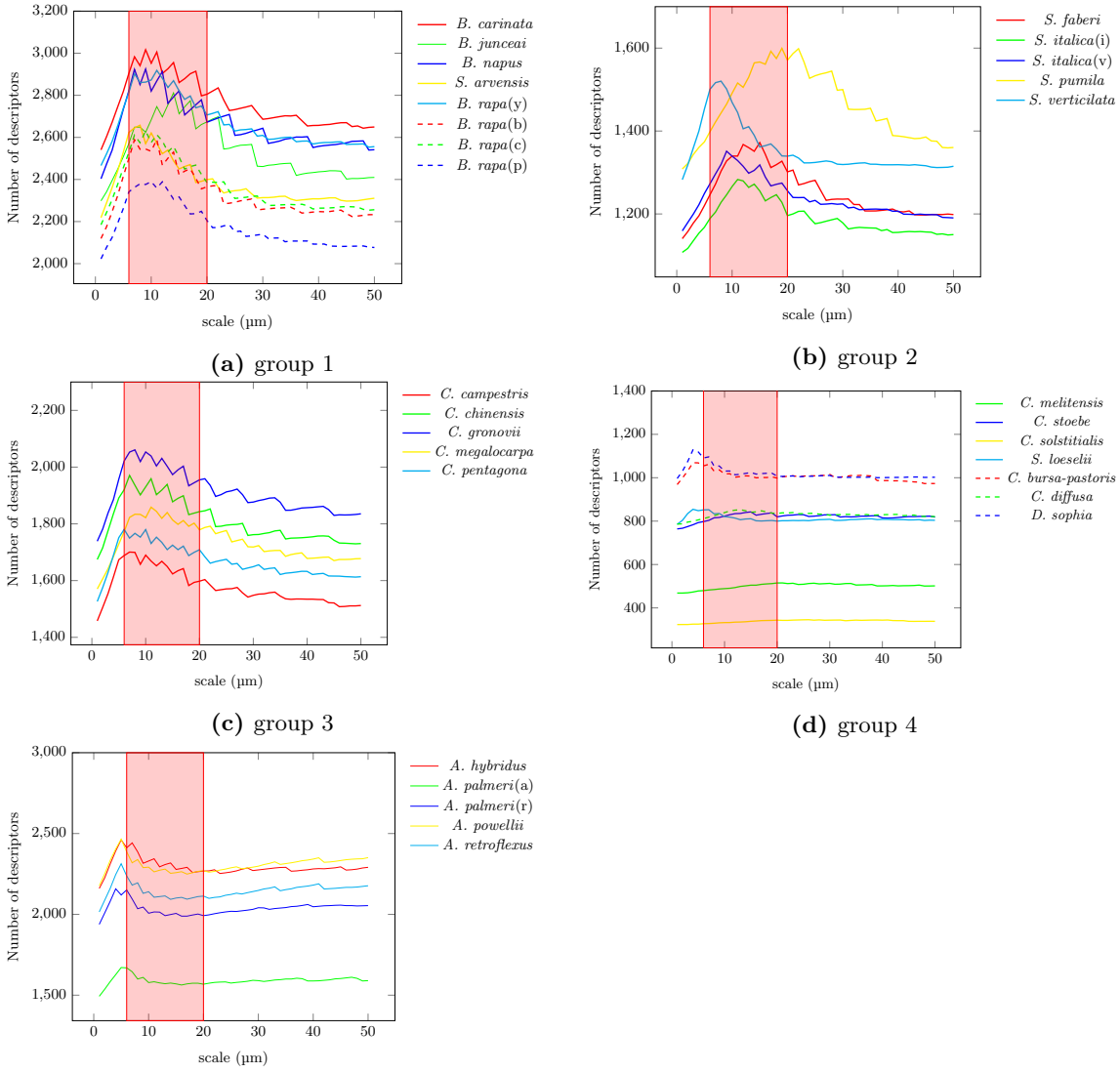
Subfigures (a), (b), (c), (d) and (e) in Figure 4.4, demonstrate that the number of descriptors at first increases with increasing scale, peaks, and begins to decrease once again. The number of grid points is fixed since grid interval is predetermined. Since dominant gradient orientations were detected around these grid points, the changing number of descriptors results from multiple dominant orientations detected by SIFT.

Computing multiple orientations of a patch is beneficial for image matching [112]. For object identification, my experiments have also shown that using multiple orientations can help increase the identification performance of these randomly oriented seeds. In general, large numbers occur for scales between 6 and 20 µm. Reduction of the scale will shrink the support region to that of a single pixel (for large seeds like *C. melitensis*) which is inappropriate for descriptor extraction. Increasing the scale will introduce too much Gaussian blur which will decrease the distinctiveness of the patch. The ideal range of scales has been shaded in red in the plots in Fig. 4.4, and only these scales were used in experiment 2.

4.4.2 Experiment 2: Selection of Scales to Pool

Fig. 4.5 shows the results of pooling features from multiple scales in the range of 6 through 20 µm which were the most promising scales identified by experiment 1. Four combinations of scales were tested over a range of vocabulary sizes.

Generally, the more scales used, the better the results. From a single scale of 6 µm to a pooling of scales 6 and 9 µm, the improvement is large. However, with the pooling of more and larger scales, the degree of improvement decreases. This may be because smaller scale support regions capture fine details that are useful to discriminate morphological similar seeds, but larger scale support regions encode only large scale compositional information which is ambiguous.



(a) group 1

(b) group 2

(c) group 3

(d) group 4

(e) group 5

Figure 4.4: Average number of descriptors extracted on a regular grid for all 10 image samples of each class. Subfigures (a), (b), (c), (d) and (e) show results for seed groups 1, 2, 3, 4 and 5 from Figure 4.3, respectively. Note that number of grid points is fixed when grid interval is predetermined. Thus the changing number of descriptors is caused by the detection of multiple dominant orientations at a given grid point by SIFT. The selected scale range is shaded in red.

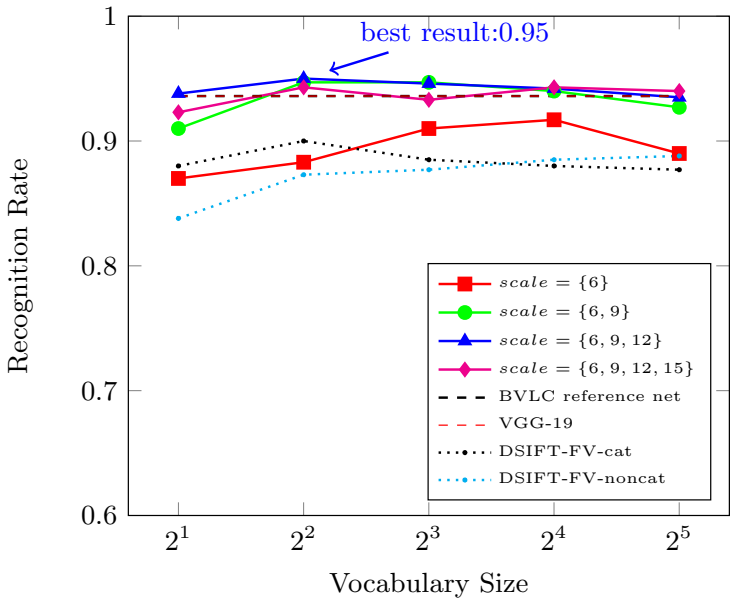


Figure 4.5: Effect of scale pooling on the classification results. Grid interval G is set to be 10 pixels in this experiment. Note that results from BVLC and VGG-19 are overlapped.

Scale set $\{6, 9, 12\}$ gets the best recognition rate at vocabulary size 4, thus I selected pooling of the set of scales $\{6, 9, 12\}$ for experiment 3.

4.4.3 Experiment 3: Grid Spacing Selection

The effect of grid interval on the classification performance when using the set of pooled scales $\{6, 9, 12\}$ is shown in Fig. 4.6 for a range of vocabulary sizes. It can be seen that using a finer grid interval leads to better performance for all vocabulary sizes tested. This is because smaller intervals cause a larger area of the seed to be covered by patches. However, the tradeoff is a greater computational burden due to the increased number of keypoints, and the consequential requirement of larger vocabulary size.

4.4.4 Discussion

The classification performance generally increases for denser grids and pooling of a larger number of scales. However, this also increases the computational costs. Therefore, to make a compromise between the accuracy and computation efficiency, the final parameters selected were a grid spacing of $G = 10$, and combination of scales {6, 9, 12}. Using these parameters, the classification accuracy was 0.95, or 285 out of 300 seed images correctly classified.

Instead of just comparing vertically, I have also included the result of BVLC reference net, VGG-19, DSIFT-FV-noncat, DSIFT-FV-cat for horizontal comparison. The baseline results were shown in Figure 4.5, 4.6 as dashed line. It can be clearly seen that the proposed method surpasses these baselines. This suggests that for identification conducted in a controlled experiment, incorporating real scales can be

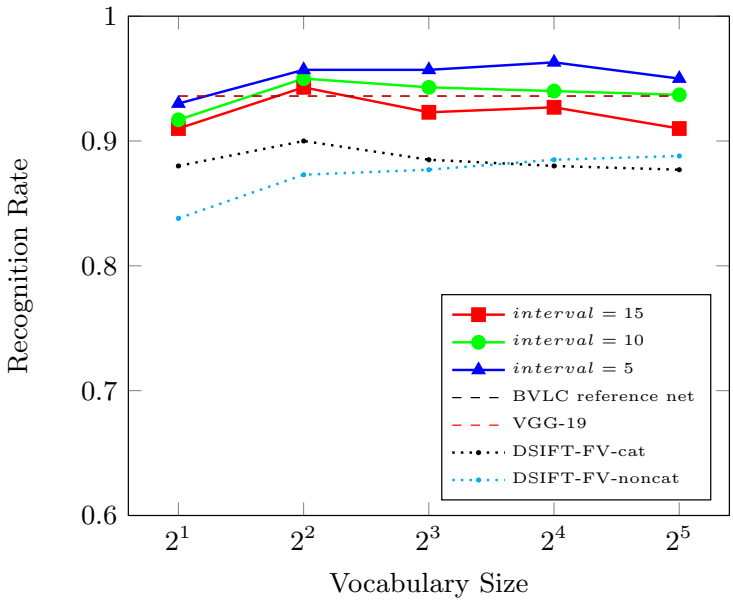


Figure 4.6: Effect of grid interval on the classification results. The set of pooled scales {6, 9, 12} was used for these results. Note that results from BVLC and VGG-19 are overlapped.

beneficial.

Comparing DSIFT-FV-noncat with DSIFT-FV-cat, one finds that even though the latter has a representation dimensionality four times the size of the non-concatenated one, the performance is worse when vocabulary size exceeds 8. This suggests that simple feature concatenation from various sizes of local regions is not working for object identification even when objects have small scale variance. However, if one can relate the size of the local region to the real world scale, feature concatenation across scale becomes beneficial because the scale-wise matching kernel compares the objects in a scale-normalized fashion.

The CNN feature is more discriminative than the traditional hand-crafted baseline features even for identifying seeds that have never been seen by the trained network. It can easily achieve a average recognition rate of 0.936 (VGG-19 and BVLC reference net got the same performance) without any parameter tweaking. However, by incorporating real scale information, the proposed method can surpass it. One of the problems with these pre-trained CNN is that the feature at the last stage is domain-specific and needs fine tuning when used with a dataset other than ILSVRC 2012. Another problem associated with CNN is that the input size of the network has a small spatial support. Therefore when resizing seed images to this small size, critical high frequency details for separation of look-alike seed might be lost. In the future when the a larger seed dataset is available, it might be of interest to fine-tune the higher-level portion of the network to further improve the CNN's performance.

Given its superior performance, it would be helpful to see what kind of information my feature representation captures. Therefore, I visualized the proposed feature representation with t-sne [115], a manifold learning approach for non-linear dimensionality reduction. The feature representations for all 300 seed images

were reduced to 2D as shown in Figure 4.7. This technique retains probabilities rather than distances between neighbouring points in the high dimensional feature space. The aim of this visualization is to show the underlying structure in the representation and see how it correlates with prior knowledge about the existing seed species (subspecies or seed types).

Generally speaking, seeds from the same groups (share similar morphological features) are more likely to stay together which suggests the effectiveness of this representation. More specifically, it can be seen that, samples from group 2A (No. 7, 8, 9, 13), 2B (No. 10, 11, 14), 3 (No. 16, 17, 18, 19, 20), 5 (No. 26, 27, 28, 29, 30) tend to form their own clusters lying far way from each other. This means that the probability of confusion between these groups is much smaller than confusion within groups. As for group 1, *B. rapa*(c), *B. rapa*(p) tend to stay much closer to each other which can be explained by the fact that they are all subspecies from the same species *Brassica rapa*. A similar phenomenon was observed for group 4 where *S. italica*(i) and *S. italica*(v) are more similar to each other than that of *S. faberi*, *S. pumila*, *S. verticilata*. One interesting finding is that *B. rapa*(y) is more likely to stay with *B. carinata* rather than its same species peers (*B. rapa*(b), *B. rapa*(c), *B. rapa*(p)). Group 5 is difficult for humans because of the lack of visible surface features. But as shown in the visualization, the feature representation is pretty consistent as compared to Group 4 which has strong texture but varies a lot.

In addition, I have selected 5 species that scatter much more widely and mapped the visualized 2D points of them back to its original images.

1. *B. rapa*(y) (5)

   There are two points that lie far away from the remaining eight points. The ten samples were shown in Figure 4.9 with the two outliers coloured with a red and blue rectangular box. The outlier of sample index 8 could result from the hilum texture. The outlier of sample index 4 could result from both the hilum and the visually distinguishable surface texture.

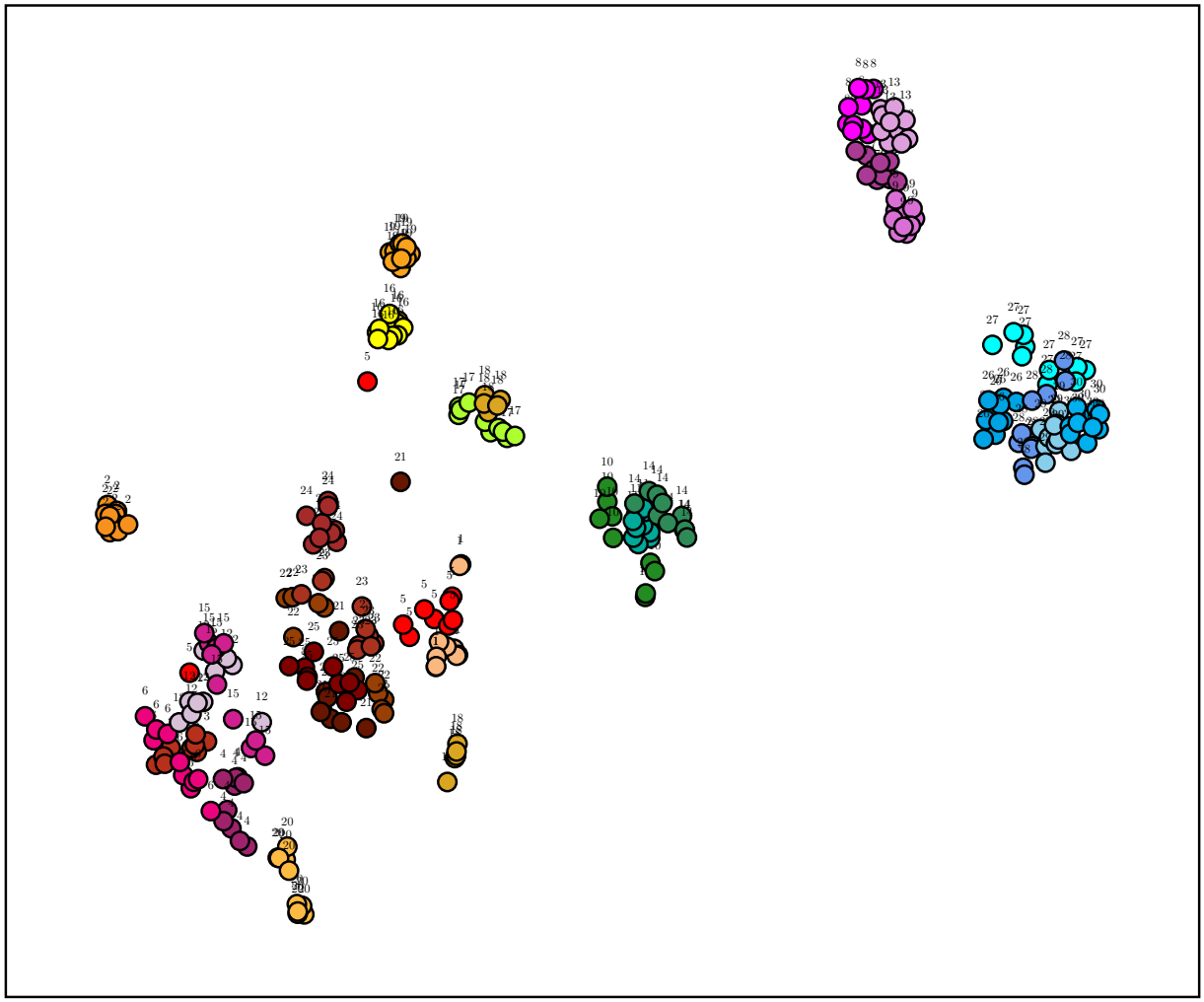
2. *S. italica*(v) (18)

   These samples are generally splitted into two parts, five of them stays with *S. italica*(i) (16); the other five form another cluster. If we look at the original images, the two sets of species were actually sampled at different times. The second batch of samples are intended to have a different sample variation. This variation is successfully captured by the proposed feature representation. Similar happens for *S. verticilata* (20) and *A. palmeri*(a) (27) as shown in Figure 4.11 and Figure 4.12.

3. *S. italica*(i) (17)

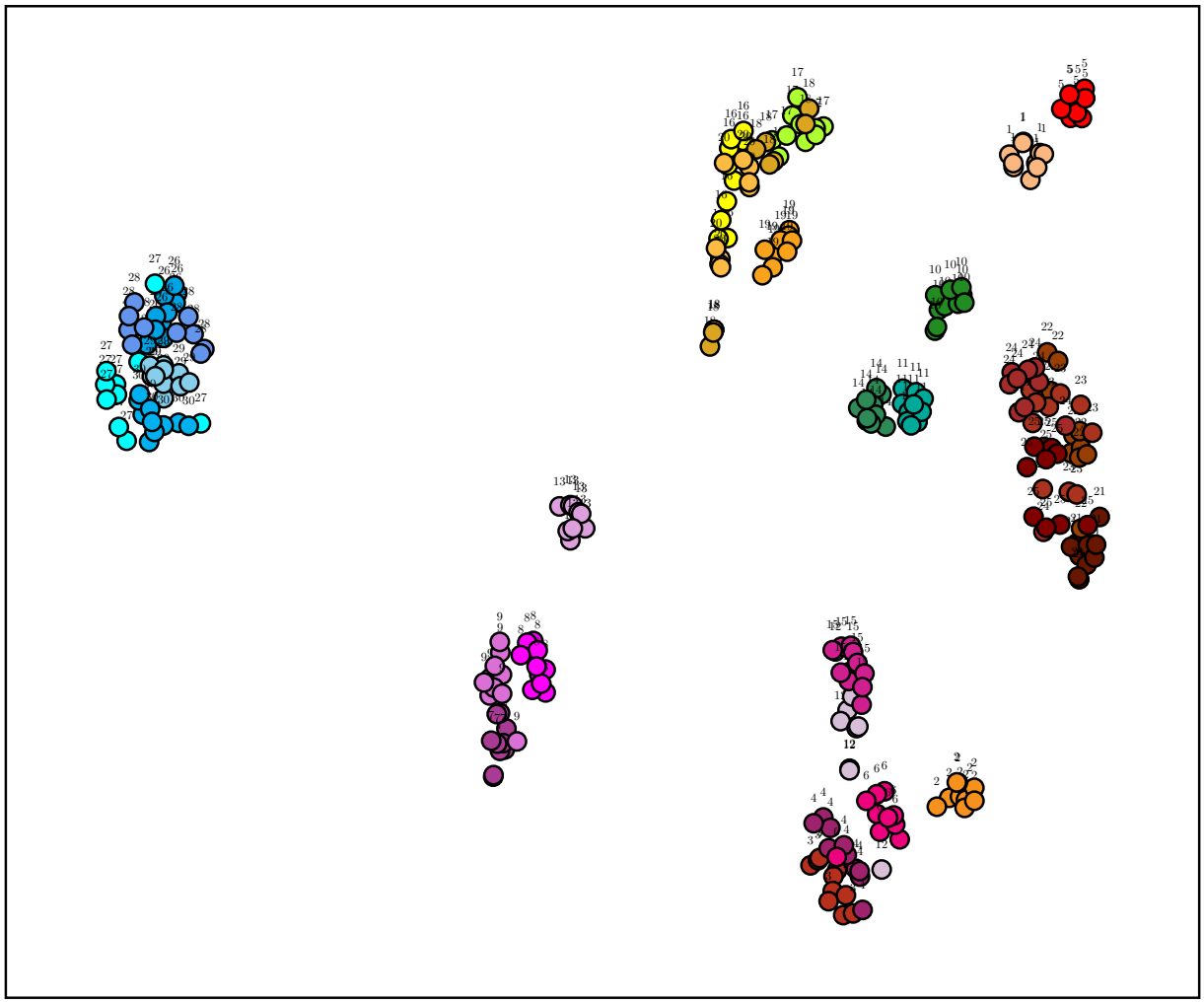
   This species also forms two clusters as shown in Figure 4.13. It can be easily seen that, the separation is mainly due to the surface reflectance.

4. *B. rapa*(c) (12) and *B. rapa*(p) (15)



B. carinata (1) *B. junceai* (2) *B. napus* (3) *S. arvensis* (4)
B. rapa(y) (5) *B. rapa*(b) (6) *B. rapa*(c) (12) *B. rapa*(p) (15)
C. diffusa (13) *C. melitensis* (7) *C. stoebe* (8) *C. solstitialis* (9)
S. loeselii (10) *C. bursa-pastoris* (11) *D. sophia* (14) *S. faberi* (16)
S. italica(i) (17) *S. italica*(v) (18) *S. pumila* (19) *S. verticilata* (20)
C. campestris (21) *C. chinensis* (22) *C. gronovii* (23) *C. megalocarpa* (24)
C. pentagona (25) *A. hybridus* (26) *A. palmeri*(a) (27) *A. palmeri*(r) (28)
A. powellii (29) *A. retroflexus* (30)

Figure 4.7: Visualization for the proposed feature. $interval = 10$, $scale = \{6, 9, 12\}$. t-sne was used to reduce the feature representation dimension to 2. This technique is able to retain the local structure of the data (neighbouring points in the high dimensional space mapped together) while also revealing some important global structure (dissimilar points got mapped far away). Readers are referred to the seed images in Figure 4.3 for better comprehension of this visualization.



B. carinata (1) *B. junceai* (2) *B. napus* (3) *S. arvensis* (4)
B. rapa(y) (5) *B. rapa*(b) (6) *B. rapa*(c) (12) *B. rapa*(p) (15)
C. diffusa (13) *C. melitensis* (7) *C. stoebe* (8) *C. solstitialis* (9)
S. loeselii (10) *C. bursa-pastoris* (11) *D. sophia* (14) *S. faberi* (16)
S. italica(i) (17) *S. italica*(v) (18) *S. pumila* (19) *S. verticilata* (20)
C. campestris (21) *C. chinensis* (22) *C. gronovii* (23) *C. megalocarpa* (24)
C. pentagona (25) *A. hybridus* (26) *A. palmeri*(a) (27) *A. palmeri*(r) (28)
A. powellii (29) *A. retroflexus* (30)

Figure 4.8: Visualization for the VGG-19 feature.

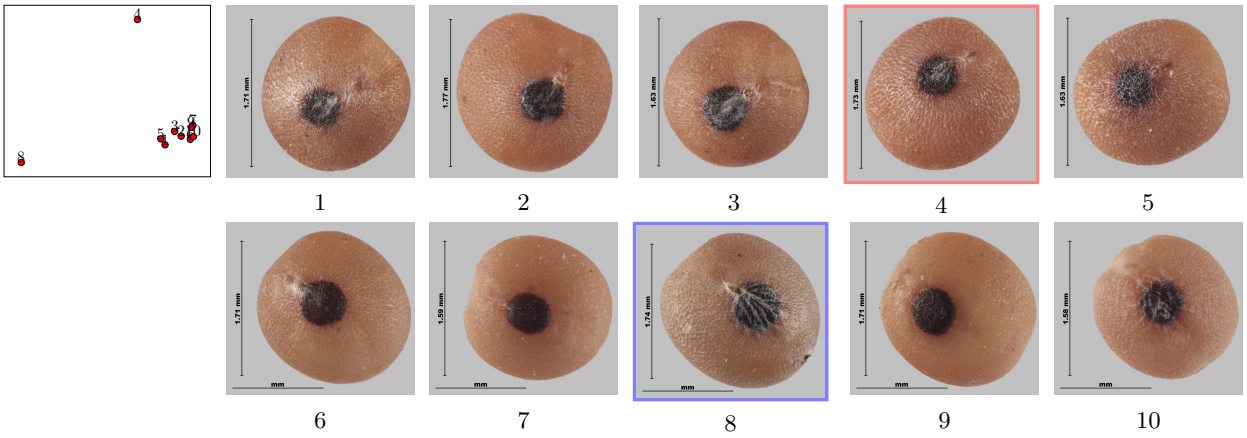


Figure 4.9: The original images of *B. rapa*(y) (5). Number below each image shows the seed sample index. Number also marked up for each point in the top left visualization. Distinctive images are outlined with colour boundary for easy read.

Figure 4.10: The original images of *S. italica*(v) (18). Number below each image shows the seed sample index. Number also marked up for each point in the top left visualization. Distinctive images are outlined with colour boundary for easy read.

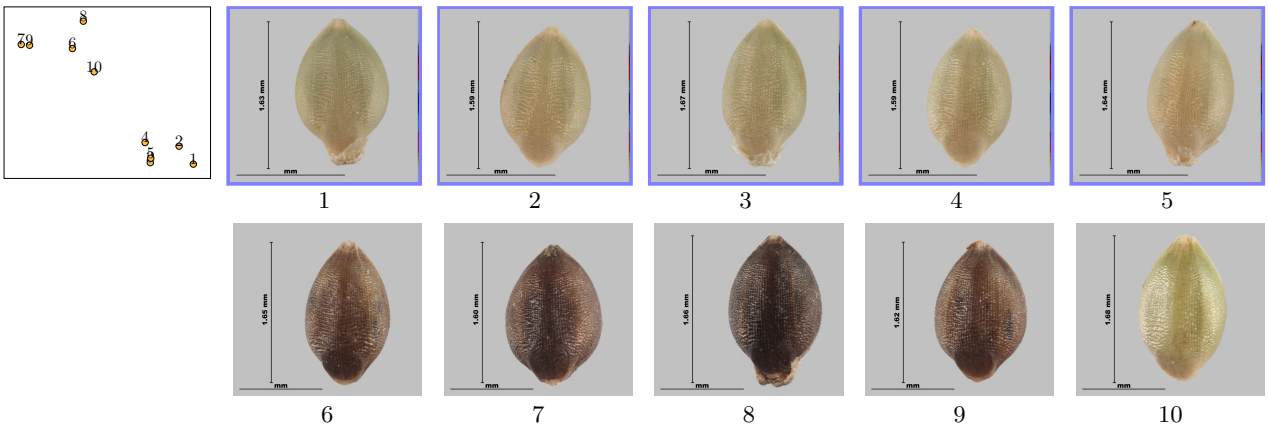


Figure 4.11: The original images of *S. verticilata* (20). Number below each image shows the seed sample index. Number also marked up for each point in the top left visualization. Distinctive images are outlined with colour boundary for easy read.

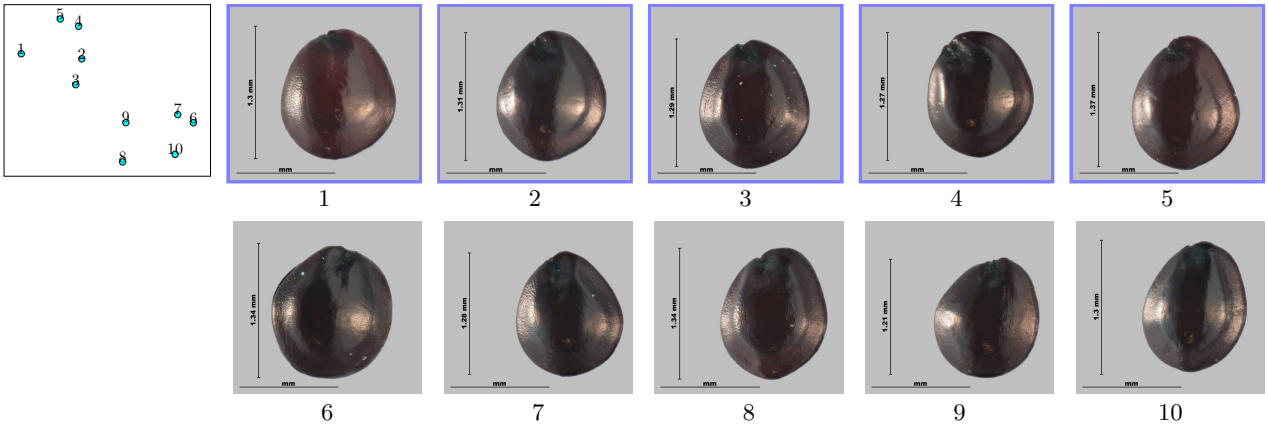


Figure 4.12: The original images of *A. palmeri*(a) (27). Number below each image shows the seed sample index. Number also marked up for each point in the top left visualization. Distinctive images are outlined with colour boundary for easy read.

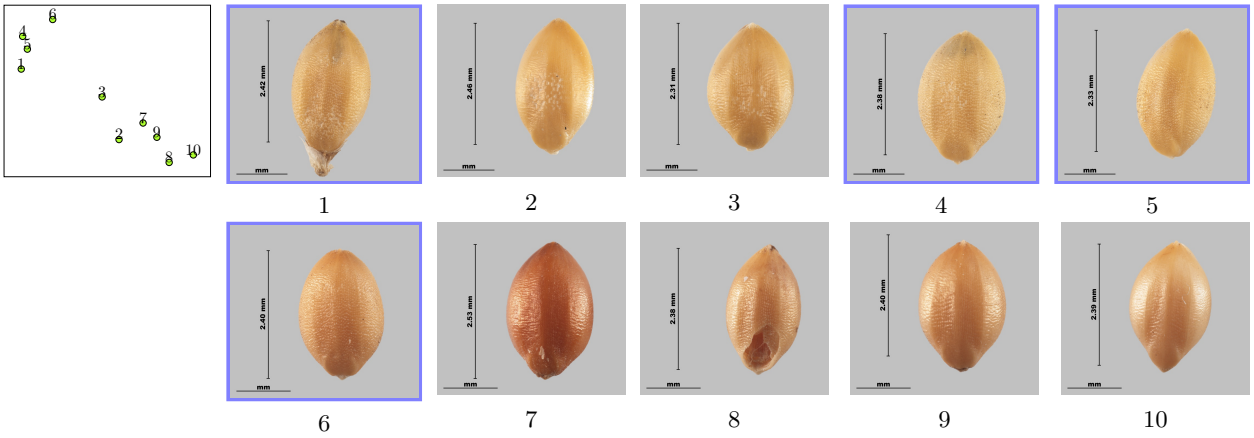


Figure 4.13: The original images of *S. italica*(i) (17). Number below each image shows the seed sample index. Number also marked up for each point in the top left visualization. Distinctive images are outlined with colour boundary for easy read.

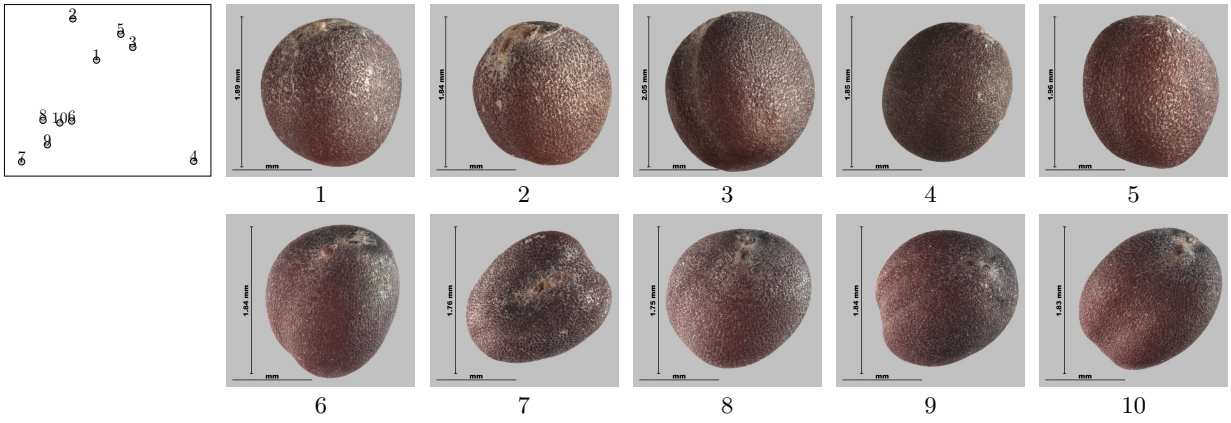


Figure 4.14: The original images of *B. rapa*(c) (12). Number below each image shows the seed sample index. Number also marked up for each point in the top left visualization.

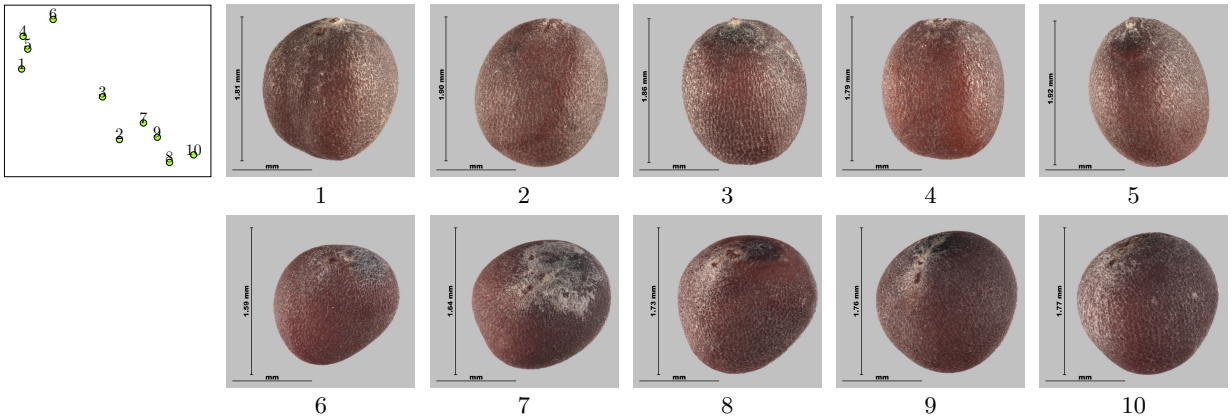


Figure 4.15: The original images of *B. rapa*(p) (15). Number below each image shows the seed sample index. Number also marked up for each point in the top left visualization. Distinctive images are outlined with colour boundary for easy read.

For this two subspecies, the scatter of the sample points is mainly due to changes in viewpoint. As shown in Figure 4.14 and Figure 4.15, not all samples has the hilum shown in the image. This implicitly informs us that the hilum actually is a critical feature for identification.

Visualization of the VGG-19 representation is shown in Figure 4.8. If comparing it with the visualization of the proposed method, you can notice that VGG-19 is good at differentiating round seeds that could have various poses, e.g. seeds that are in group 1 and 4. In contrast, the proposed method is good at picking up the subtle differences for the other flat seed groups that could have much less variation in pose.

4.5 Conclusion

In this chapter, I have proposed a mid-level feature representation using scale-wise pooling. It normalizes the local image patches with physical pixel size and can be treated as an extension of the commonly adopted

pyramid matching technique. I have proven with experiments that utilizing information from real scales can lead to improvement in the identification rate achieved on the seed dataset. The accuracy achieved (95%) with only texture features is higher than the threshold ($> 90\%$) that is expected from a trained seed analyst [154].

This feature representation is suitable for image datasets that have limited scale changes. Otherwise the scales selected as demonstrated in Figure 4.4 would not be appropriate for all the object classes. For example, if some seeds were imaged very small, the number of grid points would decrease accordingly. The size of the support region of the local descriptors would also shrink in this case. In extreme cases, the support region size would be a single pixels and no texture information could be extracted.

The successful application of this proposed method would require the seed in the image to have similar size and sharpness as the one in the training set. Any blurriness introduced would violate the underlying assumption that images are of limited scale changes. The reason for this is that blur attenuates high frequency information. Thus it alters the information local descriptor extracts. In the next chapter, I have incorporated both the techniques proposed in chapter 3 and 4 into a seed identification tool. We will see identification performance degradation as resulted by blur in the acquired image.

Chapter 5

User Study

This chapter describes the evaluation of the practical usage our seed identification system by human specialists in seed identification. A digital tool was built for seed identification based on the real-time focus-stacking method described in Chapter 3 and the scale-pooling representation from Chapter 4. Our system was tested by professional seed analysts working in actual laboratory conditions at the CFIA seed testing laboratory in Saskatoon using specimens from the Canadian National Seed Herbarium. Currently, seed analysts in this lab must identify large numbers of plant seeds on a daily basis, manually, with limited assistive tools. It would be beneficial for them if our seed identification system can accomplish the task in an accurate and proficient manner. The design of the user experiment was to investigate two conditions: 1) current practice of seed identification and 2) computer assisted identification with our tool, and whether there are observable significant differences between workload, average time per sample, and recognition rate.

5.1 Experiment Setup

In this experiment, we recruited experienced seed analysts from CFIA because they are the target users of the built identification system. There are around 100 professional seed analysts in total across the country of Canada and we manage to recruit six of them directly from the Saskatoon laboratory of CFIA. These participants were divided into 3 groups based on the level of expertise with level 1 corresponding to the novice group (1-2 years), level 2 corresponding to intermediate proficiency (2-10 years) and level 3 corresponding to expert group (more than 10 years of experience).

5.1.1 Conditions

The objective of this study is to evaluate whether the provided identification tool can be beneficial for the seed identification. The two conditions involved are straightforward and are described below:

1. **Computer-assisted**

   In this condition, participants are provided with the digital aiding tool that is designed to automatically identify plant seeds (a detailed description of this tool can be found in Section 5.2.1). The tool reports what it thinks are the three most likely species for a given sample in descending order of its confidence in each such potential decision.

2. **Manual**

   This condition resembles the current workflow of seed identification task carried out in the seed laboratory. Usually professionals are equipped with an optical microscope and some reference books. Samples in the seed herbarium can be resorted to when needed.

   In practice, seed samples are classified mainly based on morphological features and their similarities. The identification involves comparison of certain characteristics and then assigning a particular seed to a known taxonomic group, ultimately arriving at a species [161]. Knowledge of seed structures is critical to achieving an accurate determination of unknown samples. Sometimes creating table of characteristics, including as many morphological features as possible, is desirable, such that all available features can be examined thoroughly [3]. Table 5.1 shows morphological characters of five *Trifolium* species which is illustrated in seed technologist training manual [3].

Species	T. *fragiferum* L.	T.*hybridum* L.	T. *pratense* L.	T. *repens* L.	T. *vesiculosum* Savi
General Shape	broadly ovate	oval to heart-shaped	triangular to mitten-shaped	oval to heart-shaped	round to oval
Radicle Compared to Cotyledon Lobe	equal to or longer	equal or slightly shorter	>1/2 the length of the cotyledon lobe	equal to or slightly shorter	equal to or longer
Radicle Divergent from Cotyledon Lobe	no	yes	yes	yes	slightly
Surface Texture	smooth	smooth	smooth	smooth	tuberculate
Color	yellow to terracotta with dark motting	yellow to green with purple, blue-green, or black motting	yellow with red and purple tinge to entirely purple	yellow to terra cotta some with green tinge	terra cotta to red
Luster	lustrous	dull	dull	dull to lustrous	dull

Table 5.1: Comparision of seed characters of five *Trifolium* species.

5.1.2 Dependent Variables

Three dependent variables are measured in this study:

1. **Workload**

   Participants undertook the NASA Task Load Index (NASA-TLX) measure which consists of a set of 6 scales and 15 pairwise comparisons for subjective workload measurement. These 6 scales are mental demand (w_{ment}), physical demand (w_{phyc}), temporal demand (w_{temp}), performance (w_{perf}), effort

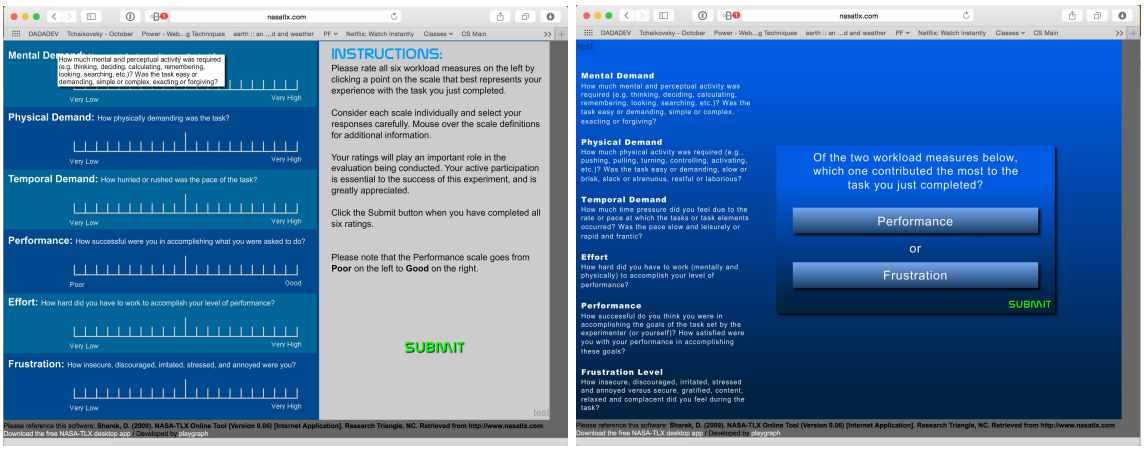


Figure 5.1: Example of NASA Task Load Index measure. It measures the subjective workload in 6 scales and using 15 pairwise comparisons between each two to assign weights for each scales for normalization. In practice, I used the paper version for easy management of the data.

(w_{eff}), frustration (w_{frst}). The overall task load index is a weighted score range from 0 to 100, with higher numbers indicating higher workload. A screen shot is provided in Figure 5.1. The workload under two conditions are denoted as w_1, w_2.

2. **Time per Sample**

   It is defined as the time that consumed for processing each test sample. The mean time per sample under the two conditions described in Section 5.1.1 are denoted as μ_{t_1}, μ_{t_2}.

3. **Recognition Rate**

   Recognition rate is defined as the percentage of samples that are correctly identified to the species level (subspecies or seed types if applicable). One measure of accuracy per participant per condition was computed. The mean accuracy under two conditions are denoted as μ_{a_1}, μ_{a_2}. For the computer assisted condition, I used top-3 recognition rate as the measurement. The test sample is treated as successfully identified as long as the correct species name is among the top-3 candidates recommended by the algorithm.

Competing with professionals in the current setup poses a big challenge to the proposed identification system since the computer is generally considered as inferior at the high-level vision tasks than human beings [21], despite the ongoing efforts trying to bridge the gap between each other.

5.1.3 Hypothesis

In this study, the participants in both groups are the same which makes the samples paired. Also, because of the number of limited samples, the underlying distribution can not be treated as normal distribution. Therefore, Wilcoxon signed-rank test was adopted for the statistical test to compare the matched samples to

assess whether there is any improvement of using the provided identification tool. Three pairwise comparisons were carried out among these two conditions.

The hypotheses to be tested are (null hypothesis is denoted as H_0 and alternative hypothesis is denoted as H_1):

1. **Hypotheses for Workload**

   (a) H_0: difference between the workload under condition 1 and 2 follows a symmetric distribution around zero. In mathematically form, they can be expressed as $median(w_1 - w_2) = 0$.

   (b) H_1: the median of the difference between the mean workload under condition 1 and 2 is less than zero. In mathematically form, they can be expressed as $median(w_1 - w_2) < 0$.

   Rather than only comparing the aggregated score, workload in each individual dimension of the NASA TLX questionnaries was also compared. The alternative hypotheses for them are $median(w_{ment1} - w_{ment2}) < 0$, $median(w_{phyc1} - w_{phyc2}) < 0$, $median(w_{temp1} - w_{temp2}) < 0$, $median(w_{pref1} - w_{pref2}) < 0$, $median(w_{eff1} - w_{eff2}) < 0$, $median(w_{frst1} - w_{frst2}) < 0$

2. **Hypotheses for Time per sample**

   (a) H_0: difference between the time per sample under condition 1 and 2 follows a symmetric distribution around zero. In mathematically form, they can be expressed as $median(\mu_{t_1} - \mu_{t_2}) = 0$.

   (b) H_1: the median of the difference between the mean time under condition 1 and 2 is less than zero. In mathematically form, they can be expressed as $median(\mu_{t_1} - \mu_{t_2}) < 0$.

3. **Hypotheses for Recognition Rate**

   (a) H_0: difference between the mean recognition rate under condition 1 and 2 follows a symmetric distribution around zero. In mathematically form, they can be expressed as $median(\mu_{rr_1} - \mu_{rr_2}) = 0$.

   (b) H_1: the median of the difference between the mean recognition rate under condition 1 and 2 is larger than zero. In mathematically form, they can be expressed as $median(\mu_{rr_1} - \mu_{rr_2}) > 0$.

5.2 Overview of the Identification System

5.2.1 Software

The interface of the seed identification tool was implemented using Qt 5.5.1 with C++ and a screen shot of it is provided in Figure 5.2. This tool can be operated with two modes: a static image mode which suitable for external image sources; or a live image mode which directly obtains from the image sensor.

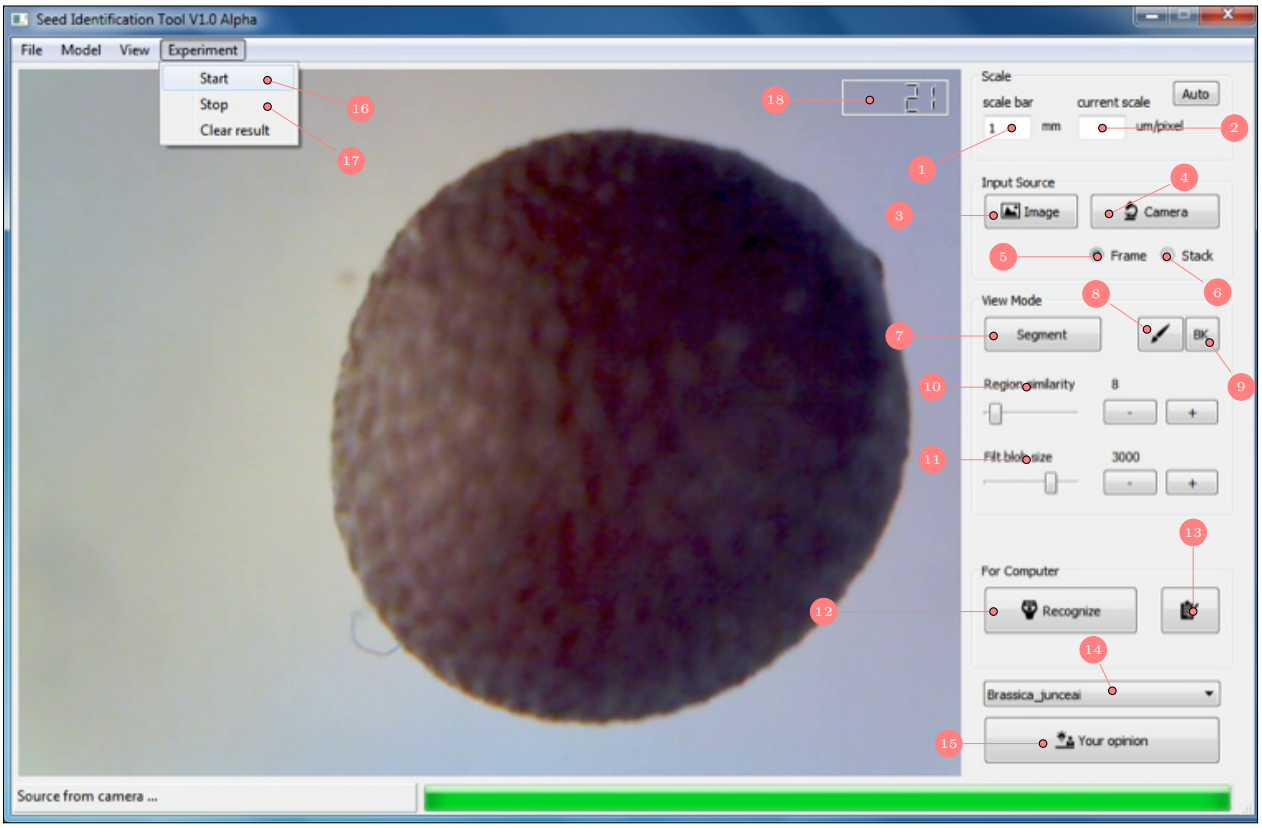


Figure 5.2: Overview of the seed identification tool with each functionality highlighted by numbers. Detailed descriptions can be found below and the usage instruction can be found in section 5.2.1.

1. Length of the scale bar.

   It is set to be 1mm by default as we used a stage micrometer in the experiment. The user is expected to calibrate the system in the very beginning of each test.

2. Current scale.

   The size of each pixel in the image/video frame, measured in µm.

3. Input source: image.

   This mode loads any static image from the local hard drive.

4. Input source: camera.

   live image from the sensor (default).

5. Frame mode.

   Only working with source from camera. Only the current live image frame will be processed.

6. Stack mode.

   Only working with source from camera. Live images aggregated to create focal stack and the user is expected to adjust the focus knob during this mode. The proposed sharpness metric in chapter 3 is employed here for the stacking.

7. Segment.

   Isolate the test seed from its surrounding background. Region growing is employed for the segmentation with the initial seeds chosen as the pixels on the image boundary.

8. Brush.

   Used to repaint the segmented image if automatic segmentation gives unsatisfactory result.

9. BK/FG.

   Toggled to change the brush stroke type used in repainting. BK: background stroke subtracting the background. FG: foreground stroke bringing back erroneous subtracted foreground.

10. Region similarity.

    Range of variation of the background pixels. Increase this value to cope with complex background resulted by unexpected lighting changes.

11. Filter blob size.

    Filter out scattered unconnected background regions.

12. Recognize.

    Let the tool make the decision of which species the tested sample belongs to. Only works when image system is calibrated and background is subtracted.

13. Record the tool's identification result.

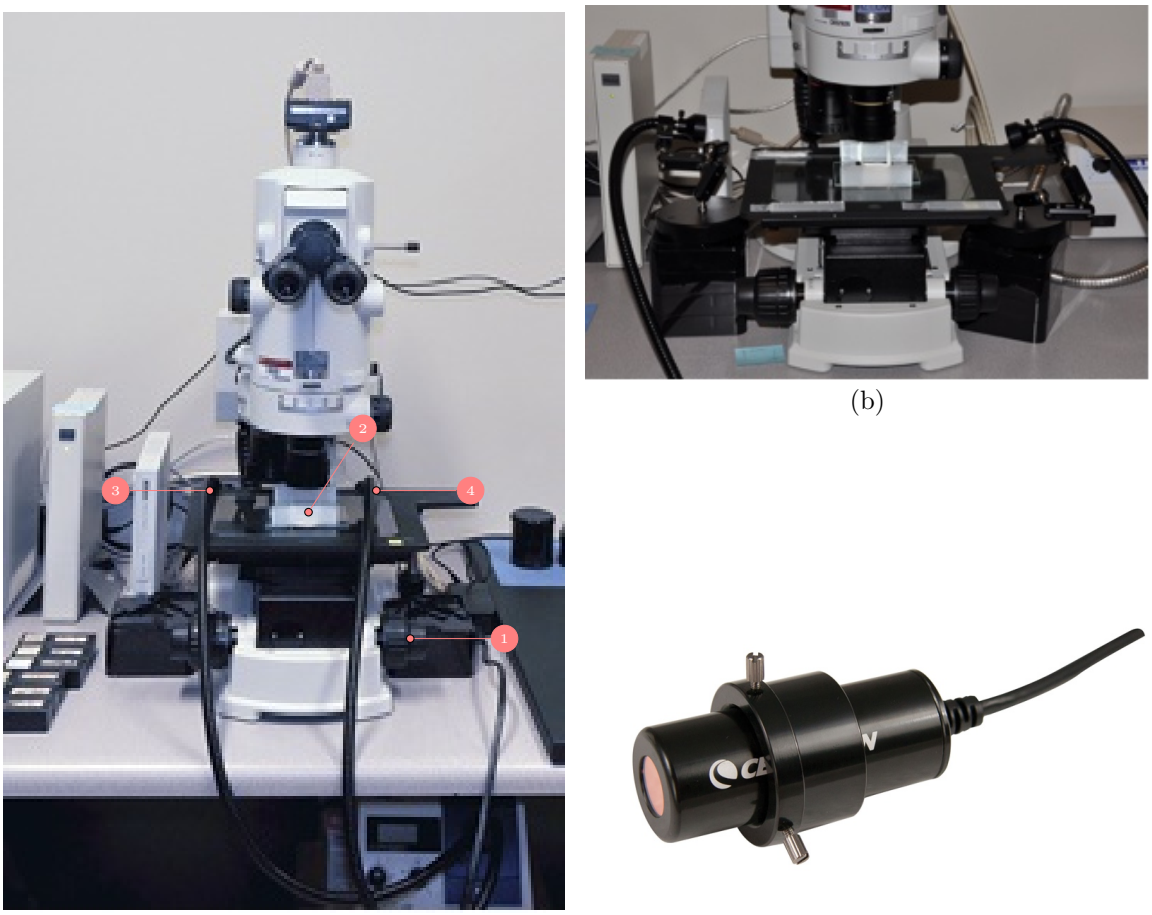
14. List of seed species that can be identified by the identification tool.

15. Record participant's identification result. Should be always used with 14.

16. Click to start the experiment.

17. Click to stop the experiment.

18. LCD indicator used to show the number of samples user has already processed.



(a) (b) (c)

Figure 5.3: An overview of the hardware system. (a) is the NIKON stereoscopic microscope. (b) shows a close up view of the stage. The fine-tuning knob (1) is used to adjust the focal point. A diffusor (2) is placed on the stage to get even lighting and two external light sources (3 4) are positioned on both sides pointing in oppsoit direction. (c) is the digital eyepiece where the live image frame is coming from.

5.2.2 Hardware

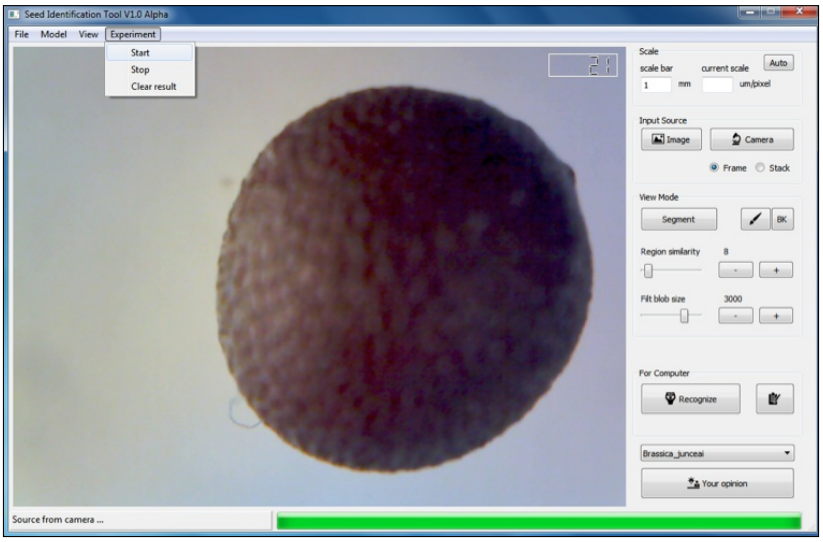
In this study, the same microscope used in chapter 4 is employed for the seed imaging. However, the live image frames are acquired from a digital eye piece manufactured by Celestron instead of from its internal image sensor. The reason is that this NIKON microscope (AZ100M Motorised Multi-Purpose Zoom Microscope) comes with its own proprietary imaging software (NIS-elements) and does not provide an SDK for easy customization to third party developers. Thus for the concern of easy customization and future improvement, the digital eye piece is adopted to act as a bridge between the microscope and the software. An overview of the hardware system is given in Figure 5.3.

5.2.3 Operation Pipeline

The operation of this whole system for identification of seed samples can be divided into this four steps:

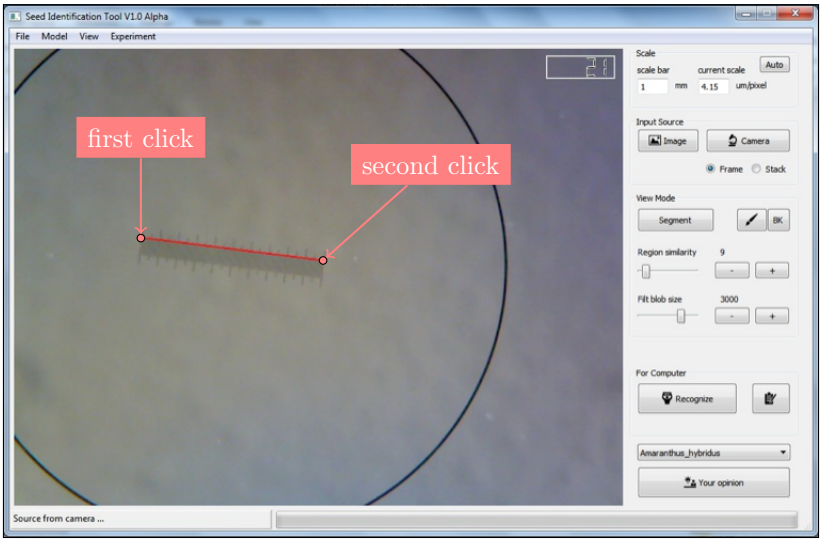
Step 1:

Put the test seed sample on the stage micrometer and adjust the zoom and focus knob to get clear view of the seed. Make sure the seed lies in the centre of the view finder and does not touch the boundary. ***Also make the seed sample as large as possible so that the textures on the seed surface can be clearly rendered.*** Once a satisfactory view of the seed is obtained, keep the zoom knob untouched until the next sample.



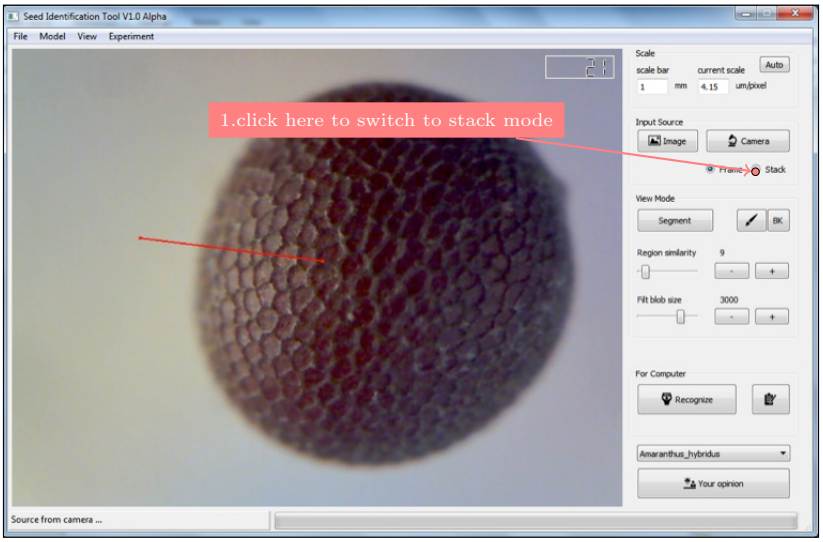
Step 2:

Slide the stage micrometer over and adjust the focus knob to get a clear view of the micrometer. The imaging system is calibrated by left clicking on both ends of the micrometer as shown in the above image.



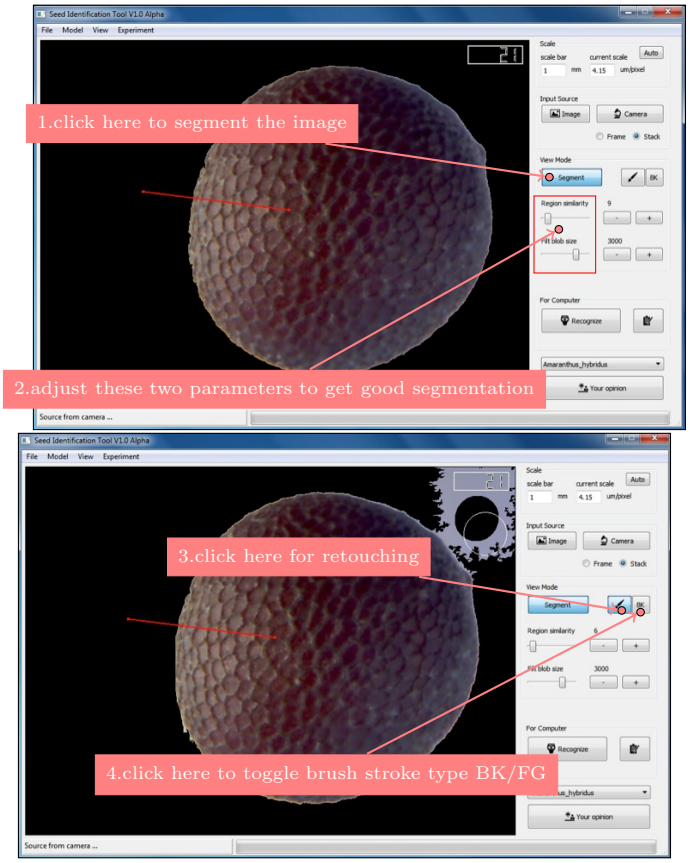
Step 3:

Slide the stage micrometer back to the seed sample. Switch to stack mode to create focal stack. The seed sample in the viewfinder should become sharper gradually. When there is visually no change on the sample's apperance, proceed to the next step to segment the seed out from the white background.



Step 4:

Segment the image by clicking the segment button. Better results can be achieved by adjusting the two slider marked in the top image. If automatic segmentation does not produce satisfactory result, brush can be used for manual refinement as shown in the below image. Since this retouch is not for artistic purpose, a coarse segmentation is sufficient for the recognition.



5.3 Experiment results

5.3.1 Results for workload

Figure 5.4 shows the aggregated TLX score for both conditions. For the computer assisted condition, the workload generally decreases with the increase of level of expertise. This is contrary to our intuition since one might think the novice participants would get the lowest workload score because they might be the ones that benefit most from this tool. For the traditional method, the greatest workload was observed for participants with the middle level of expertise. The most reasonable interpretation of this is that, they are actually the one that spend the most effort on this task given its correlation to the trend of the average time per sample as shown in Figure 5.6. Novice analysts might guess whenever come across hard samples whereas senior analysts can fully rely their own experience without going for the references. The mid-level analysts actually need more time to confirm their choice with the references. Although on average $\mu_{w1} < \mu_{w2}$, there is no statistical significant difference between these two group of samples by Wilcoxon sign ranked t test. The null hypothesis cannot be rejected. This could be owing to the insufficient number of participants.

In addition to the aggregated TLX score, here I also plot the score individually for each dimension after normalization with its corresponding weights, as shown in Figure 5.5. A statistical test for each individual dimension was conducted. The mental demand for traditional method is statistical significant higher than computer assisted method with $p = 0.00245$. This is due to the fact that, with the software, the operator is not required to memorize all the features or rules used for identification. The physical demand on average is higher as for computer assisted method although with no statistical significance. This is not surprising if you consider how humans identify objects. Some seeds are inherently easy for human to identify with naked eye. However, in order to have the computer analyze the sample, it has to be always taken out from the vial, and placed on the glass slide at the right spot. Because the microscope used have very large magnification, it sometimes hard to position the seed in the field of view, and to set the best magnification. Even if the participants are proficient enough with operating the seed samples, the method requires scale recalibration whenever the objective distance is changed. This can be mitigated if I can have access to the control module of the hardware but with current setup and implementation, it has to be done manually and is tedious from my observation. Afterwards, the image needs to be segmented which also requires some effort from the user. Another solution to avoid the repetitive scale calibration is to preprocess the test seed sample with a seed sorter to ensure that the seed samples identified in contiguous are homogeneous in size.

5.3.2 Results for Average Time per Sample

As for the throughput results shown in Figure 5.6, it can be seen that for the computer-assisted condition, the average time spent is fairly consistent across the levels of expertise. A possible explanation for this is that the majority of time in the computer-assisted condition is spent acquiring the image, which is an independent

skill from expertise in seed identification. For the traditional condition, the level of expertise correlates with the workload and effort measure which suggests that these group of analysts need more time to confirm their results. Even though there is no statistical evidence to reject the null hypothesis due to the insufficient number of participants, it is still worth highlighting that there is potential to increase the throughput by as much as 100% with sufficient advances in the imaging technology and the image acquisition procedure coupled with my feature representation and enough data to produce a well-trained classifier.

5.3.3 Results for Recognition Rate

The recognition rates shown in Figure 5.7 are around 50%, which is much lower than those reported in chapter 3 which is 95% in the cross validation. If compared with the traditional method, the recognition rate is even lower with the seed identification system. Before I start to investigate the underlying causes of the modest performance, one has to be noted that the manual identification was performed by the experienced seed analysts. Unlike ordinary people, they are well-trained specialists and are highly proficient at their job, which makes the baseline of the comparison much higher than just recruiting people from the general population.

Bearing this in mind, the underlying reason I think is twofold. First, the deployed model was trained on the high quality data and the hyperparameters used for training were selected based on the cross-validation on the same dataset, therefore, the performance degradation could be partially resulted by the overfitting. Second, the distribution differences of the training and testing data could also affect the identification performance. The common practice for training and testing scheme is that one splits the obtained data into training, validation and testing sets. The hyperparameters of the model are selected based on the validation error and the final performance of the model is measured by the testing error. In small-scale datasets where it is impractical to obtain a separate validation sets, people use cross validation to leverage the problem. The key part in this scheme is that data in these sets must come from the same data manifold. However, in my case, the testing data and the training data exhibit a very different distribution as visualized in Figure 5.12 and 5.13. The testing data as denoted by the triangles do not lie close to the cluster centres formed by the training data, which implies that images of the training and testing dataset actually come from different data manifold. The test data distribution is altered by the following reasons:

1. **Varying Illumination**

   There are two external light sources that are used to light the surface of the seed, pointing in different directions as can be seen from Figure 5.3 (b). A diffuser is used to distribute the light evenly on the seed surface to prevent harsh light and dead spots. However, since the position of the seed can vary a lot and the participants do not have much experience to adjust the lighting in an optimal way, the obtained images sometime have either a saturated colour or a shadow, both of which obscure surface texture. Some examples are shown in Figure 5.8.

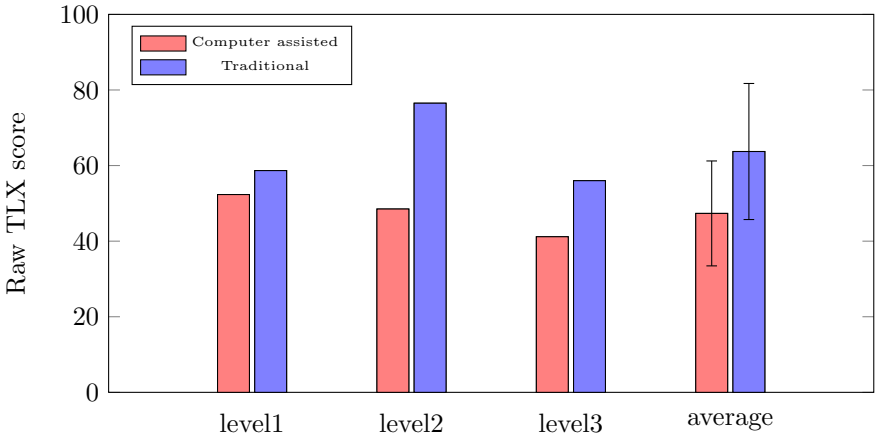


Figure 5.4: Raw TLX score for each level of expertise (the lower the score, the higher the workload). Red is the workload pretending to use everyday, whereas blue is the workload for manual traditional method.

2. **Shifting in the Image Plane**

   When rotating the fine adjusting knob, as shown in 5.3, towards the end, the seed in the viewfinder gets slightly shifted due to an artifact of the microscope. Since the image frames are not fully aligned, the obtained all-in-focus image appear motion blurred as shown in Figure 5.9. This happens mostly when the knob is rotated to the end position.

3. **Operational Errors**

   In the focal stacking phase, the focus range should cover the entire visible part of the seed ranging from the top to the peripheral. This is achieved by rotating the coarse and fine adjusting focus knobs. Ideally, the coarse knob is initially adjusted to focusing on the centre of the seed sample and the fine knob for subtle focus tweaking. However, in practice, this was sometimes not conducted correctly such that when rotating the fine adjust knob, the focus point did not cover certain parts of the sample and sometimes was not moved at all. This often resulted in a partially blurred image as can be seen in Figure 5.10.

   This error can be partially resolved by extending the participant training phase and can be completely mitigated with the right hardware. If I could have software-level access to the control system of the microscope, the manual focus stacking would be eliminated. The focus-stacking procedure could be completely automated, and done in real-time.

5.4 Discussion

The training images I have are of very high quality. Although for each species (subspecies, seed types), there are 10 samples carefully selected to cover the biological variation, these images do not contain image

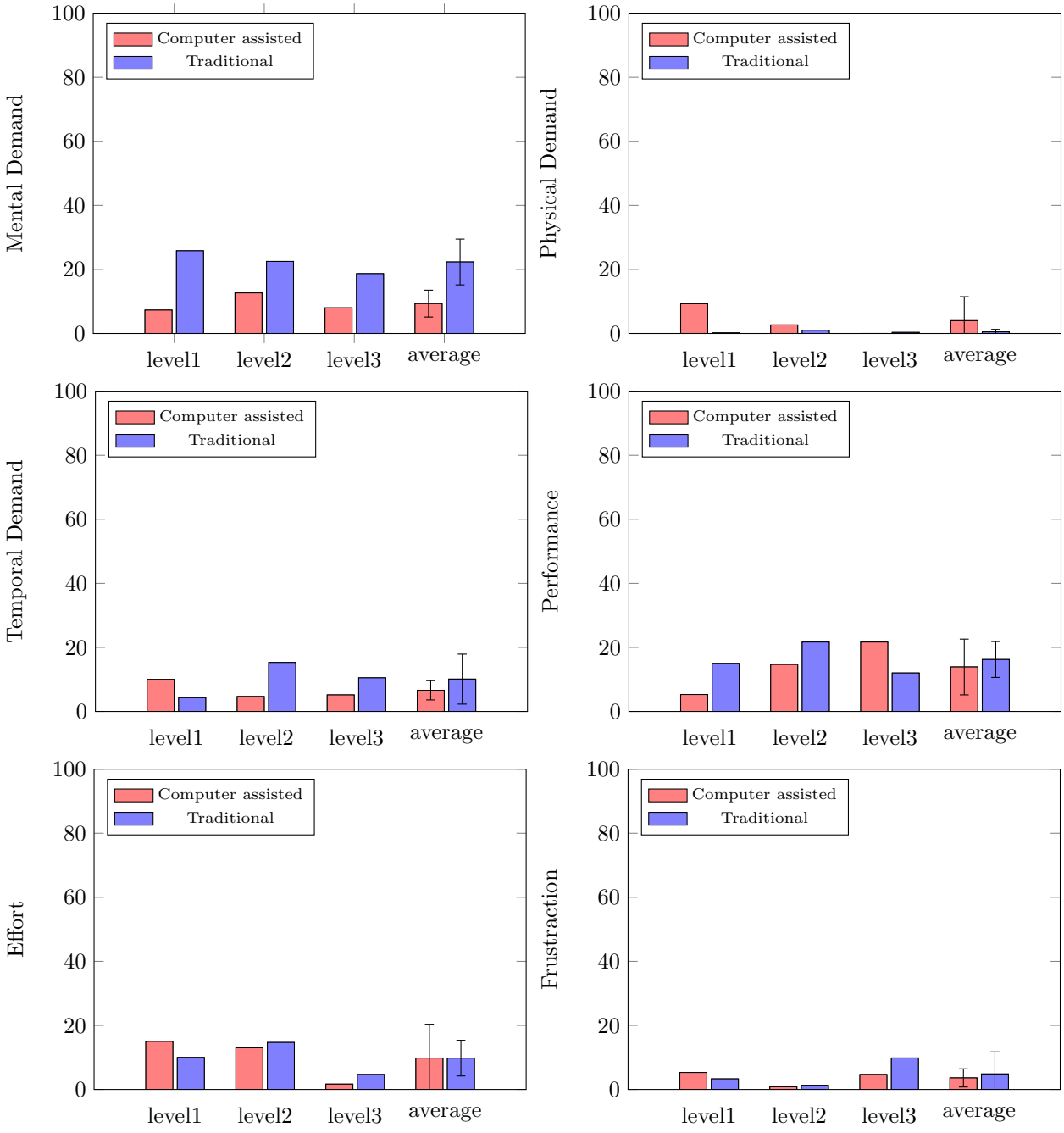


Figure 5.5: Raw TLX score of each dimension after normalization for each level of expertise. Red is the workload for the computer assisted method, whereas blue is the workload for manual traditional method.

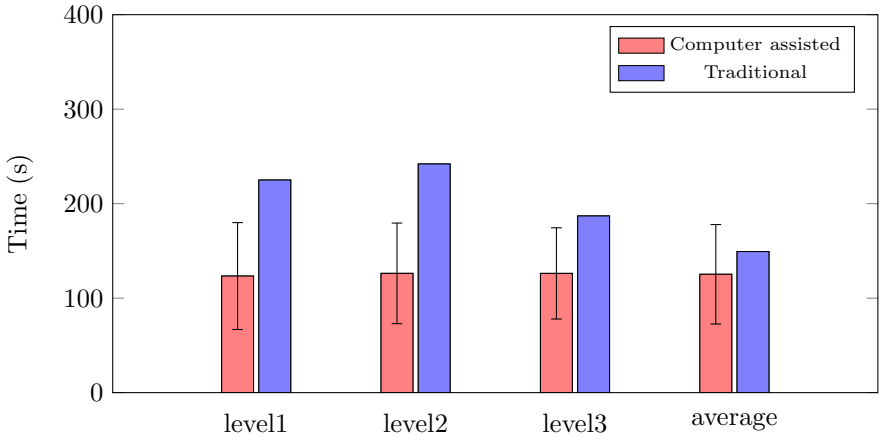


Figure 5.6: Average time spent per sample for each level of expertise.

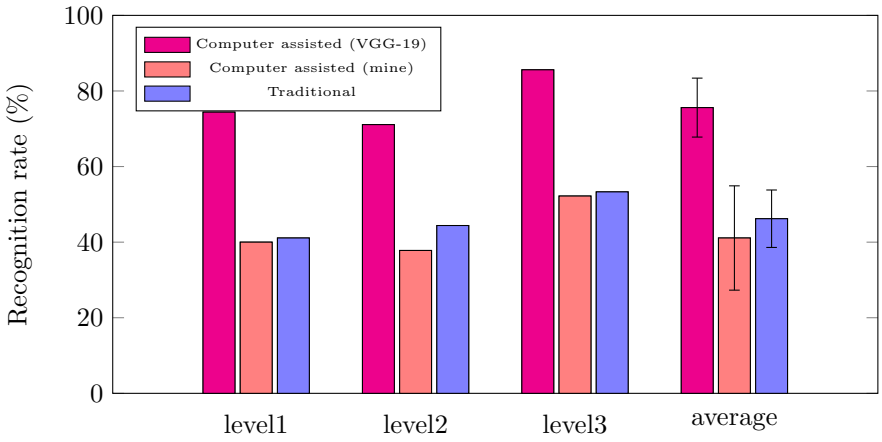


Figure 5.7: The percentage of seed samples correctly identified for level of expertise. For computer assisted and VGG-19, top-3 recognition rate was shown. The seed sample is considered as correctly identified as long as the correct result is among the top-3 candidates shown on the screen.

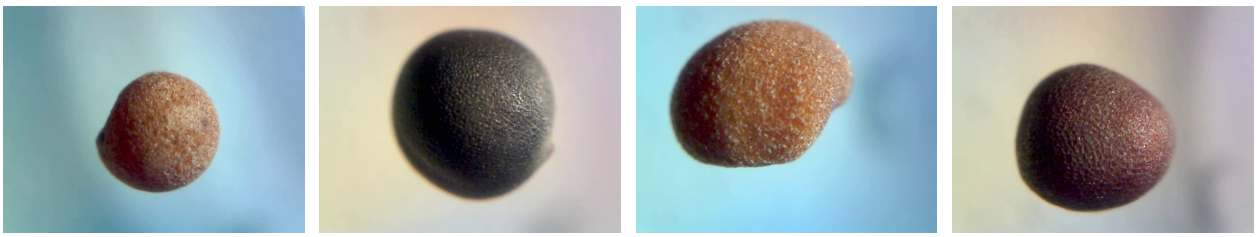


Figure 5.8: Examples of bad illumination. It leaves strong shadows on the seed surface which prevents the correct rendering of the surface texture.

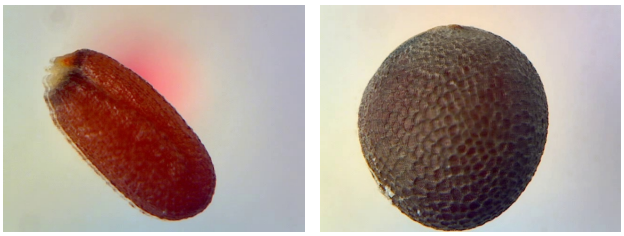


Figure 5.9: Examples of plane shifting where image frames not fully aligned with each other. Here shows the stacked image.

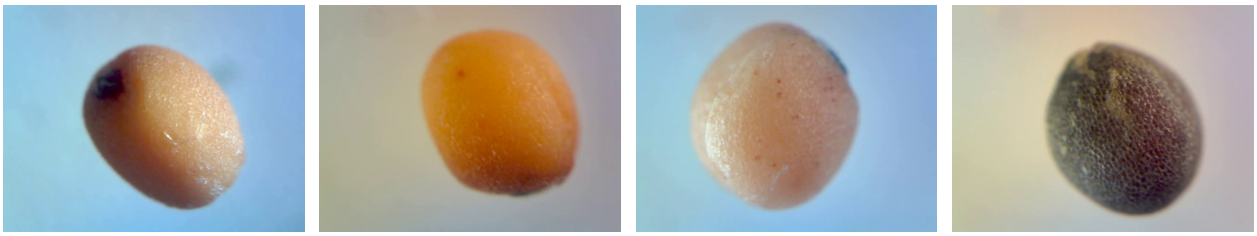


Figure 5.10: Examples of operation errors. The stacked images are still blurry due to failing change of the focus.

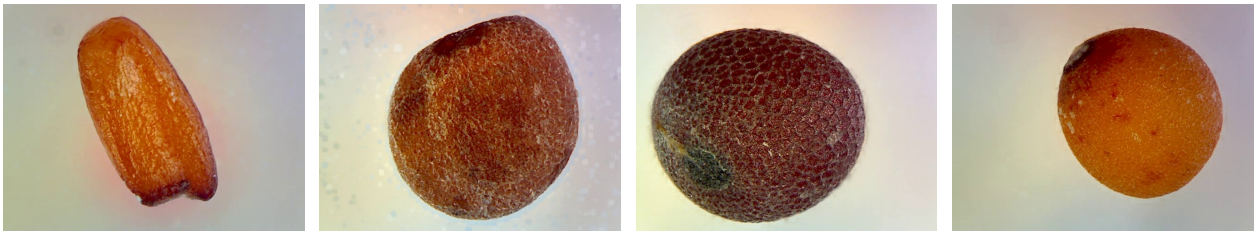


Figure 5.11: Examples of correctly identified sample.

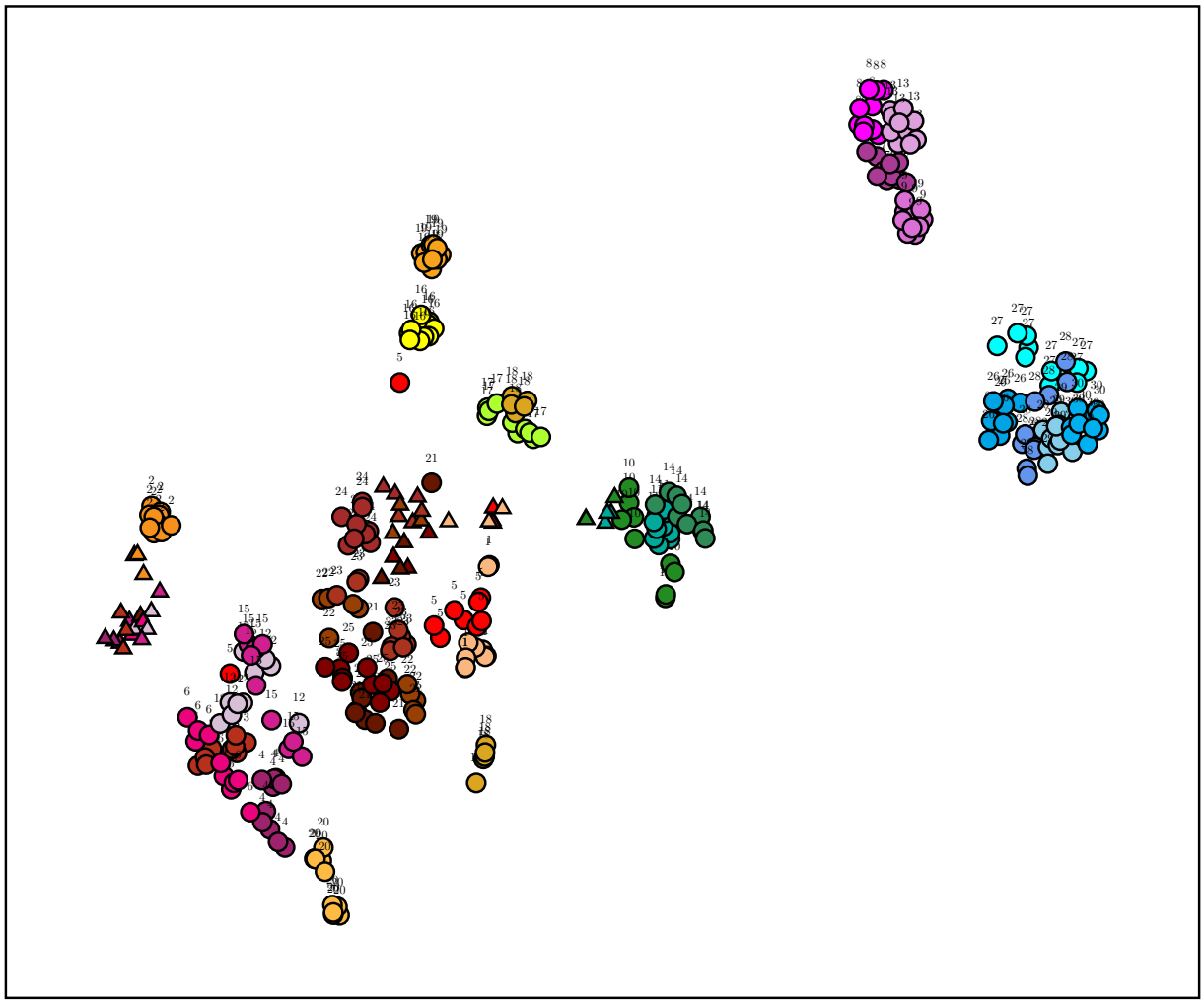


Figure 5.12: The same visualization as in chapter 3. Difference is that in this graph the test sample and the training sample are plotted altogether. The test samples come from participant 6 because its best performance (performance of each participant is shown in the appendix.) among the others and are denoted as triangles rather than circles for distinction.

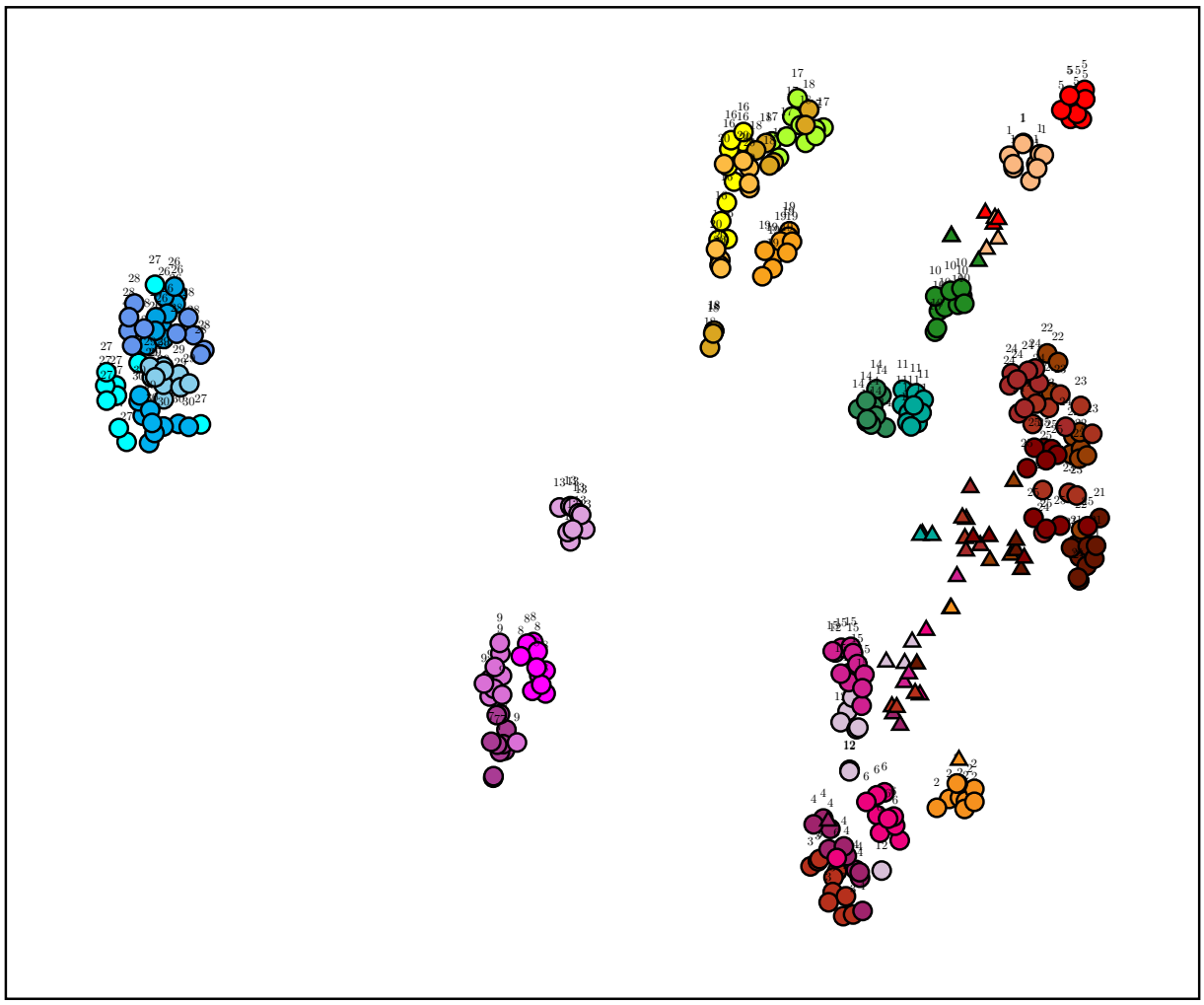


Figure 5.13: The same visualization as in Figure 5.12 using VGG-19 to extract the mid-level feature representation. Training samples are shown in circles and test samples are shown in triangles.

variations that might occur in the real testing conditions, for example, shadows, reflections, noise, blur, etc. That is one reason for the modest identification rate in the live testing. In order to make the proposed method to work, the image has to possess certain characteristics: sharp, even lighting and visibility of the hilum. I have shown some test images in Figure 5.11 that fulfill these requirements and were correctly identified.

With the raw test images, I also conducted a post-analysis with the VGG-19 network to see how CNN performs with these low quality images. The result is shown in Figure 5.7. We can see that the Top-3 identification rate is better than the my proposed method in Chapter 3, which implicitly demonstrates its ability of better robustness to the image degradation. But the performance is still worse than the one reported in chapter 4 because of the image variations. Visualization is also shown in Figure 5.13 to highlight this problem.

Therefore, in practice, if similar high quality images with real scale information is available, the proposed scale-pooling representation should be used for accurate identification. In contrast, if the input is some low quality images, it is better to switch to CNN-based method as an alternative to narrow down the number of possibilities. Moreover, from the user study, we can see that the time required for computer is pretty constant regardless of the difficulty of the seed and the level of expertise participants have. The reason behind this is that every seed sample has to undergo the same imaging pipeline. In contrast, analysts gain their expertise over time.

Seeds are inherently of different levels of difficulty for analysts. In other words, analysts can probably recognize certain seeds with a single glance but have to go through the seed specimens in the herbarium for assistance for others. Therefore the potential approach to best use the developed computer model is to combine the strength of both computer and analyst, that is asking analyst to do a pre-examination to filter out seeds that they are confident and easy to recognize. As for those left, they can be imaged with the help of an experienced photographer and then sent to the software for identification. This objective score form the computer would be beneficial for the seed regulation, not only for seed regulation agencies, but also companies that conduct import and export seed businesses. Further, if the operator can be well trained, this tool can then be used as a dedicated tool for efficient all-in-focus image capturing. This ability of quickly expanding the seed image dataset would enable the fine-tuning of the CNN.

5.5 Conclusion

In this chapter I designed a seed identification tool and conducted a user study for the evaluation of its effectiveness, with the users being highly trained human experts in seed identification. Results have shown significantly lower mental demand by using this tool. The identification rate was compromised by the input image quality degradation due to shadows, blur, and operation errors. This suggests a much longer training phase for the participants is necessary for the successful application of image-based identification (currently training phase takes only 15 minutes which is not sufficient for users to get familiar with the tool). A

tradeoff can be made between throughput and identification rate by switching between the scale-pooling representation and the CNN-based representation. The first one requires more time for image acquisition but has higher identification rate and the latter one can operate on low quality images and gives more accurate recommendations. In addition, this tool can be used as a dedicated imaging tool for efficient all-in-focus seed image capturing if we want to further expand the seed dataset. If 10 times as much or more data were ready for us to ensure proper training of the neural network (either fine-tuning or training from scratch), the CNN-based approach would likely be preferable due to its multi-scale nature and test-timed efficiency (one forward pass).

In this experiment, the identification system is compared with highly trained human experts. Although only modest results are achieved, we do see the potential of computer vision based identification methods. In the next chapter, several directions for future research are discussed.

Chapter 6

Conclusions and Future study

This thesis investigated new ideas to address the plant seed identification problem. I mainly concentrated on differentiating morphological similar plant seeds that is difficult for seed analysts to identify. The challenges of this problem are high-quality all-in-focus image acquisition and effective feature representation of seed images. Shallow depth of field is problematic for the observation of seed specimens in 2D images.

This thesis proposed software solutions to address a few aspects of these challenges. Chapter 3 introduced a sharpness metric with linear time complexity by exploiting the distribution difference of uniform LBP patterns in blurred and non-blurred image regions. It better measures the sharpness on low contrast sharp regions and behaves monotonically to the extents of defocus blur. A single-image-based defocus segmentation algorithm that also has linear time complexity was developed on top and achieved state-of-the-art performance. This metric has enabled the very first online focal stacking algorithm to my knowledge that does not require focal stacks to be captured before hand. Comparable results with state-of-the-art were achieved under low noise condition. Chapter 4 introduced a scale-pooling-based feature representation by using the commonly available pixel scale information for all-in-focus images with limited scale variations. I have found a series of pixel scales that better describes local image region and compute representations under these real scales for scale invariance. Multi-scale representations were concatenated for a scale-wise comparison in the classifier. A superior identification rate (95%) with all-in-focus images was achieved by just using the proposed representation and a linear SVM. In chapter 5, I designed the very first seed identification tool based on the proposed techniques and tested its effectiveness with a human study. I evaluated workload, throughput, and identification rate under computer-assisted condition and manual conditions. Significantly less mental demand is needed when using the tool. Although the identification rate in these tests is not as good as those reported in chapter 3, I have identified common mistakes that were made during the imaging capture step and possible ways to correct and avoid the problems.

Beyond this thesis, avenues for further work on plant seed identification can be divided into three categories: those that involve the acquisition of seed images, those that involve using 3D information rather than just 2D, and those that relate to the improvement of the blur segmentation algorithm, in particular the sharpness metric. These avenues are discussed in the following sections.

6.1 Image Acquisition

The training images are high quality images post-processed by professional imaging technician. For each species (subspecies, seed types), there are 10 samples to cover the biological variation. These images, however, do not contain image variations that could possiblly occur in real testing scenarios, for example, variable lighting, reflections, noise, scale variance etc. One remedy is by synthesizing these effects with parameterized variances on the high quality all-in-focus images (1000 images per species would be reasonable to generate by proper sampling in the parameterized space). Although this data augmentation technique can solve this problem to some extent, imaging more test samples would enable the full harnessing of the power of CNNs. Taking high quality all-in-focus images of many more seed samples (double the sample size would be more practical under current situation) under different view points would be beneficial to further boost the size of training data and ensure proper training of the CNNs. This should be assigned higher priority for future improvements. When data get expanded, it may be helpful to perform a statistical test to quantify the inherent difficulties of our different subsets of seeds as similar to Doddington et al.'s work [40].

Further more, due to the time-consuming nature of physical archival, retrieval, and equipment setup, imaging physical seed samples becomes a tedious job. Our use of off-the-shelf hardware components put some restrictions on the possible image operations. The proprietary software shipping with digital microscopes often do not have much flexibility for customization. In the future, customized accessory hardware can be designed and incorporated to facilitate the image acquisition. One such example would be using motorized stages which can move freely in the x, y, z axes for automatic microscope calibration and focal stacking. This can not only free users from manual glass slide moving but also make possible other digital representations of microscopic specimens, e.g. the virtual reflected-light microscopy (VRLM) representation [62]. Another hardware that could be employed is a seed sorter to preprocess the test seed sample to ensure that the seed samples identified in contiguous are homogeneous in size, therefore avoiding frequent scale calibration operation.

6.2 Exploring 3D information

In this thesis, the main focus is on software solutions for seed identification. We are currently dealing with 2D colour images that are projected from 3D world. The discarded 3D shape information (depth) could be useful for identification but has not been explored. Therefore, another direction worth exploring would be recovering depth cues from imaging and extracting features from 3D surfaces, either from focal stacks or stereo image pairs [7, 153].

6.3 Improving the Blur Segmentation Method

The proposed metric was inspired by the statistical difference of local binary patterns of a set of partial blurred images. Since the source of blurriness is mainly defocus blur, my metric currently is only capable of detecting defocus blur. Given that there are other type of blurriness such as those introduced by low qualities of lens and materials in imaging systems and motion blur, it would be worth studying the blur model due to the properties of optical devices [92] and at the same time exploring properties of different patterns such as the non-uniform binary patterns and local ternary pattern (LTP) [166] on blur regions of different types. Moreover, the ideas used in noise-resistant LBP (NRLBP) [146], which treats pixels susceptible to noise as having uncertain state and then determines the corresponding bit value based on the other bits of the LBP code, might worth borrowing if explicit handling of noise in blur detection is desired.

Alternatively, since CNN is particularly good at extracting cascading abstraction features, it might be useful to use CNN to learn the sharpness feature directly from the training data in an end-to-end fashion. The input can be a image patch that is manually blurred by a pre-defined blur kernel, e.g. the Gaussian kernel. Then the output of this network should be the parameter σ. This could be formed as a regression problem by using the L_2 loss as the loss function and back propagation to learn the weights automatically. One advantage of doing this is that the training data is unlimited. Basically any sharp image patches can be used for the training so we do not have to worry about the size of the training data. More interestingly, Felix et al. found that LBP can be generalized and implemented as local binary convolution as an efficient alternative to convolutional layers [81]. It would be worthwhile to research how such kinds of network architectures can be explored to learn an optimal sharpness metric and such metrics correlate with my proposed hand-engineered metric. In the meantime, instead of using CNN to lean local sharpness features, they can also be used to directly generate defocus maps by utilizing the high-level information in the deeper layers of CNN. These high-level semantics are claimed to be important to solve the ambiguity of smooth regions [114].

Finally, the conditional random field used to combine sharpness information from multiple scales could be also formulated as a Recurrent Neural Network (RNN) [206]. Thus the whole segmentation method could be formulated in a way so that only CNNs are used.

References

[1] The Australasian pollen and spore atlas. http://apsa.anu.edu.au.

[2] Brotaz texture dataset. http://multibandtexture.recherche.usherbrooke.ca/original_brodatz.html.

[3] Seed technologist training manual. http://www.seedtechnology.net/seed_technologists_training_manual.

[4] Weed seeds order. http://laws-lois.justice.gc.ca/eng/regulations/sor-2005-220/index.html, 2005.

[5] Results of imagenet challenge 2012. http://www.image-net.org/challenges/LSVRC/2012/results.html, 2012.

[6] Radhakrishna Achanta, Sheila Hemami, Francisco Estrada, and Sabine Susstrunk. Frequency-tuned salient region detection. In *Computer vision and pattern recognition, 2009. cvpr 2009. ieee conference on*, pages 1597–1604. IEEE, 2009.

[7] François Aguet, Dimitri Van De Ville, and Michael Unser. Model-based 2.5-d deconvolution for extended depth of field in brightfield microscopy. *Image Processing, IEEE Transactions on*, 17(7):1144–1153, 2008.

[8] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. Face description with local binary patterns: Application to face recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(12):2037–2041, 2006.

[9] José Luis Araus, Anna Febrero, Ramon Buxó, Maria Oliva Rodrı guez Ariza, Fernando Molina, Marı a Dolores Camalich, Dimas Martı n, and Jordi Voltas. Identification of ancient irrigation practices based on the carbon isotope discrimination of plant seeds: a case study from the south-east iberian peninsula. *Journal of Archaeological Science*, 24(8):729–740, 1997.

[10] Hossein Azizpour, Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. From generic to specific deep representations for visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 36–45, 2015.

[11] Soonmin Bae and Frédo Durand. Defocus magnification. In *Computer Graphics Forum*, volume 26, pages 571–579. Wiley Online Library, 2007.

[12] Khosro Bahrami, Alex C Kot, and Jiayuan Fan. A novel approach for partial blur detection and segmentation. In *Multimedia and Expo (ICME), 2013 IEEE International Conference on*, pages 1–6. IEEE, 2013.

[13] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. SURF: Speeded up robust features. In *Computer Vision–ECCV 2006*, pages 404–417. Springer, 2006.

[14] Yoshua Bengio, Pascal Lamblin, Dan Popovici, Hugo Larochelle, et al. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19:153, 2007.

[15] Yoshua Bengio, Pascal Lamblin, Dan Popovici, Hugo Larochelle, et al. Greedy layer-wise training of deep networks. *Advances in neural information processing systems*, 19:153, 2007.

[16] Tamara L Berg, Alexander C Berg, and Jonathan Shih. Automatic attribute discovery and characterization from noisy web data. In *Computer Vision–ECCV 2010*, pages 663–676. Springer, 2010.

[17] Thomas Berg and Peter N Belhumeur. POOF: Part-based one-vs.-one features for fine-grained categorization, face verification, and attribute estimation. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 955–962. IEEE, 2013.

[18] Thomas Berg, Jiongxin Liu, Seung Lee, Michelle Alexander, David Jacobs, and Peter Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2011–2018, 2013.

[19] Gedas Bertasius, Jianbo Shi, and Lorenzo Torresani. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. *arXiv preprint arXiv:1412.1123*, 2014.

[20] J Derek Bewley, Michael Black, and Peter Halmer. *The encyclopedia of seeds: science, technology and uses*. CABI, 2006.

[21] Ali Borji and Laurent Itti. Human vs. computer in scene and object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 113–120, 2014.

[22] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Image classification using random forests and ferns. In *International Conference on Computer Vision*, pages 1–8, 2007.

[23] Anna Bosch, Andrew Zisserman, and Xavier Muoz. Scene classification using a hybrid generative/discriminative approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(4):712–727, 2008.

[24] Lubomir Bourdev and Jitendra Malik. Poselets: Body part detectors trained using 3D human pose annotations. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1365–1372. IEEE, 2009.

[25] Y-Lan Boureau, Jean Ponce, and Yann LeCun. A theoretical analysis of feature pooling in visual recognition. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 111–118, 2010.

[26] Steve Branson, Grant Van Horn, Catherine Wah, Pietro Perona, and Serge Belongie. The ignorant led by the blind: A hybrid human–machine vision system for fine-grained categorization. *International Journal of Computer Vision*, pages 1–27, 2014.

[27] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 60–65. IEEE, 2005.

[28] GJ Burton and Ian R Moorhead. Color and spatial structure in natural scenes. *Applied Optics*, 26(1):157–170, 1987.

[29] Ayan Chakrabarti, Todd Zickler, and William T Freeman. Analyzing spatially-varying blur. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2512–2519. IEEE, 2010.

[30] MY Chaloupka and SB Domm. Role of anthropochory in the invasion of coral cays by alien flora. *Ecology*, pages 1536–1547, 1986.

[31] Shi Changjiang and Ji Guangrong. Recognition method of weed seeds based on computer vision. In *Image and Signal Processing, 2009. CISP'09. 2nd International Congress on*, pages 1–4. IEEE, 2009.

[32] Ken Chatfield. bag of words. `http://www.robots.ox.ac.uk/~vgg/research/encoding_eval/`, 2011.

[33] Ken Chatfield, Victor Lempitsky, Andrea Vedaldi, and Andrew Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *British Machine Vision Conference*, 2011.

[34] Taeg Sang Cho. *Motion blur removal from photographs*. PhD thesis, Citeseer, 2010.

[35] Florent Couzinie-Devy, Jian Sun, Karteek Alahari, and Jean Ponce. Learning to estimate and remove non-uniform image blur. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1075–1082. IEEE, 2013.

[36] Nello Cristianini, John Shawe-Taylor, and Huma Lodhi. Latent semantic kernels. *Journal of Intelligent Information Systems*, 18(2-3):127–152, 2002.

[37] Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, page 22, 2004.

[38] 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.

[39] Jeffrey Dean, Greg Corrado, Rajat Monga, Kai Chen, Matthieu Devin, Mark Mao, Andrew Senior, Paul Tucker, Ke Yang, Quoc V Le, et al. Large scale distributed deep networks. In *Advances in Neural Information Processing Systems*, pages 1223–1231, 2012.

[40] George Doddington, Walter Liggett, Alvin Martin, Mark Przybocki, and Douglas Reynolds. Sheep, goats, lambs and wolves: A statistical analysis of speaker performance in the nist 1998 speaker recognition evaluation. Technical report, DTIC Document, 1998.

[41] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[42] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2650–2658, 2015.

[43] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014.

[44] Ali Farhadi, Ian Endres, and Derek Hoiem. Attribute-centric recognition for cross-category generalization. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2352–2359. IEEE, 2010.

[45] Ali Farhadi, Ian Endres, Derek Hoiem, and David Forsyth. Describing objects by their attributes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1778–1785. IEEE, 2009.

[46] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

[47] Pedro F Felzenszwalb and Daniel P Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005.

[48] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T Roweis, and William T Freeman. Removing camera shake from a single photograph. In *ACM Transactions on Graphics (TOG)*, volume 25, pages 787–794. ACM, 2006.

[49] Robert Fergus, Pietro Perona, and Andrew Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–264. IEEE, 2003.

[50] Rony Ferzli and Lina J Karam. A no-reference objective image sharpness metric based on the notion of just noticeable blur (JNB). *Image Processing, IEEE Transactions on*, 18(4):717–728, 2009.

[51] David J Field. Relations between the statistics of natural images and the response properties of cortical cells. *JOSA A*, 4(12):2379–2394, 1987.

[52] Martin A Fischler and Robert A Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 22(1):67–92, 1973.

[53] Efstratios Gavves, Basura Fernando, Cees GM Snoek, Arnold WM Smeulders, and Tinne Tuytelaars. Fine-grained categorization by alignments. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1713–1720. IEEE, 2013.

[54] Kurt R Geitzenauer. Coccoliths as late quaternary palaeoclimatic indicators in the subantarctic pacific ocean. 1969.

[55] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *Computer Vision–ECCV 2014*, pages 392–407. Springer, 2014.

[56] Pablo M Granitto, Hugo D Navone, Pablo F Verdes, and HA Ceccatto. Weed seeds identification by machine vision. *Computers and Electronics in Agriculture*, 33(2):91–103, 2002.

[57] Pablo M Granitto, Pablo F Verdes, and H Alejandro Ceccatto. Large-scale investigation of weed seed identification by machine vision. *Computers and Electronics in Agriculture*, 47(1):15–24, 2005.

[58] Alan Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6645–6649. IEEE, 2013.

[59] Werner Greuter, J McNeill, F_R Barrie, HM Burdet, V Demoulin, TS Filgueiras, DH Nicholson, PC Silva, JE Skog, P Trehane, et al. International code of botanical nomenclature (saint louis code): Sixteenth international botanical congress, st louis, missouri, usa, july-august 1999. In *International code of botanical nomenclature (Saint Louis Code): Sixteenth International Botanical Congress, St Louis, Missouri, USA, July-August 1999*. International Association for Plant Taxonomy, 2000.

[60] Anthony E Hall, Glen H Cannell, Harry W Lawton, et al. *Agriculture in semi-arid environments.* Springer Verlag., 1979.

[61] Chris Harris and Mike Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988.

[62] Adam P Harrison, Cindy M Wong, and Dileepan Joseph. Virtual reflected-light microscopy. *Journal of microscopy*, 244(3):293–304, 2011.

[63] Hadzli Hashim, Fairul Nazmie Osman, Syed Abdul Mutalib Al Junid, Muhammad Adib Haron, and Hajar Mohd Salleh. An intelligent classification model for rubber seed clones based on shape features through imaging techniques. In *Intelligent Systems, Modelling and Simulation (ISMS). International Conference on*, pages 25–31. IEEE, 2010.

[64] Rania Hassen, Zhou Wang, Magdy M Salama, et al. Image sharpness assessment based on local phase coherence. *Image Processing, IEEE Transactions on*, 22(7):2798–2810, 2013.

[65] Kaiming He, Jian Sun, and Xiaoou Tang. Fast matting using large kernel matting laplacian matrices. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2165–2172. IEEE, 2010.

[66] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[67] Glenn E Healey and Raghava Kondepudy. Radiometric ccd camera calibration and noise estimation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 16(3):267–276, 1994.

[68] Marko Heikkila and Matti Pietikainen. A texture-based method for modeling the background and detecting moving objects. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):657–662, 2006.

[69] David Held, Jesse Levinson, and Sebastian Thrun. A probabilistic framework for car detection in images using context and scale. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1628–1634. IEEE, 2012.

[70] Geoffrey Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[71] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[72] Richard J Hobbs and Stella E Humphries. An integrated approach to the ecology and management of plant invasions. *Conservation Biology*, 9(4):761–770, 1995.

[73] Yongzhen Huang, Zifeng Wu, Liang Wang, and Tieniu Tan. Feature coding in image classification: A comprehensive study. *IEEE transactions on pattern analysis and machine intelligence*, 2013.

[74] David H Hubel and Torsten N Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243, 1968.

[75] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[76] Tommi Jaakkola, David Haussler, et al. Exploiting generative models in discriminative classifiers. *Advances in neural information processing systems*, pages 487–493, 1999.

[77] Hans Arne Jensen. *Bibliography on seed morphology*. CRC Press, 1998.

[78] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.

[79] Jiaya Jia Jianping Shi, Li Xu. Just noticeable defocus blur detection and estimation. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, 2015.

[80] Thorsten Joachims. *Text categorization with support vector machines: Learning with many relevant features*. Springer, 1998.

[81] Felix Juefei-Xu, Vishnu Naresh Boddeti, and Marios Savvides. Local binary convolutional neural networks. *arXiv preprint arXiv:1608.06049*, 2016.

[82] Le Kang, Peng Ye, Yi Li, and David Doermann. Convolutional neural networks for no-reference image quality assessment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1733–1740, 2014.

[83] Vladimir Katkovnik, Alessandro Foi, Karen Egiazarian, and Jaakko Astola. From local kernel to nonlocal multiple-model image denoising. *International journal of computer vision*, 86(1):1–32, 2010.

[84] Koray Kavukcuoglu, M Ranzato, Rob Fergus, and Yann LeCun. Learning invariant features through topographic filter maps. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1605–1612. IEEE, 2009.

[85] Changick Kim. Segmenting a low-depth-of-field image using morphological filters and region merging. *Image Processing, IEEE Transactions on*, 14(10):1503–1511, 2005.

[86] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[87] Piotr Koniusz, Fei Yan, and Krystian Mikolajczyk. Comparison of mid-level feature coding approaches and pooling strategies in visual concept detection. *Computer Vision and Image Understanding*, 2012.

[88] Dilip Krishnan and Rob Fergus. Fast image deconvolution using hyper-laplacian priors. In *Advances in Neural Information Processing Systems*, pages 1033–1041, 2009.

[89] 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.

[90] Eric Krotkov. Focusing. *International Journal of Computer Vision*, 1(3):223–237, 1988.

[91] Neeraj Kumar, Alexander C Berg, Peter N Belhumeur, and Shree K Nayar. Attribute and simile classifiers for face verification. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 365–372. IEEE, 2009.

[92] Deepa Kundur and Dimitrios Hatzinakos. Blind image deconvolution. *Signal Processing Magazine, IEEE*, 13(3):43–64, 1996.

[93] Griffin Lacey, Graham W Taylor, and Shawki Areibi. Deep learning on fpgas: Past, present, and future. *arXiv preprint arXiv:1602.04283*, 2016.

[94] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[95] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 951–958. IEEE, 2009.

[96] Natalia Larios, Hongli Deng, Wei Zhang, Matt Sarpola, Jenny Yuen, Robert Paasch, Andrew Moldenke, David A Lytle, Salvador Ruiz Correa, Eric N Mortensen, et al. Automated insect identification through concatenated histograms of local appearance features: feature vector generation and region detection for deformable objects. *Machine Vision and Applications*, 19(2):105–123, 2008.

[97] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178. IEEE, 2006.

[98] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[99] Yann LeCun, Koray Kavukcuoglu, and Clément Farabet. Convolutional networks and applications in vision. In *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, pages 253–256. IEEE, 2010.

[100] Thomas Leung and Jitendra Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, 2001.

[101] Anat Levin, Dani Lischinski, and Yair Weiss. A closed-form solution to natural image matting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):228–242, 2008.

[102] Anat Levin, Yair Weiss, Fredo Durand, and William T Freeman. Understanding and evaluating blind deconvolution algorithms. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1964–1971. IEEE, 2009.

[103] Guanbin Li and Yizhou Yu. Visual saliency based on multiscale deep features. *arXiv preprint arXiv:1503.08663*, 2015.

[104] Tony Lindeberg. Feature detection with automatic scale selection. *International journal of computer vision*, 30(2):79–116, 1998.

[105] Ce Liu, Richard Szeliski, Sing Bing Kang, C Lawrence Zitnick, and William T Freeman. Automatic estimation and removal of noise from a single image. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):299–314, 2008.

[106] Fayao Liu, Chunhua Shen, and Guosheng Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5162–5170, 2015.

[107] Jiongxin Liu, Angjoo Kanazawa, David Jacobs, and Peter Belhumeur. Dog breed classification using part localization. In *Computer Vision–ECCV 2012*, pages 172–185. Springer, 2012.

[108] Renting Liu, Zhaorong Li, and Jiaya Jia. Image partial blur detection and classification. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[109] Xinhao Liu, Masayuki Tanaka, and Masatoshi Okutomi. Single-image noise level estimation for blind denoising. *IEEE transactions on image processing*, 22(12):5226–5237, 2013.

[110] Huma Lodhi, Craig Saunders, John Shawe-Taylor, Nello Cristianini, and Chris Watkins. Text classification using string kernels. *The Journal of Machine Learning Research*, 2:419–444, 2002.

[111] Sajad Ali Lone. Seed collection and identification of forest trees of kashmir. 2014.

[112] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[113] Benjamaporn Lurstwut and Chomtip Pornpanomchai. Plant seed image recognition system (PSIRS). *IACSIT International Journal of Engineering and Technology*, pages 600–605, 2011.

[114] Kede Ma, Huan Fu, Tongliang Liu, Zhou Wang, and Dacheng Tao. Local blur mapping: Exploiting high-level semantics by deep neural networks. *arXiv preprint arXiv:1612.01227*, 2016.

[115] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[116] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5188–5196, 2015.

[117] Julien Mairal, Francis Bach, Jean Ponce, Guillermo Sapiro, and Andrew Zisserman. Non-local sparse models for image restoration. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 2272–2279. IEEE, 2009.

[118] Y Marc'Aurelio Ranzato, Lan Boureau, and Yann LeCun. Sparse feature learning for deep belief networks. *Advances in neural information processing systems*, 20:1185–1192, 2007.

[119] Pina Marziliano, Frederic Dufaux, Stefan Winkler, and Touradj Ebrahimi. A no-reference perceptual blur metric. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 3, pages III–57. IEEE, 2002.

[120] Krystian Mikolajczyk. *Detection of local features invariant to affines transformations*. PhD thesis, Institut National Polytechnique de Grenoble-INPG, 2002.

[121] Krystian Mikolajczyk and Cordelia Schmid. Scale & affine invariant interest point detectors. *International journal of computer vision*, 60(1):63–86, 2004.

[122] Krystian Mikolajczyk and Cordelia Schmid. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615–1630, 2005.

[123] Kevin P Murphy, Yair Weiss, and Michael I Jordan. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 467–475. Morgan Kaufmann Publishers Inc., 1999.

[124] Albina F Musil et al. Identification of crop and weed seeds. *Identification of crop and weed seeds.*, 1963.

[125] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.

[126] Niranjan D Narvekar and Lina J Karam. A no-reference image blur metric based on the cumulative probability of blur detection (cpbd). *Image Processing, IEEE Transactions on*, 20(9):2678–2683, 2011.

[127] Paul G Nevill, Mark J Wallace, Joseph T Miller, and Siegfried L Krauss. Dna barcoding for conservation, seed banking and ecological restoration of acacia in the midwest of western australia. *Molecular ecology resources*, 13(6):1033–1042, 2013.

[128] M-E Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Computer Vision, Graphics and Image Processing Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on*, pages 722–729. IEEE, 2008.

[129] Eric Nowak, Frédéric Jurie, and Bill Triggs. Sampling strategies for bag-of-features image classification. In *Computer Vision–ECCV 2006*, pages 490–503. Springer, 2006.

[130] Timo Ojala and Matti Pietikäinen. Unsupervised texture segmentation using feature distributions. *Pattern Recognition*, 32(3):477–486, 1999.

[131] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition*, 29(1):51–59, 1996.

[132] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, 2002.

[133] Lori A Overturf, Mary L Comer, and Edward J Delp. Color image coding using morphological pyramid decomposition. *IEEE Transactions on Image Processing*, 4(2):177–185, 1995.

[134] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3498–3505. IEEE, 2012.

[135] Otávio AB Penatti, Keiller Nogueira, and Jefersson A dos Santos. Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 44–51, 2015.

[136] Alex Paul Pentland. A new sense for depth of field. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (4):523–531, 1987.

[137] Federico Perazzi, Philipp Krahenbuhl, Yael Pritch, and Alexander Hornung. Saliency filters: Contrast based filtering for salient region detection. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 733–740. IEEE, 2012.

[138] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the Fisher kernel for large-scale image classification. In *Computer Vision–ECCV 2010*, pages 143–156. Springer, 2010.

[139] Said Pertuz, Domenec Puig, and Miguel Angel Garcia. Analysis of focus measure operators for shape-from-focus. *Pattern Recognition*, 46(5):1415–1432, 2013.

[140] Said Pertuz, Domenec Puig, Miguel Angel Garcia, and Andrea Fusiello. Generation of all-in-focus images by noise-robust selective fusion of limited depth-of-field images. *Image Processing, IEEE Transactions on*, 22(3):1242–1251, 2013.

[141] Matti Pietikäinen, Tomi Nurmela, Topi Mäenpää, and Markus Turtinen. View-based recognition of real-world textures. *Pattern Recognition*, 37(2):313–323, 2004.

[142] KS Pradeep and AN Rajagopalan. Improving shape from focus using defocus cue. *Image Processing, IEEE Transactions on*, 16(7):1920–1925, 2007.

[143] Creative Commons PROYECTO AGUA**/**WATER PROJECT. Microagua. `https://www.flickr.com/photos/microagua/`.

[144] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 413–420. IEEE, 2009.

[145] Ali S Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on*, pages 512–519. IEEE, 2014.

[146] Jianfeng Ren, Xudong Jiang, and Junsong Yuan. Noise-resistant local binary pattern with an embedded error-correction mechanism. *Image Processing, IEEE Transactions on*, 22(10):4049–4060, 2013.

[147] Douglas Reynolds. Gaussian mixture models. *Encyclopedia of Biometrics*, pages 659–663, 2009.

[148] Jo Jones Ruojing Wang, Jennifer Neudorf. Canadian food inspection agency seed dataset, 2015.

[149] Olga Russakovsky, Jia Deng, Zhiheng Huang, Alexander C Berg, and Li Fei-Fei. Detecting avocados to zucchinis: what have we done, and where are we going? In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2064–2071. IEEE, 2013.

[150] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *arXiv preprint arXiv:1409.0575*, 2014.

[151] Olga Russakovsky, Yuanqing Lin, Kai Yu, and Li Fei-Fei. Object-centric spatial pooling for image classification. In *Computer Vision–ECCV 2012*, pages 1–15. Springer, 2012.

[152] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image classification with the fisher vector: Theory and practice. *International journal of computer vision*, 105(3):222–245, 2013.

[153] HW Schreier, D Garcia, and MA Sutton. Advances in light microscope stereo vision. *Experimental mechanics*, 44(3):278–288, 2004.

[154] Saskatoon Laboratory (Seed Science and Technology Section). *Accredited seed testing laboratory proficiency monitoring program.* Canadian Food Inspection Agency, 2.0 edition, January 2012.

[155] Asma Rejeb Sfar, Nozha Boujemaa, and Donald Geman. Vantage feature frames for fine-grained categorization. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 835–842. IEEE, 2013.

[156] Qi Shan, Jiaya Jia, and Aseem Agarwala. High-quality motion deblurring from a single image. In *ACM Transactions on Graphics (TOG)*, volume 27, page 73. ACM, 2008.

[157] W. C. Shaw. Integrated weed management systems technology for pest management. *Weed science*, 30:2–12, 1981.

[158] Jianping Shi, Li Xu, and Jiaya Jia. Blur detection dataset. `http://www.cse.cuhk.edu.hk/~leojia/projects/dblurdetect/dataset.html`, 2014.

[159] Jianping Shi, Li Xu, and Jiaya Jia. Discriminative blur detection features. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2965–2972. IEEE, 2014.

[160] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[161] Michael G Simpson. *Plant systematics.* Academic press, 2010.

[162] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[163] Bolan Su, Shijian Lu, and Chew Lim Tan. Blurred image region detection and classification. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 1397–1400. ACM, 2011.

[164] Murali Subbarao, Tae-Sun Choi, and Arman Nikzad. Focusing techniques. *Optical Engineering*, 32(11):2824–2836, 1993.

[165] Nima Tajbakhsh, Jae Y Shin, Suryakanth R Gurudu, R Todd Hurst, Christopher B Kendall, Michael B Gotway, and Jianming Liang. Convolutional neural networks for medical image analysis: Full training or fine tuning? *IEEE transactions on medical imaging*, 35(5):1299–1312, 2016.

[166] Xiaoyang Tan and Bill Triggs. Enhanced local texture feature sets for face recognition under difficult lighting conditions. *Image Processing, IEEE Transactions on*, 19(6):1635–1650, 2010.

[167] Thirapiroon Thongkamwitoon, Hani Muammar, and Pier-Luigi Dragotti. An image recapture detection algorithm based on learning dictionaries of edge profiles. *Information Forensics and Security, IEEE Transactions on*, 10(5):953–968, 2015.

[168] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. 2012.

[169] Alexander Toet. Image fusion by a ratio of low-pass pyramid. *Pattern Recognition Letters*, 9(4):245–253, 1989.

[170] Alexander Toet, Lodewik J Van Ruyven, and J Mathee Valeton. Merging thermal and visual images by a contrast pyramid. *Optical Engineering*, 28(7):287789–287789, 1989.

[171] Simon Tong and Daphne Koller. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2:45–66, 2002.

[172] Lisa Torrey and Jude Shavlik. Transfer learning. *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, 1:242, 2009.

[173] Yanghai Tsin, Visvanathan Ramesh, and Takeo Kanade. Statistical calibration of ccd imaging process. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 480–487. IEEE, 2001.

[174] Antonio G Valdecasas, David Marshall, Jose M Becerra, and JJ Terrero. On the extended depth of focus algorithms for bright field microscopy. *Micron*, 32(6):559–569, 2001.

[175] Domenec Puig Valls and MiguelAngel Garcıa Garcıa. Modeling and applications of the focus cue in conventional digital cameras.

[176] van A van der Schaaf and JH van van Hateren. Modelling the power spectra of natural images: statistics and information. *Vision research*, 36(17):2759–2770, 1996.

[177] Andrea Vedaldi, Brian Fulkerson, Karel Lenc, Daniele Perrone, Michal Perdoch, Milan Sulc, and Hana Sarbortova. Fisher kernel. `http://www.vlfeat.org/api/fisher-kernel.html`, 2013.

[178] Andrea Vedaldi, Siddarth Mahendran, Stavros Tsogkas, Subhransu Maji, Ross B Girshick, Juho Kannala, Esa Rahtu, Iasonas Kokkinos, Matthew B Blaschko, David Weiss, et al. Understanding objects in detail with fine-grained attributes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3622–3629, 2014.

[179] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. *arXiv preprint arXiv:1411.4555*, 2014.

[180] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[181] Cuong T Vu, Thien D Phan, and Damon M Chandler. : A spectral and spatial measure of local perceived sharpness in natural images. *Image Processing, IEEE Transactions on*, 21(3):934–945, 2012.

[182] Catherine Wah, Steve Branson, Pietro Perona, and Serge Belongie. Multiclass recognition and part localization with humans in the loop. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2524–2531. IEEE, 2011.

[183] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. *Computation and Neural Systems Technical Report*, 2011.

[184] James Z Wang, Jia Li, Robert M Gray, and Gio Wiederhold. Unsupervised multiresolution segmentation for images with low depth of field. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(1):85–90, 2001.

[185] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, and Yihong Gong. Locality-constrained linear coding for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3360–3367. IEEE, 2010.

[186] Xingxing Wang, LiMin Wang, and Yu Qiao. A comparative study of encoding, pooling and normalization methods for action recognition. In *Computer Vision–ACCV 2012*, pages 572–585. Springer, 2013.

[187] Zhenhua Wang, Bin Fan, and Fuchao Wu. Local intensity order pattern for feature description. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 603–610. IEEE, 2011.

[188] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612, 2004.

[189] Mary F Willson and Anna Traveset. The ecology of seed dispersal. *Seeds: The ecology of regeneration in plant communities*, 2:85–110, 2000.

[190] John Winn, Antonio Criminisi, and Thomas Minka. Object categorization by learned universal visual dictionary. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1800–1807. IEEE, 2005.

[191] Chee Sun Won, Kyungsuk Pyun, and Robert M Gray. Automatic object segmentation in images with low depth of field. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 3, pages 805–808. IEEE, 2002.

[192] Earl Wong. A new method for creating a depth map for camera auto focus using an all in focus picture and 2d scale space matching. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, 2006.

[193] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 3485–3492. IEEE, 2010.

[194] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1794–1801. IEEE, 2009.

[195] Xiaoyan Yang, Zhiwei Wan, Linda Perry, Houyuan Lu, Qiang Wang, Chaohong Zhao, Jun Li, Fei Xie, Jincheng Yu, Tianxing Cui, et al. Early millet use in northern china. *Proceedings of the National Academy of Sciences*, 109(10):3726–3730, 2012.

[196] Bangpeng Yao, Gary Bradski, and Li Fei-Fei. A codebook-free and annotation-free approach for fine-grained image categorization. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3466–3473. IEEE, 2012.

[197] Xin Yi, Mark Eramian, Ruojing Wang, and Eric Neufeld. Identification of morphologically similar seeds using multi-kernel learning. In *Computer and Robot Vision (CRV), 2014 Canadian Conference on*, pages 143–150. IEEE, 2014.

[198] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015.

[199] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[200] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833. Springer, 2014.

[201] Jianguo Zhang, Marcin Marszałek, Svetlana Lazebnik, and Cordelia Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International journal of computer vision*, 73(2):213–238, 2007.

[202] Ning Zhang, Ryan Farrell, and Trevor Darrell. Pose pooling kernels for sub-category recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3665–3672. IEEE, 2012.

[203] Jufeng Zhao, Huajun Feng, Zhihai Xu, Qi Li, and Xiaoping Tao. Automatic blur region segmentation approach using image matting. *Signal, Image and Video Processing*, 7(6):1173–1181, 2013.

[204] Wencang Zhao and Junxin Wang. Study of feature extraction based visual invariance and species identification of weed seeds. In *Natural Computation (ICNC), 2010 Sixth International Conference on*, volume 2, pages 631–635. IEEE, 2010.

[205] Liang Zheng, Shengjin Wang, Fei He, and Qi Tian. Seeing the big picture: Deep embedding with contextual evidences. *arXiv preprint arXiv:1406.0132*, 2014.

[206] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537, 2015.

[207] Xi Zhou, Kai Yu, Tong Zhang, and Thomas S Huang. Image classification using super-vector coding of local image descriptors. In *Computer Vision–ECCV 2010*, pages 141–154. Springer, 2010.

[208] Xiang Zhu. *Measuring spatially varying blur and its application in digital image restoration.* PhD thesis, UNIVERSITY OF CALIFORNIA, SANTA CRUZ, 2013.

[209] Xiang Zhu, Scott Cohen, Stephen Schiller, and Peyman Milanfar. Estimating spatially varying defocus blur from a single image. *Image Processing, IEEE Transactions on*, 22(12):4879–4891, 2013.

[210] Shaojie Zhuo and Terence Sim. Defocus map estimation from a single image. *Pattern Recognition*, 44(9):1852–1858, 2011.

[211] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 479–486. IEEE, 2011.

Appendix A

Raw results for the user experiment

A.1 Raw TLX score for each participant

Figure A.1 shows the aggregated TLX score for each participant.

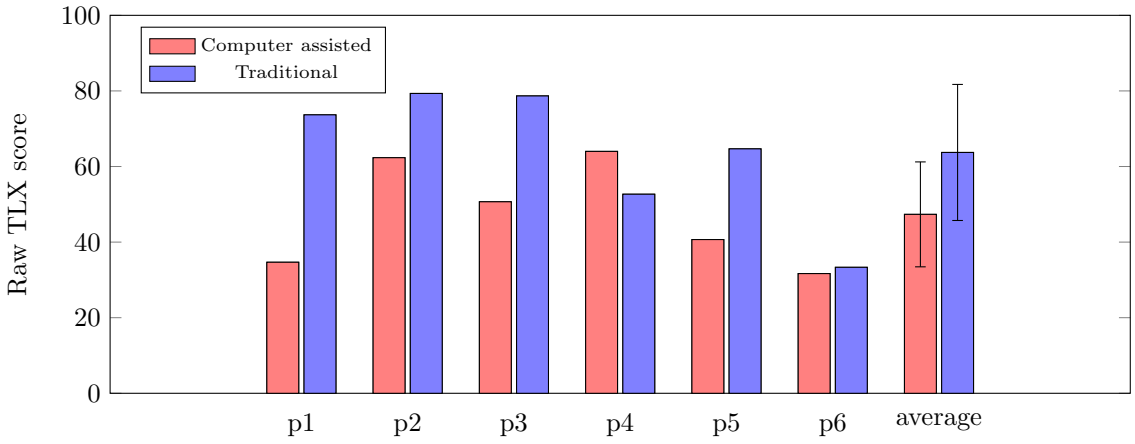


Figure A.1: Raw TLX score for each participant. Red is the workload pretending to use everyday, whereas blur is the workload for manual traditional method.

A.2 TLX score in each dimension

Figure A.2 A.3 shows the TLX score in the six dimension respectively.

A.3 User feedback

Following shows the feedback of the participants by using the questionnaire.

- What do you like the best of the tool?
  - p1: The fact that it could possibly be used to confirm an identification. It may be used instead of consulting another analyst, i.e. provide a 2nd opinion.
  - p2: Easy to use
  - p3: Easy to operate. Gives results quickly.
  - p4: It can give me some clue in finding the family/Genus of the unknown sample.
  - p5: Stacked (3-D) image. provides 3 best matches for seed ID.
  - p6: How easy it is to use.
- Which part do you think can be improved?
  - p1: It is hard to orientate the seeds correctly-hard to get into the field of view. (the computer program is simple and user friendly though)
  - p2: It would be nice to set seed specific so does not roll or move.

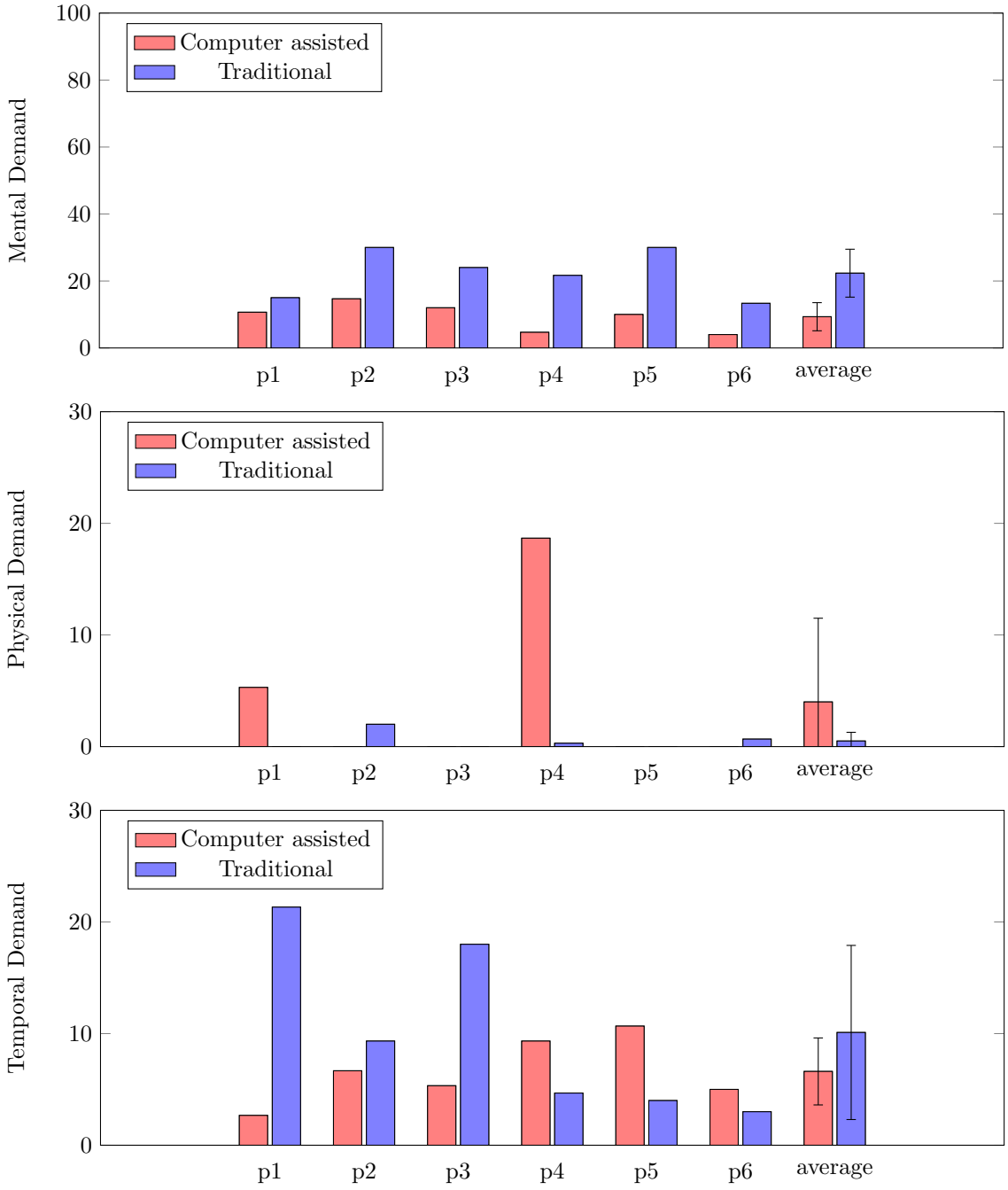


Figure A.2: TLX score in each dimension for each participant. Red is the workload pretending to use everyday, whereas blur is the workload for manual traditional method.

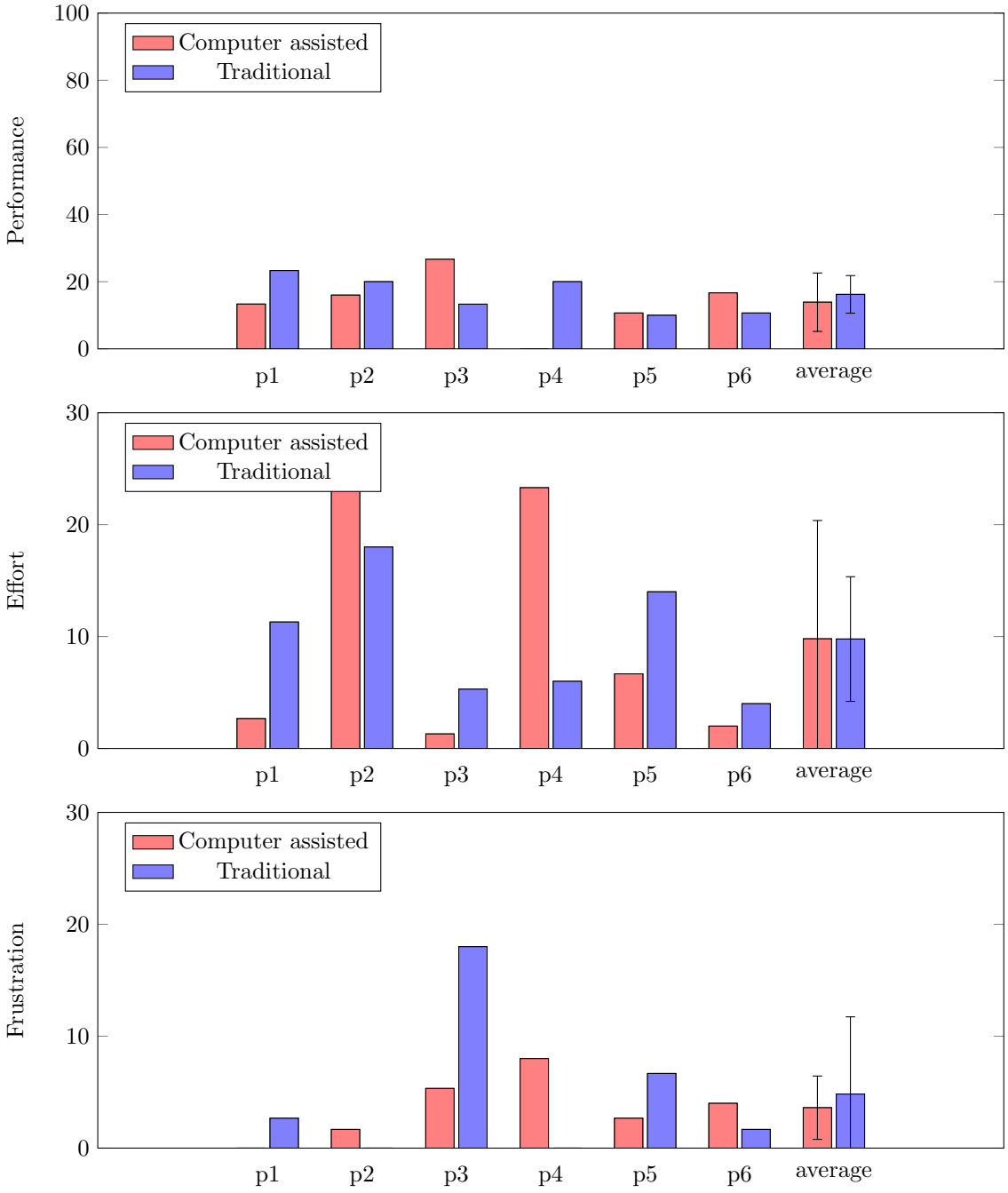


Figure A.3: Continue of the above Figure. TLX score in each dimension for each participant. Red is the workload pretending to use everyday, whereas blur is the workload for manual traditional method.

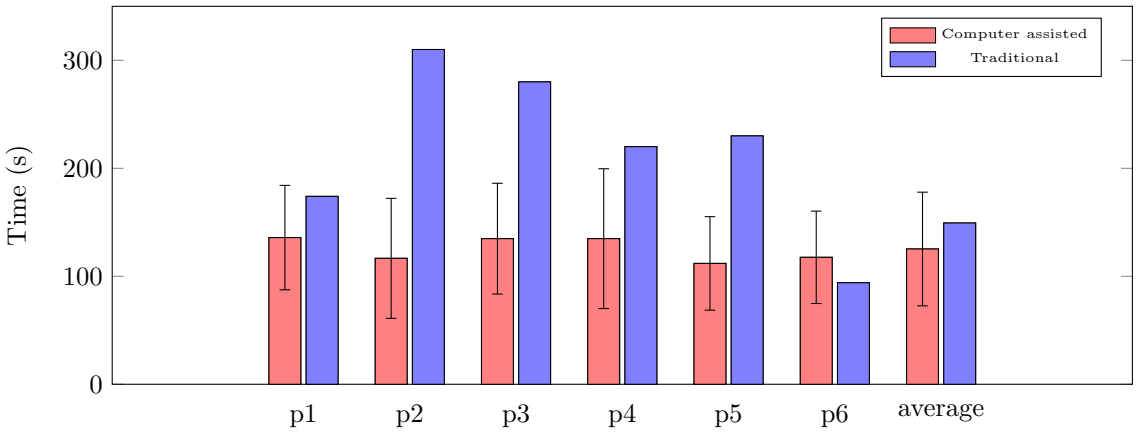


Figure A.4: Average time spent on one sample for each participant.

  - p3: Sometimes can not focus all parts of the seed completely.
  - p4: Camera/Monitor Quality? Ringlighting instead of the swan neck lighting?
  - p5: Somehow have slide/surface that prevents very round seeds from rolling away so easily.
  - p6: Incorporate more species if use on greater scale. Right now it is pretty good.

- What extra features do you like to be incorporated into the tool?
  - p1: Perhaps the light diffuser could be permanently attached to the stage. It would be better if the computer could calibrate the size of the seed automatically.
  - p2: Have a ruler or measure to compare size of seed. Be able to not have as much zoom on seed.
  - p3: Being able to hold a seed in a certain position
  - p4: None
  - p5: None
  - p6: Can not think of anything right now.

A.4 Throughput

Figure A.4 shows the throughput for both two conditions.

A.5 Recognition rate

Figure A.5 shows the recognition rate for both two conditions.

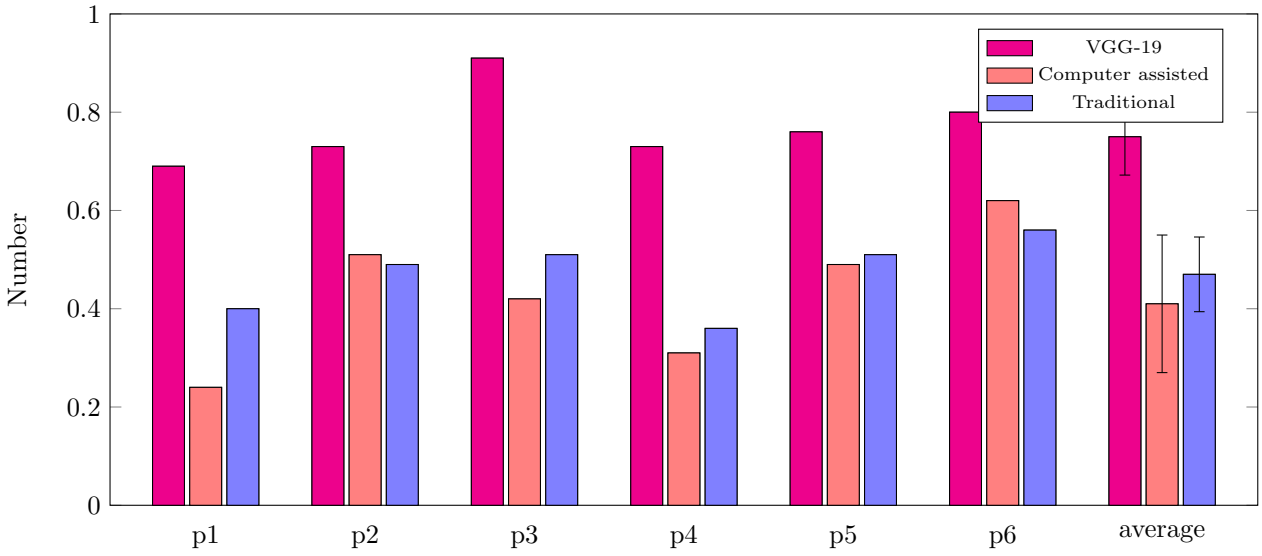


Figure A.5: Recognition rate for each participant. For computer assisted and VGG-19, if the correct result is in the top-3 candidates, it is counted as correctly identified.