COPING WITH LIMITED DATA: MACHINE-LEARNING-BASED IMAGE UNDERSTANDING APPLICATIONS TO FASHION AND INKJET IMAGERY

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Zhi Li

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

December 2019

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Jan Allebach, Chair
School of Electrical and Computer Engineering, Purdue University
Dr. Zygmunt Pizlo
Department of Cognitive Sciences, University of California-Irvine
Dr. Amy Reibman
School of Electrical and Computer Engineering, Purdue University
Dr. Fengqing Zhu
School of Electrical and Computer Engineering, Purdue University

Approved by:

Dr. Dimitrios Peroulis
Head of the School of Electrical and Computer Engineering

To the people who helped me grow:
the lovely folks who raised me, Li Yuedong, Zhou Hong, and my vast extended family,
my circles of friends who lift me up and keep me grounded.

To all the hardships and difficult times that I had in the past that strengthened my character, broadened my horizon, and taught me to enjoy every second of my life.

ACKNOWLEDGMENTS

First and foremost, I would like to extend my gratitude to my individuals and organizations that sponsored me through my PhD program: Poshmark Inc. and MIME for sponsoring my research; Professor Carla Zoltowski and Purdue Vertically Integrated Projects (VIP) program for giving me the opportunity to serve as Teaching Assistant and let me shine; my family for supporting me while I was seeking my research interest; and ultimately, Professor Allebach for making all these things happen. These people trusted in me and allowed me to walk out of Purdue debt free.

I want to thank my advisor Professor Jan Allebach, and my other committee members, Professor Zygmunt Pizlo, Professor Amy Reibman, and Professor Fengqing Zhu for their insights on my work. Their critical and constructive ideas fueled my research and helped me find my true passion in my PhD program.

I would also like send my appreciation to all the scientists and engineers I have worked with during my two internships: Dr. Palghat Ramesh, Dr. Chuheng Liu, Dr. Thanh Ha, and other excellent people at Xerox Advanced Technology Group and Midea Emerging Technology Center. Their professionalism has become the goal for my future career, and it is their inspiring discussions with me on image processing and machine learning that profoundly changed my way to approach different problems.

I also want to express my immense gratitude to my family for their wise counsel, sympathetic ear, and always open arm. They are the rock of my life, and helped me go through the toughest time, especially when my forever-loved grand fathers, Zhou Yongwang, and Li Qin, passed away.

In addition, I am very grateful to have many good friends around me. My friends at University Choir like Ben Coomer, Kendal Norman, and Calvin Yau, always comforted me through music and healed me using their contagious smiles and warm hugs. Guys and gals from Eta Kappa Nu Beta Chapter gave me a sense of belonging in

the sea of ECE graduate students and let me find myself through community and volunteer events. I also want to thank all my previous roommates, Yang Xiaozhao, Cheng Yang, Meng Changping, Ding Ziyun, and Yang Kai, for giving me a comfy and supportive harbor I could call home while thousands miles away from my hometown. And there are so many more friends that I want to thank. I wouldn't my friendships with them for the world.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

GLOSSARY

AGCES	Autonomous Garment Color Extraction System
B2C	Business to Customer
BOW	Bag Of Words
CNN	Convolutional Neural Network
CNS	Color Naming System
CRF	Conditional Random Field
FCN	Fully Convolutional Network
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
LBP	Local Binary Pattern
LSH	Locality Sensitive Hashing
P2P	Peer to Peer
RF	Random Forest
SLIC	Simple Linear Iterative Clustering
SVM	Support Vector Machine

ABSTRACT

Li, Z. Ph.D., Purdue University, December 2019. Coping with Limited Data: Machine-Learning-Based Image Understanding Applications to Fashion and Inkjet Imagery. Major Professor: Jan P. Allebach (School of Electrical Computer Engineering).

Machine learning has been revolutionizing our approach to image understanding problems. However, due to the unique nature of the problem, finding suitable data or learning from limited data properly is a constant challenge. In this work, we focus on building machine learning pipelines for fashion and inkjet image analysis with limited data.

We first look into the dire issue of missing and incorrect information on online fashion marketplace. Unlike professional online fashion retailers, sellers on P2P marketplaces tend not to provide correct color categorical information, which is pivotal for fashion shopping. Therefore, to assist users to provide correct color information, we aim to build an image understanding pipeline that can extract garment region in the fashion image and match the color of the fashion item to a pre-defined color categories on the fashion marketplace. To cope with the challenges of lack of suitable data, we propose an autonomous garment color extraction system that uses both clustering and semantic segmentation algorithm to extract the identify fashion garments in the image. In addition, a psychophysical experiment is designed to collect human subjects' color naming schema, and a random forest classifier is trained to given close prediction of color label for the fashion item. Our system is able to perform pixel level segmentation on fashion product portraits and parse human body parts and various fashion categories with human presence.

We also develop an inkjet printing analysis pipeline using pre-trained neural network. Our pipeline is able to learn to perceive print quality, namely high frequency noise level, of the test targets, without intense training. Our research also suggests that in spite of being trained on large scale dataset for object recognition, features generated from neural networks reacts to textural component of the image without any localized features. In addition, we expand our pipeline to printer forensics, and the pipeline is able to identify the printer model by examining the inkjet dot pattern at a microscopic level. Overall, the data-driven computer vision approach presents great value and potential to improve future inkjet imaging technology, even when the data source is limited.

1. INTRODUCTION

Image understanding problem aims to extract information of objects or understand the scene of the image. Thanks to the recent development of consumer camera and speciality camera, image understanding problem has been drastically expanded to a much broader spectrum.

On the one hand, some research efforts focus on image semantic understanding, that is to extract characteristics or information from the semantic objects in real world images. This has been one of the most active research fields in the computer vision disciplinary, covering topics from autonomous driving to cancer cell identification.

Another school of image understanding research is focused on analyzing and improving imaging technologies by studying the configuration and characteristics of imaging devices. One of the most interesting developments is using image understanding to improve cell phone camera image quality. For example, equipped with the capability of analyzing the scene, modern Image Signal Processing (ISP) pipeline can reduce noise under low-light shooting and create bokeh effect for portrait photos. And this is just a facet of how imaging engineers are rethinking and revolutionizing traditional ISP across the modern imaging industry.

Thanks to the recent development on neural network and large scale image datasets like ImageNet [1] and COCO [2], neural networks have become a very prominent solution for image understanding problems for its supreme accuracy and great compatibility for hardware acceleration. Fig. 1.1 shows the ground breaking performance improvement on image classification and localization led by AlexNet [3] in 2012. Since then, an astronomical amount of interest has been invested into developing neural-network-based image understanding research.

It is acknowledged that large amount of training data plays a crucial role in building machine learning or deep learning models. Large dataset usually contains larger

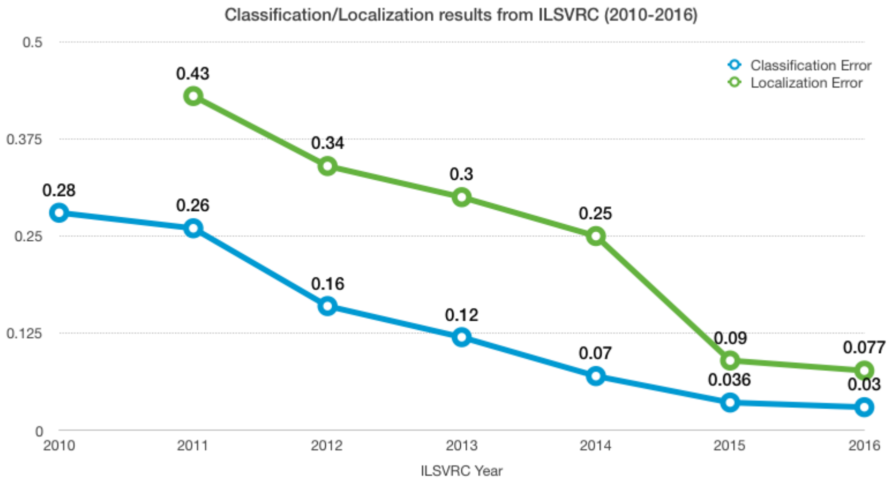


Fig. 1.1.: The evolution of best result in ImageNet Large Scale Visual Recognition Challenge (ILSVRC) from 2011 to 2012 [4]. 2012 winning submission, AlexNet, improves the classification error by 10%.

variation and diversity within the dataset, encouraging the model to improve generalization capability. Smaller datasets also tend to be imbalanced, meaning certain classes are much less represented. This will allow the model to be ”utilitarian”, omitting the importance of certain under-represented classes in favor of most frequent class.

However, for a particular image understanding problem, it can be hard to find and build suitable datasets in such a large scale. For instance, medical imaging datasets often only contains hundreds or even dozens of images due to the privacy issues and the cost of annotation collection. It is critical to study machine learning models with only limited data provided.

In this work, we aim to build image understanding algorithms for fashion and inkjet imagery with limited image data. In Chapter 2, we present the problem of identifying the color of the garment in online fashion images for online fashion marketplace, and Autonomous Garment Color Extraction System is proposed. Section

2.1 illustrate the importance of understanding the color in a fashion image, and related research fields are reviewed in Section 2.2. We specifically highlight the fact the difference between garment parsing (not necessarily having human body involved) and human parsing (with human body involved) and the reason a less data-driven method is desired. To deal with garment parsing without human body, in Section 2.3 and Section 2.4, our traditional image processing solution uses manually engineered features and a novel clustering method to extract the garment pixels from fashion images. As a compliment to the traditional image processing method, in Section 2.5, we adapt Multiple Human Parsing dataset, and use semantic segmentation networks to identify different fashion pieces utilizing the human body cues.

To extract the garment color and map the color into a verbal color description, in Section 2.6, we study existing color naming theories, especially fashion color naming theory. We conclude that, compared with traditional color naming method, a machine-learning-based color naming method is better at adapting the fast changing nature of fashion naming trends and capturing the subtle difference between some of the fashion color names. In this section, we design and conduct a psychophysical experiment called *Reversed Color Naming Experiment* to collect how human subjects associate color names with color appearances. Then, we build a random forest classifier trained on our collected data to map a CIEL*a*b* color coordinate to predefined color name. Finally, we present our pipeline integration and results in Section 2.8.

Then, in Chapter 3, we are showing another approach to leverage the power of neural networks without intensive training for image understanding tasks. In this chapter, we are aiming to build image processing pipelines to analyze Inkjet imaging systems. Our objectives are:

- To characterize the image quality of inkjet prints, specifically graininess noise in the inkjet printing.

- To develop a inkjet printer forensic system that can predict the source printer by looking at the printed dot patterns in the microscopic level.

In Section 3.1, we illustrate the importance of printer forensics, as well as developing new image analysis pipeline for printing quality analysis. Next, Section 3.2 revisits recent development of neural networks on image processing, particularly on image quality and printer forensics. The next two sections 3.3 and 3.4 dive into each project respectively and explain how we use neural networks to do image recognition without training the network. Our evaluation shows robust performance on for both forensics and image quality tasks.

In the end, we readdress the problem proposed in our work – the challenges of build image processing algorithms with machine learning where data is limited. Chapter 4 summarizes different approaches developed in our work, and our contributions and conclusions are drawn in our closing statement.

2. AUTONOMOUS GARMENT COLOR EXTRACTION SYSTEM USING MACHINE LEARNING

2.1 Background Overview

Online shopping is an exponentially growing market. Retail sales world-wide, including both in-store and inter- net purchases, totaled approximately $22.5 trillion in 2014, with $1.316 trillion of sales occurring online. By 2018, ecommerce retail spending is projected to increase to nearly $2.5 trillion[1] [5]. Online marketplace not only has changed our lives with its convenience and huge diversity of products and services, but also cultivated a number of e-commerce giants. However, in recent times, public attention has switched to Peer-to-Peer (P2P) business model from traditional Business-to-Customer (B2C).

The P2P online marketplace allows its users to post and sell their items to other users on the platform, and P2P economy has been growing rapidly since PayPals acquisition by online retailer eBay in 2002. However, the traditional P2P e-commerce marketplace also became the disreputable nest of Internet fraud because the sellers were usually anonymous, the information is incomplete and messy as listed in table 2.1. Thankfully, modern P2P marketplace has evolved into a much advanced and specialized form: first, specialized marketplaces on particular markets like Airbnb (a service that allows users share the surplus space in their properties with travelers for profit), Uber (a ride sharing service), or Poshmark (a fashion marketplace that users can trade clothes and other fashion items) are thriving; second, by connecting to social networks, users can obtain a better knowledge with each other via checking each other's social profiles; third, these modern P2P websites are more positioned to be

[1] `http://www.emarketer.com/Article/Retail-Sales-Worldwide-Will-Top-22-Trillion-This-Year/1011765`

everyday users' selling platform with lower entering barrier by developing easy-to-use mobile applications.

Despite the fact that improvement and progress has been made to the P2P economy, most P2P e-commerce websites only maintain the web services and provide marginal services to the users. This creates a more friendly and easy-to-use atmosphere for both shoppers and sellers, leading to a more dynamic shopping experience. However, it also creates some problems due to the lack of management and details. Some of the information provided by average sellers is not correct, and some products are missing some very important information. This type of problems often cause miscommunication or publicity issues, leading to a slower sale performance or worse shopping experience.

Hence, an Artificial Intelligence (AI)-based shopping/marketing assistant is highly anticipated by the sellers and the service providers. This type of the assistant should be developed to

1. identify the item the sellers are selling based on the users' input
2. capture and understand the features of the listed item
3. fill out the missing information automatically from given information
4. correct the users' potential mistake if necessary
5. improve the quality of the users' listings if possible

This chapter provides an overview of the project achieved and proposed by this research. Section 2.1.1 takes a closer look at the how color information is managed on fashion online marketplace, and we analyze some of case where the color misinformation occurs. Our accomplished work to solve this problem is highlighted in part 2.1.2.

Table 2.1.: Comparison on information organization on between B2C and P2P

Model	B2C	P2P
Visuals	• Shot in professional settings • Professional camera deployed • Models and mannequins • Balanced and ample lighting	• Shot in household • Most commonly cell phone camera • Hangers, self-modeled or none • Various lighting
Texts	• Accurate categorization • Specific details are provided • less missing fields	• Wrong categorization • Less details and vague descriptions • More missing fields
Management and Moderation	• Well managed and updated by professional teams. Regulations and restrictions are enforced. • Listings only include sale related information.	• Managed by sellers, minimum interference from the website • Listings may include informations for other purposes: networking/self promoting, sale events, and other advertisement or even spam.

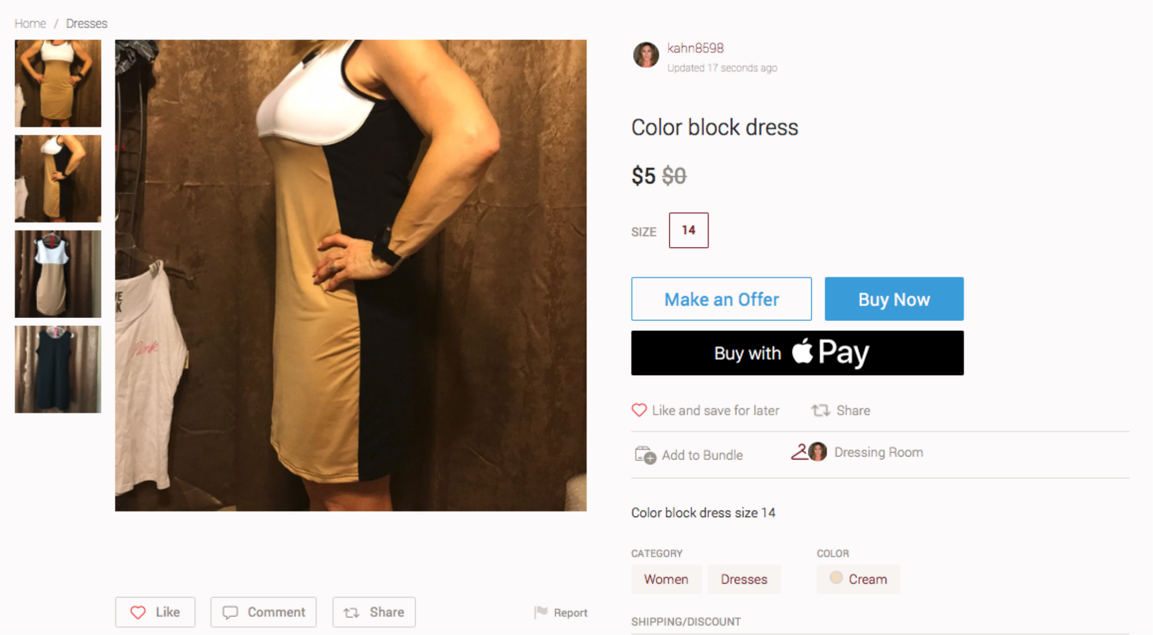
2.1.1 Color Management on Fashion P2P Marketplace

As stated in the introduction, information organization has been a critical challenge for the online marketplace. Unfortunately, for fashion market, these problems on information organization also persist. For example, as one of the most defining features of a fashion product, color information on P2P websites is not always provided by the sellers, and sometimes the color information provided by the sellers can be inaccurate.

- **Incomplete or inaccurate user input.** Incomplete user input includes two kinds: first, sellers might simply forget or omit the part of inputing item's color(s). Another scenario is that sellers fail to report the full spectrum of the color. An example is given in fig. 2.1, in which only one of three colors is labeled by the user. There are also some cases where the sellers provide the wrong color. Inaccurate input is less likely to happen compared with incomplete input, but both of these two cases are frequent enough to influence overall site searching accuracy hence to effect the selling/shopping experience.

- **Photography color inconsistency.** This means that the color shown in the images does not agree with the color information provided by the seller, or the real colors of the item. Fig. 2.2 shows an example of color inconsistency. This can be caused by undesirable lighting condition, the device's color reproduction capabilities, or additional photo post-processing done by the user. Note that some of the colors have very similar appearance, for example, black v.s. navy blue, pink v.s. cream white, and gold v.s. yellow. In this case, it is hard for human viewers to distinguish the fine difference of the colors by looking at a single color patch. A reference color is usually recommended to get more accurate color perception.

Hence, there are several factors that come to our attention to improve the quality of the color information. First of all, a more user-friendly is hard to control the



Home / Dresses

kahn8598
Updated 17 seconds ago

Color block dress

$5 ~~$0~~

SIZE 14

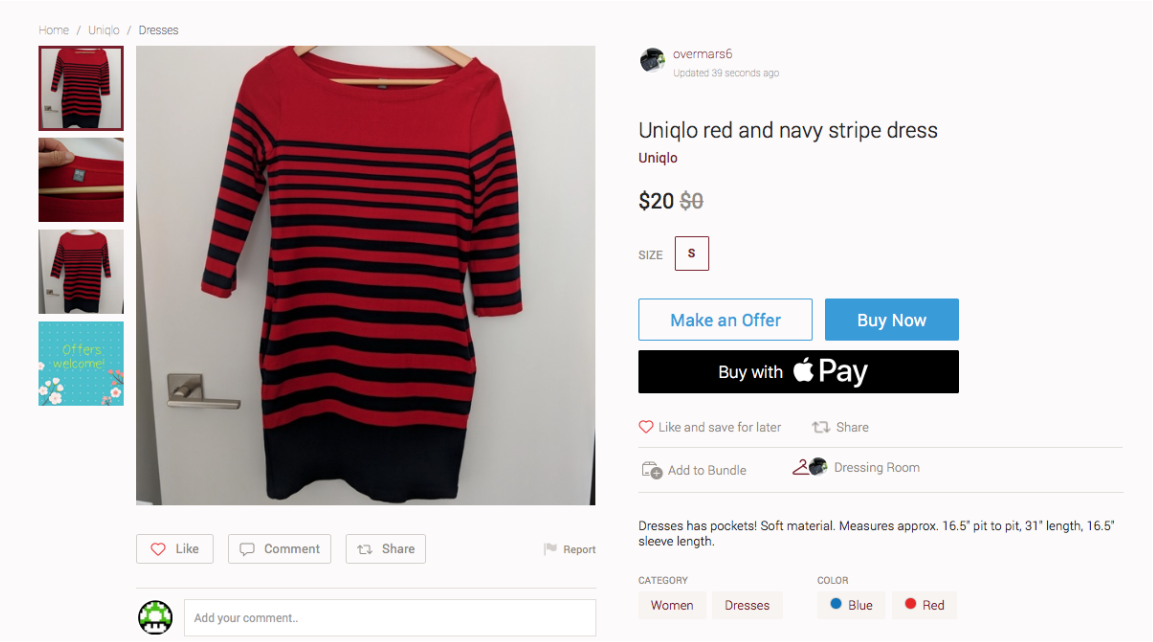
Color block dress size 14

CATEGORY: Women, Dresses

COLOR: ● Cream

SHIPPING/DISCOUNT

Fig. 2.1.: Example of the case of incomplete color input. In this listing, based on the images provided by the user, the colors input should be white, black, and cream. However, only cream color is marked. Source:Poshmark.com.



overmars6

Updated 39 seconds ago

Uniqlo red and navy stripe dress

Uniqlo

$20 ~~$0~~

SIZE S

Dresses has pockets! Soft material. Measures approx. 16.5" pit to pit, 31" length, 16.5" sleeve length.

CATEGORY: Women, Dresses

COLOR: ● Blue, ● Red

Fig. 2.2.: Example of the case of photography color inconsistency. In this listing, the seller (overmars6) is selling a dress of red and blue stripes, however due to various reasons, the blue strips appear closer to black. Source:Poshmark.com.

quality of the fashion photography. Previously we have done research to develop an SVM based aesthetic quality predictor to assess the aesthetic quality of an image.

Therefore, we would like to design a fashion image understanding system that can autonomously identify the garment region from a fashion image and extract the garment color. This result can be used to fill in some missing information on the website, and also allows user to reflect how accurate their photos and information are.

2.1.2 Accomplished Research Works

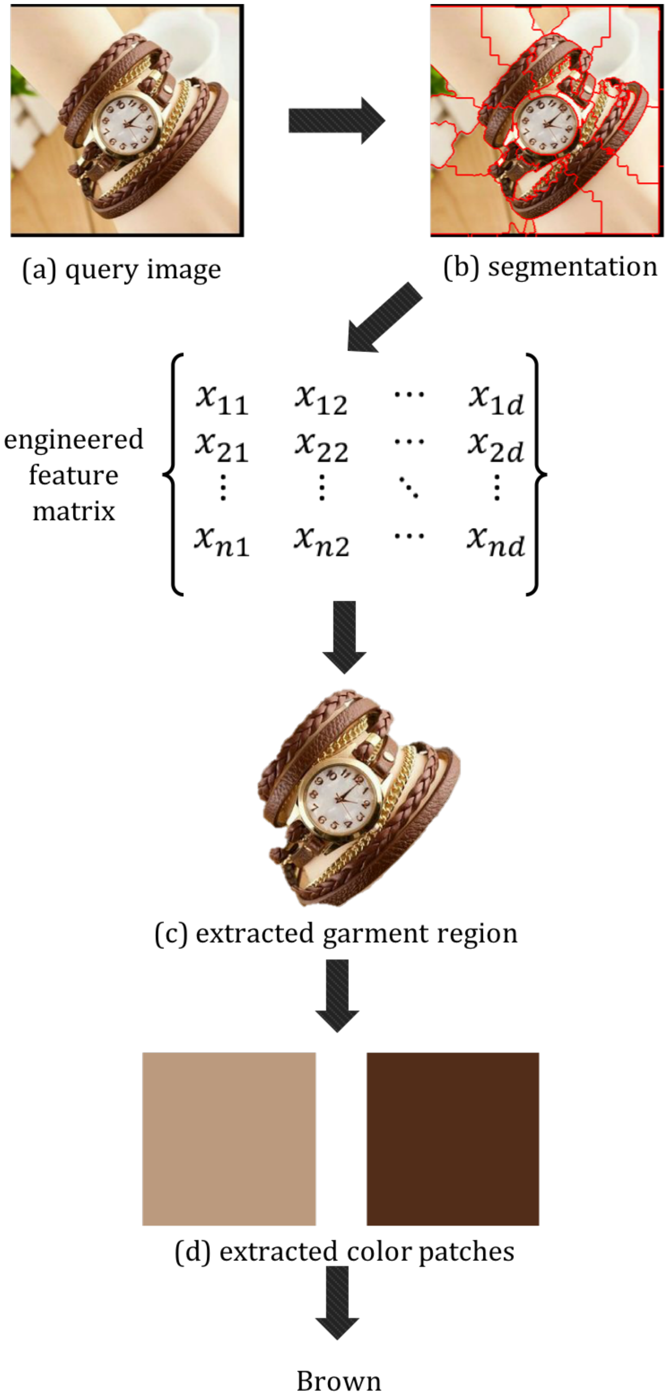
We propose the Autonomous Garment Color Extraction System to extract color from the fashion portrait image, shown in Fig.2.3. Our proposed structure has so far achieved the following contributions:

- **A novel semantic segmentation structure with pixel-level accuracy based on unsupervised learning methods.**
  Heuristically, supervised machine learning method is capable of producing more accurate semantic segmentation results. With recent exponential growth of deep neural network theory, the segmentation result on generic images produced by these new classifiers has been further improved. However, one of the major drawbacks of most of supervised learning methods is that a large dataset is desirable as it tends to improve the classification performance. This is especially true in the case of neural networks, where a huge amount of labeled ground truth is required. The pros and cons of these method will be further discussed in Section 2.2. For our research on images on fashion marketplace, we study and discover some of the common traits that are shared by these studied images. Therefore, a simpler yet effective structure aimed to fashion product images is built based on the features at the mean time to reduce the requirement of labeled training data. We believe that despite of the unignorable success of the recent applications of the deep neural network, it is possible to use computer vision

based features to develop a classifier with certain types of semantic segmentation performance.

- **Study the fashion color naming system and create a computational color model to extract the color.**

  In this system, we use the Gaussian Mixture Model to extract the dominate color in the garment region, and matching the garment color with a color description. As discussed in detail in the Section 2.2, most of the previous works, that are related to assign a descriptive phrase to a certain color coordinate, are focused on how to name a color in a systematic way with universal dictionary. Specifically for color naming in fashion domain, few work has been done to look into the process that maps color coordinates to words, as most of the fashion retailing business names the color of the product before releasing the products to the market. Therefore, our research finds some of the potential problems in the P2P online fashion marketplaces, and provides a preliminary color categorizing system based on different natures of the colors.



(a) query image

(b) segmentation

(c) extracted garment region

(d) extracted color patches

engineered feature matrix
x_{11} x_{12} ⋯ x_{1d}
x_{21} x_{22} ⋯ x_{2d}
x_{n1} x_{n2} ⋯ x_{nd}
Brown

Fig. 2.3.: An overview of the Autonomous Garment Color Extraction System (AGCES). The system utilizes the segmentation, and use engineered features to select garment region. The final segmentation mask is used to find the color of the item.

2.2 Related Work

As stated in the previous section, in order to extract the color from the garment in the image, we need to find the garment region in the image first. This task falls into the category of semantic segmentation, where each pixel of an image is labeled with a physical meaning corresponding an object class. After we retrieve the pixel values of the garment region, we will be able to use color matching algorithm to assign the correct label to the color, hence name the color of the garment. In this chapter, we provides overview on previous works on related research fields, namely image semantic segmentation, fashion image understanding and color naming analysis.

2.2.1 Semantic Segmentation

Semantic segmentation, also known as *scene parsing*, is one of the most studied image understanding problems. Unlike unsupervised image segmentation, which aims to partition an image into coherent small regions according to the low-level cues including color or other proximity [6], semantic segmentation requires the algorithm to understand the image, and classify each pixel of an image into one of the several predefined object classes [7], and has been studied and highlighted for long time due to the fact that more applications nourish from inferring knowledge from imagery [8], including autonomous driving [9–11], medical imaging [12] and more.

Prior to the recent exploding computer vision applications of Convolution Neural Networks (CNNs), the most common approach to the semantic segmentation problem was to use Conditional Random Fields (CRFs) [7, 13]. Shotton et al. [14] designed a texture-layout filter and proposed a discriminative model based on conditional random field to capture the spatial interactions between labels. Schroff et al. [15] introduced a pipeline that uses Random Forests on class based pixel-wise segmentation by combining spatial context and discriminative classifiers. Verbeek et al. [16] introduced a method based on learning CRFs considering both local features and contextual features from datasets that are not fully labeled. However, these traditional computer vision scene parsers are limited by the structure of the classifiers, and it is difficult to engineer capable image features to work on all the scenes.

CNNs and recent high-performance computing system development enable image recognition research to achieve new state-of-art performance. Krizhevsky et al. [3] train a ImageNet network with 1.2 million images to do classification for 1000 different classes. Simonyan et al. [17] investigate the performance of the network regarding various network depths, hence improve the performance on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [18]. Szegedy et al. [19] developed a deeper and wider network GoogLeNet, further improved the ILSVRC14 classification result. These recent success proves that neural networks are capable of recognizing comprehensive scenes, but it would not produce pixel-wise semantic labeling result.

Long et al. [20] re-structured the Convolutional Network to direct, dense prediction of semantic segmentation. This Fully Convolutional Network (FCN) structure provides an end-to-end learning solution to semantic segmentation problems. However, like any other supervised learning methods, these neural network based methods are limited by the ground truth and the predefined class. Although most of the popular datasets include many instances of humans, the datasets that include different clothing information are relatively rare. In next section, more research on fashion image understanding and fashion image segmentation is briefly examined.

2.2.2 Fashion Image Understanding

Fashion understanding has been a very popular topic recently, not only because of the growing fashion industry that is estimated to be worth 2.5 trillion dollars in next four or five years [21], but also because clothing understanding can be an important clue of many human-centric research and analysis. Some of the current advanced methods include clothing semantic attribute summarization [22–25], fashion landmark detection [21,26] and trending discovery [27–29]. And one of the most active topics is clothing retrieval.

Recognizing clothing is a challenging problem because of the wide variation of clothing appearance, layering, style and interaction with human body. However, some progress has been made to pursue the goal. Some recent practices combine visual cues from human body detection and fashion landmark localization. A popular approach is to train a bounding box detector to find the garments in the image [5, 30]. Fu at el. [31] introduced a pipeline that uses the human parts as semantic parts, and encodes semantic context into the Bag-Of-Words (BOW) model to label pixels of the garment. Kalantidis et al. [32] utilized the pose estimation, and segmented and classified the garments based on the pixel position related to the human body using a multi-probe locality-sensitive hashing (LSH) index.

In addition, for pixel-wise clothing parsing, Yamaguchi et al. [33] developed a clothing parser based on pose estimation and CRF based on the Fashionista dataset. More recently, Yamaguchi et al. [34] proposed a framework to produce pixel-wise labeling by analyzing images that are similar to the query image.

One of the challenges is that there is not a sufficiently generic public fashion dataset available for fashion semantic segmentation. Most of the datasets are similar: street shots with full or part of human body. Xiao et al. [35] developed Fashion-MNIST data set that consisting of more than 60,000 28x28 grayscale images. Each image has a label from 10 classes including T-shirt, Trouser, Pullover, etc. More comprehensive, Liu et al. [25,26] developed a dataset that contains more than 800,000 real-life fashion images mostly for bounding box and fashion landmark localization. Fashionista dataset [33] includes 158,235 photos and labeled pixelwise by human subjects. However, this dataset only has images with full human body modeling the garments.

2.2.3 Color Naming System

Many computer vision research and applications have been done using color information of the image or the video, yet, it is still under development the ability or complete theories to name individual colors, pinpoint objects of specific colors, and communicate the impression of a certain color composition [36]. Color interpretation is highly convoluted with the image content, and other factors like human perception, culture and linguistics. Therefore a flexible computational model for color categorization is desired. Some of challenges unaddressed include:

1. The critical problem of color categorization is the definition of the basic colors that are considered "most representative". These colors are prototypical colors, and they are the foundation of the different color categories.

2. Another open issue of building a color categorization system is generalization from the prototypical ones to non-prototypical ones.

3. There are also difficulties capturing color appearance from a complex scene or interpreting from human viewers' description.

As stated above, color perception and description is a multi-disciplinary cross-cultural research topic. Linguists and Psychologists have made many efforts to understand the mechanism of color naming. Berlin and Kay [37] investigated 20 languages by experiments and another 78 through literature review, and defined 11 basic terms in English: *black, white, red, green, yellow, blue, brown, pink, orange, purple, and gray.* Berlin and Kay also discovered that human subjects perform better at pick "representative example" for each of the color terms than in forming clear decision boundaries between these terms. This theory enables the definition of *focal colors*, which represent the centers of color categories, and the hypothesis of graded (*fuzzy*) classification [36]. Based on the fuzzy membership theory, Kay and McDaniel's model [38] defined the color categorization based on fuzzy set theory by giving a certain color a degree (from 0 to 1) of belonging a certain set. This means that the focal color of each color set has the degree of 1 of belonging its corresponding color set, and other non-focal colors' membership degrees decrease according to the distance from the focal color in some color space. However this model only considered four sets of colors (*red, green, yellow, and blue*), and it does not include the solution on nonspectral basic color such as brown, pink and gray. Despite the fact that the universally proper and accurate wording is still an ongoing research topics in linguistics and psychology, in real life situation there are some color naming and notation standards implemented. In this section, we first look into general Color Naming Systems (CNSs) and color describing dictionaries, and then we review how color information has been managed in fashion industry and related research.

Color Naming Standard

Although many well defined numerical color spaces have been developed, and these methods have been proved to be effective in terms of most of color-related image processing and computer graphics tasks, in everyday life the most common way to communicate about color is through verbal description. Thus Color Naming System is desired to transform information from color spaces to color names. So far there are several proposed work mapping color coordinates to verbal description. One of the most commonly used color naming system is Munsell color standard [39]. Munsell Color System was developed in the late 1800s by Albert Henry Munsell[2]. It has been widely used in paint and textile production [36]. However, this proposed system is lack of the color vocabulary and exact transform from any color space to Munsell. Miyahara [40] improved the Munsell system by proposing a transform from $CIEXYZ$ to Munsell system, however it is not always accurate in certain regions in $CIEXYZ$. To expand the vocabulary of the color naming system, Maerz and Paul [A Dictionary of Color] developed the first version of color naming dictionary including 3000 English words and phrases, and later, The National Bureau of Standards published a more detailed dictionary including 7500 different names that has been used in specific fields such as biology, textile, dyes and paint industry. However, both dictionaries are not organized in a systematic way, which makes the generalization more difficult. To

[2] `http://munsell.com/color-blog/munsell-color-order-system-what-is-it-and-how-is-it-used/`

address this issue, NBS developed ISCC-NBS dictionary of color names according to the recommendation of Inter-Society Council. ISCC-NBS includes color names for 267 regions, and all the terms are sorted by the dimension of the color space: hue, lightness and saturation. This system divides the lightness into 5 levels (*very dark*, *dark*, *medium*, *light* and *very light*), and 4 levels for saturation (*grayish*, *moderate*, *strong*, and *vivid*), and three terms that consider both lightness and saturation (*brilliant*, *pale*, and *deep*). ISCC-NBS also expands to 28 basic sets (*red*, *orange*, *yellow*, *green*, *blue*, *violet*, *purple*, *pink*, *brown*, *brown*, *olive*,*black*, *white*, and *gray*).

Some other alternative systems have been proposed as well. An extension of the CNS model is color naming method (CNM). This was originally proposed by Tominaga [41]. This method utilizes a predefined set of color names in the Munsell color space, and proposes a method to map pixel values to specific color names using coptical measurement system. The color names in this method are specified at one of four accuracy levels (fundamental, gross, medium, and minute) so that names from the higher accuracy level correspond to smaller color regions in the Munsell space [36]. However, there are some drawbacks: nonstandard vocabulary used in the system increases the difficulties mapping to standard color names; the color space conversion is beyond the closely controlled setting in CNM. Lammens [42] proposed a color categorization system based on Gaussian normal distribution. Belpaeme [43] built a another color categorization framework based on the notion of color primitives surrounded by color regions with fuzzy boundaries and modeling via adaptive radial basis function networks [36]. Mojsilovic [36] studied the *National Bureau of Standards'* color recommendation for color names and developed a new set of vocabulary and syntax, and a new perceptually based color-naming metric was proposed to match an arbitrary input color to a color name.

[3] Photo credit: `https://www.policymap.com/2015/08/color-me-curious-why-policymap-maps-are-purple/`

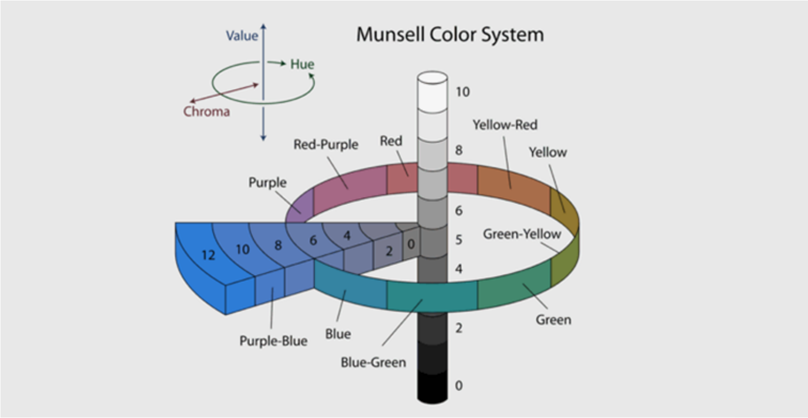


Fig. 2.4.: Munsells cylindrical color tree: the radius from the center to the peripheral represents chroma (saturation), the line in the center represents value (lightness) and the circle represents hue.[3]

Fashion Color Analysis

For color management in fashion industry, unlike other industries, where color names usually are given based on the color appearance, most of the fashion color naming are done by the manufacturers individually before products are released. Other than that, seasonal trending color names are also predefined by the team at the Pantone Color Institute in Pantone Fashion Color Trend Report [44] every season. However, these color naming practices do not use any standard color naming dictionary, only consider trending colors, and changes constantly by the season. Besides, these names are commonly used among fashion experts and high-end fashion brands, yet it is not a standard and universally recognized color naming system by all the fashion manufacturers.

However, there are several research topics has been conducted to do color mapping from color coordinates to color names by viewers. Most of the clothing attribute classification research we present in the previous section uses different sets of color. Deep Fashion dataset [25] uses 10 color set with *cyan* and *blue* covering different shades of blue. Bossard's dataset [22] has 13 color pre-defined with non=protypical color *teal* and *beige*.

2.3 System Overview

2.3.1 Bound the Problem

When it comes to fashion image analysis, it comes as no surprise that there is a huge variety of fashion images on the online fashion marketplace. Some of the images have very complex combinations of fashion items, many layers of clothings and other accompanying objects in the image. These images are usually designed and produced professionally or delicately, and they usually have higher aesthetic quality. However, for this type of images, as discussed in the previous section, traditional feature based machine learning is hard to achieve the semantic segmentation to identify different garments in pixel level accuracy. Therefore, in this case, only product portraits are considered in this research.

Product portraits are images that only feature the target items, in our case, the items that sellers want to sell. This allows the users to use some other items to "decorate" the image in a limited extent, as long as the item for sale is staged for the highlight.

Some examples of product portraits and its counterexample are given in fig.3.8. Our system is design to process images like (b) and (c).

2.3.2 System Overview

As motivated before, we propose a system that autonomously extracts the color of the featuring garment from a fashion image. As shown in fig.2.6, this autonomous color extraction system can be divided into three modules: the segmentation module, the grouping module, and the color processing module. Segmentation module takes the input image and cut into smaller segments that share similar characteristics; the grouping module uses the segmentation information, calculates the features of each segments, and groups them into two classes: garment and non-garment region. The last module is deployed to extract the color information, and to transfer numeric color

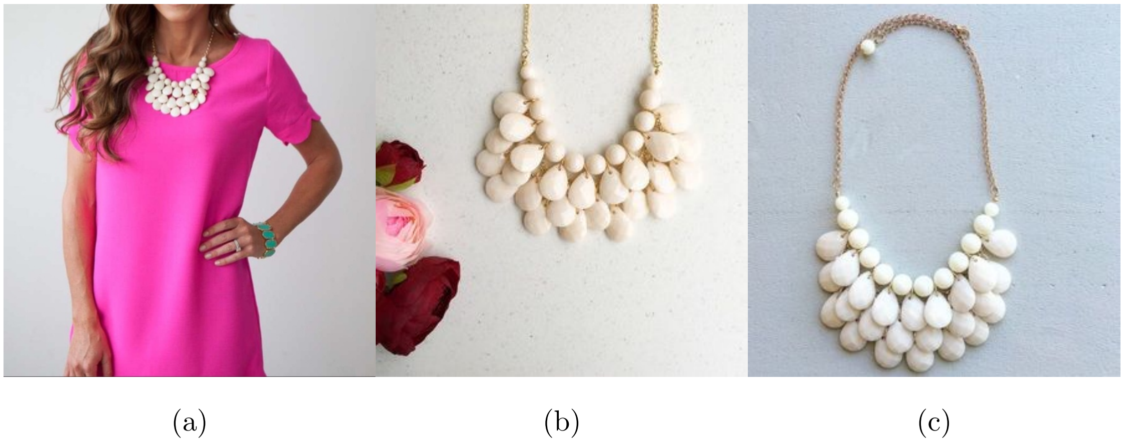

(a)
(b)
(c)

Fig. 2.5.: Source: Poshmark.com. This is a set of example images of fashion photo. All three images are posted by the same Poshmark seller in the same listing to promote the necklace, but in (a) the promoted item is overshadowed by the pink dress. Other two, (b) and (c), directly highlight the target item.

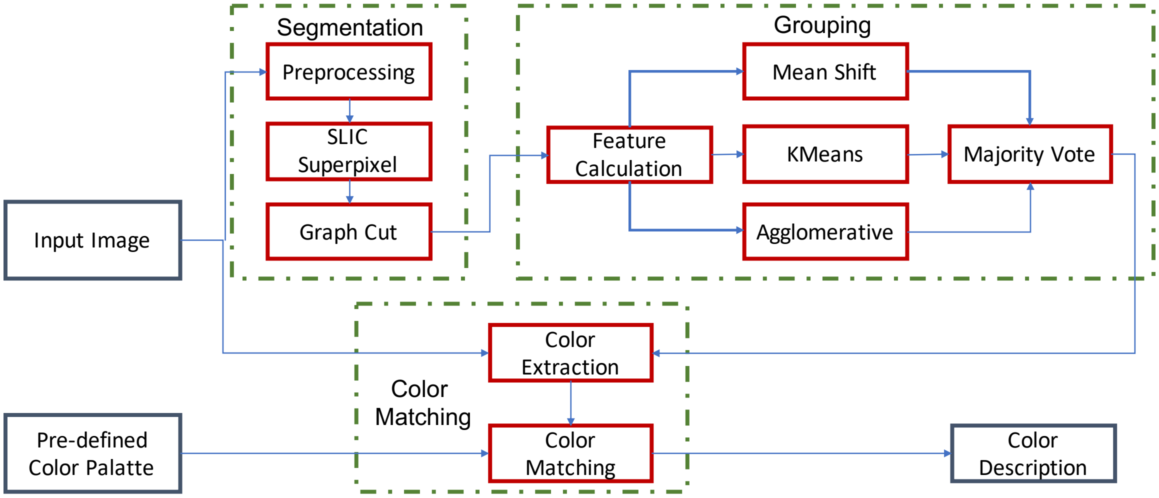


Fig. 2.6.: The pipeline of Autonomous Garment Color Extraction System (AGCES).

coordinates into descriptive words, such as ”blue”, ”pink”, or ”white”. All modules will be explained in the following sections.

2.4 Image Segmentation

2.4.1 Over Segmentation

In this module, we decompose a given image into smaller perceptually homogeneous segments. By doing this the computation and complexity of the algorithm is reduced. We propose the image segmentation structure that is prone to preserve edge in the image, and sensitive towards both color and texture.

We use SLIC superpixels [45] to abstract the image into perceptually uniform elements. SLIC algorithm calculates pixel perceptual distance in LABXY space, and deploy K-means to group similar pixels together as a superpixel. By combining both color and spatial information, SLIC superpixels are local, compact and edge aware. Because of the nature of the SLIC superpixels, the algorithm tend to render noisy superpixels where lines and edges congregate. Therefore, we adapt preprocessing steps to the image proposed by Wang et al to smooth the image before superpixel segmentation. Some result comparison is given later.

We further apply the graph-based image segmentation techniques to further group superpixels together. In this kind of problems, an image is represented as a graph $G = (\boldsymbol{V}, \boldsymbol{E})$, where each vertex $v \in \boldsymbol{V}$ is a superpixel within the image, and the edges in $\boldsymbol{E}$ is the similarity between two neighboring superpixels. A weight w is associated with each edge based on the certain properties or criteria.

In more generic situations, the graph representation usually considers color difference between two neighboring superpixels as the only aspect of the edge weight. For fashion images, however, with the goal of accentuating the different fabrics or materials, we update the weight using a new method developed by Wang at al.This method combines texture difference and color difference by calculating Local Binary Pattern (LBP) within 3x3 neighborhood and CIEΔE. The standard CIEΔE 1976 is used to calculate the color similarity measurement between average colors of two neighboring superpixels,

$$D_{color} = \sqrt{(L_1 - L_2)^2 + (a_1 - a_2)^2 + (b_1 - b_2)^2},$$

where $[L_1, a_1, b_1]$ and $[L_2, a_2, b_2]$ are two different colors. Next we need to determine the texture measurement, LBP.

The LBP an be calculated based on every pixel in each superpixel. For pixel p and its corresponding superpixel sp, if all the surrounding pixels of p are also in sp, do following:

1. Compare the L values of each surrounding pixel with central pixel in a clockwise order. If neighboring value is greater or equal to the central value, then output is 1, otherwise 0

2. Concatenate the binary results and form an 8-digit binary number

3. Convert the 8-digit binary number into a decimal number.

After getting all the decimal numbers of the eligible pixels within the superpixel, a histogram is sampled into 9 bins and normalized. The final vector of normalized occurrences from the histogram is the LBP vector.

After combining both texture and color measurements, we are able to produce the new edge weights, and hence deploy Normalized Cut to composite the graph into segments. The segmentation results and comparisons are given in fig.2.7.

In the picture of jeans, some noisy residual segments are rendered right below the folded part. And the preprocessing (shown in the second row) eliminates such noisy regions. However, there are also some other drawbacks from the smoothing. For example, for the picture of bra, the segmentation results without smoothing shows more fidelity towards the original object edges. As for the texture, In the picture of a lady with sunglasses, the texture of the background has been preserved thanks to LBP algorithm.

2.4.2 Segmentation Grouping

The group module is used to group all the segments into two groups by determining whether each segment should be in background or in the garment region. This can be further divided into two steps: to design discriminative features to differentiate garment and background as much as possible; to use the calculated features to make decision.

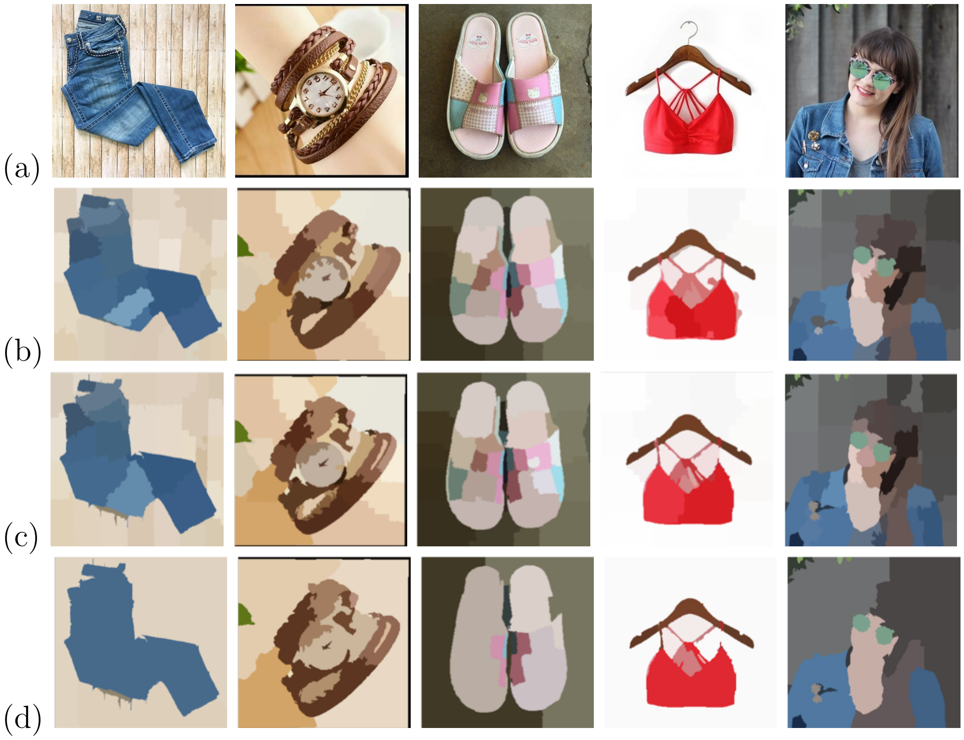


Fig. 2.7.: The segmentation results. Row (a) is the original image; row (b) is the images with preprocessing and LBP as texture cue; row (c) is the images without preprocessing; row (d) is images with regular RAG, without preprocessing and LBP as texture cue.

Feature Engineering

After we investigate many fashion product portraits, there are some features that can be generally applied to all fashion product portraits:

1. Fashion images, especially for product portraits, generally highlight the texture of the fabric or any other materials. Therefore, garment regions should have higher amount of high frequency details compared with non-garment regions.

2. Garment region usually also has greater contrast from the background region. []

3. Spatial information is also very important for product portraits, and it usually locates at a certain part of the image.

4. Color information and other low level information are very important as it can keep the classification results perceptually homogeneous.

Therefore, as shown in tab.2.2 we design four sets of measurements: Laplacian feature set, Perrazi contrast feature set, spatial feature set, and low level feature set. Among all four feature sets, the low level features are easy to obtain by averaging pixel values in CIEL*a*b space and XY space, respectively. Thus these two features will not be covered in full detail.

Laplacian Feature Set

Fashion product portraits are shot to present the fine details of the garment, therefore capturing constructive high-frequency details of the image can help determine the garment region.

Since the Laplacian pyramid can also catch image details at different scales, we adopt the Laplacian power summation features proposed by Wang at al [46], and a 2-level Laplacian pyramid is built and the pixels of Laplacian images are represented by their absolute values to focus on the detail strength. One example is shown in fig.3.1, and black or darker pixel mean the Laplacian difference at that position is low.

Table 2.2.: Feature Symbols

Feature Sets	Feature Names	Symbol
Laplacian	1st Order Laplacian	$\mathbb{L}^{(1)}$
	2nd Order Laplacian	$\mathbb{L}^{(2)}$
Perrazi	Color Uniqueness	$\mathbb{U}$
	Distribution	$\mathbb{D}$
Positioning	1st Boundary	$\mathbb{F}^{(1)}$
	2nd Boundary	$\mathbb{F}^{(2)}$
	Standard Deviation	$\mathbb{E}$
Low Level	Mean Color Values	–
	Centroid Position	–

First, we calculate the Laplacian power images $\mathbb{L}^{(1)}$ and $\mathbb{L}^{(2)}$ by generating first and second Laplacian pyramid layers of the gray scale image and taking the absolute value. For each pixel p, its corresponding Laplacian power in i-th layer $\mathbb{L}^{(i)}$ is $l_p^{(i)}$. Therefore for i-th segment, its k-th order Laplacian power $\mathbb{L}_i^{(k)}$ can be calculated by

$$\mathbb{L}_i^{(k)} = \frac{1}{|C_i^{(k)}|} \sum_{p \in C_i^{(k)}} l_p^{(k)}$$

Where $C_i^{(k)}$ is the set of pixels in k-th layer that belongs to segment i, and $|C_i^{(k)}|$ is the number of elements of this set. Note that for second layer, we also down sample the segmentation map to match the Laplacian image.

Results of Laplacian features for for some images are shown in fig.2.9 column (c) and column (d) respectively.

Perrazi Feature Set

It is also intuitive that for fashion product portraits, the garment should be highlighted in a situation where great contrast is created.

Perrazi et al. [47] proposed an algorithm that produces image saliency maps by evaluating color contrast and distribution contrast. However, the perrazi contrast is calculated in a superpixel level, therefore transform from superpixel level features to segment level features is required. First, for i-th super pixel, the color uniqueness U_i can be calculated by

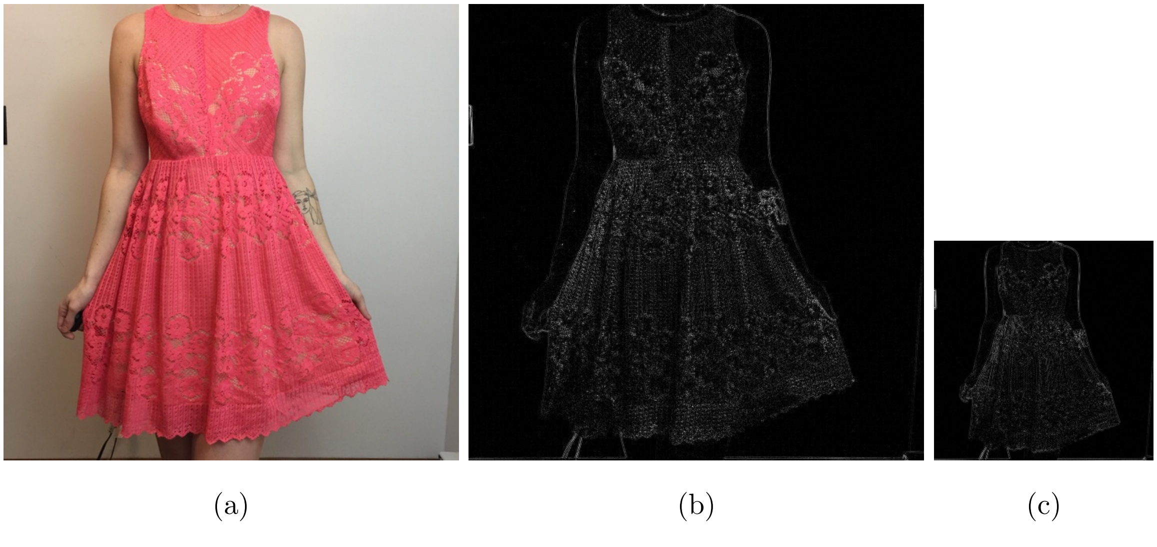
$$U_i = \sum_{j=1}^{N_s} w_{i,j}^{\mathbf{p}} |\mathbf{c}_i - \mathbf{c}_j|^2$$

where N_s is the number of SLIC superpixels in the image, $w_{i,j}^{\mathbf{p}}$ is the Gaussian weight related to spatial correlation between superpixel i and j. $\mathbf{c}_i$ is the mean color coordinates of the superpixel i.

The distribution of superpixel i is given

$$D_i = \sum_{j=1}^{N_s} w_{i,j}^{\mathbf{c}} |\mathbf{p}_i - \mathbf{p}_j|^2$$

where $w_{i,j}^{\mathbf{c}}$ is Gaussian weight related to color correlation between two superpixels. Both $w_{i,j}^{\mathbf{p}}$ and $w_{i,j}^{\mathbf{c}}$ are to yield local contrast term, which bring more sensitivity to similar color-wise or position-wise superpixels, respectively. These terms are given as



(a) (b) (c)

Fig. 2.8.: Visualization of Laplacian pyramid first 2 layers. a is the original image, and b is the 1st layer of Laplacian pyramid. Likewise c is the 2nd layer.

$$w_{i,j}^{\mathbf{p}} = \frac{1}{Z_j} \exp\left(-\frac{|\mathbf{p}_i - \mathbf{p}_j|^2}{2\sigma_p^2}\right)$$

and

$$w_{i,j}^{\mathbf{c}} = \frac{1}{Z_j'} \exp\left(-\frac{|\mathbf{c}_i - \mathbf{c}_j|^2}{2\sigma_c^2}\right)$$

Note that Z and Z' are scalars for the purpose of normalization, which means $\sum_{j=1}^{N_s} w_{i,j}^{(\mathbf{p})} = 1$ and $\sum_{j=1}^{N_s} w_{i,j}^{(\mathbf{c})} = 1$. Heuristically parameter $\sigma_p = 0.25$, and $\sigma_c = 20$.

Once the superpixel-level color uniqueness and distribution are obtained, we compute the segment-level features. For segment i and its corresponding set of superpixels C,

$$\mathbb{U}_i = \sum_{s \in C} \frac{m_s}{M_i} U_s$$

and

$$\mathbb{D}_i = \sum_{s \in C} \frac{m_s}{M_i} D_s$$

where m_s is the number of pixels in superpixel s, and M_i is the total number of pixels in segment i. The sample results for color uniqueness and distribution features are shown in fig.2.9 column (e) and (f).

Positioning Feature Set

The first feature that we would like to look at is how close the segment is to the image boundary. Although it is not a hard line, the garment region tends to locate at the center of the image, or at least on the Rule Of Third guidelines. Therefore a frame or boundary feature is developed to show how many pixels of a given segment are close to the borders. For a segment i, a frame feature $\mathbb{F}_i$ can be described as

$$\mathbb{F}_i = 1 - \frac{|F_i|}{M_i}$$

where F_i is the set of frame pixels in the segment i, and M_i is the total number of the pixel in segment i. When $\mathbb{F}_i \to 0$, more pixels inside of segment i are on the boundary, and it is less unlikely to be a garment segment; when $\mathbb{F}_i \to 1$, fewer pixels are on the boundary, and the segment is more likely to be in the garment region. However, the question unanswered is how to define the boundary pixel.

We use two different definitions of boundary for this task:

1. Circle boundary $\mathbb{F}^{(1)}$: a pixel p is a boundary pixel if its euclidean distance between p and the center of the image is smaller than a preset threshold r.
2. Box boundary $\mathbb{F}^{(2)}$: a pixel p is a boundary pixel if its distance to the closest border of the image is smaller than a preset threshold h.

$\mathbb{F}^{(1)}$ is focused on the central area of the image, and it is prone to overlook the pixels close to the corners; $\mathbb{F}^{(2)}$ is closer to the image frames in real life, but less selective. We use $r = 0.3 \times \min\{W, H\}$ and $h = 0.1 \times \min\{W, H\}$, where W and H are the width and the height of the image, respectively. Our experiment shows that a combination of both $\mathbb{F}^{(1)}$ and $\mathbb{F}^{(2)}$ yields a better result.

We also want to use the standard deviation of each segment to measure the 2D dispersion. For i-th segment, and the set of its pixels S_i,

$$\mathbb{E}_i = \sqrt{\frac{1}{|S_i|} \sum_{p \in S_i} (x_p - x_c)^2 + (y_p - y_c)^2}$$

Where (x_c, y_c) is the coordinate of centroid. Compare with distribution contrast $\mathbb{D}$ which considers the superpixel distribution and the color difference, standard deviation feature $\mathbb{E}$ is more focused on how the pixels of the segment are distributed regardless other visual clues. Results for these clues on some fashion product images are shown in fig.2.9 column (g), (h), and (i).

2.4.3 Processing Features

First step is to normalize the feature vectors. The original feature values have different ranges. For example, as one of the feature, average L channel value in CIEL*a*b* can range from 0 to 100, while a*, b* values can also be negative. Other than that, the range can also be a variable. For example, the range of centroid position, which is the average (x,y) coordinate, is determined by the size of the image itself. Hence, it would be unwise to input raw feature values without any normalization into next step.

For a given feature vector $\mathbb{V} = [v_1, v_2, ..., v_N]$, where v_i is the feature value of i-th segment. The normalized feature vector $\mathbb{V}'$ can be expressed by

$$\mathbb{V}' = \frac{\mathbb{V} - \min(\mathbb{V})}{\max(\mathbb{V}) - \min(\mathbb{V})}$$

Therefore all the feature values are restrained to [0,1], and their individual impact to our grouping system is unified. Different weights can be multiplied to the normalized feature vector to adjust the significance of a certain feature to achieve optimal selection result. Our experiment shows that a most desirable result occurs when the standard deviation feature $\mathbb{E}$ is scaled by 0.7.

As stated in the previous section of spatial feature set, a combination of two boundary features is used in the final feature set. The circle boundary $\mathbb{F}^{(1)}$ tends to be more progressive than the box boundary $\mathbb{F}^{(2)}$: it works well when the item is exclusively lated at the center of the image, but for images with large garment pieces, where the garment spread horizontally or vertically in the image, it cuts through the garment region. Therefore in the final feature set we are only use the normalized $F^{(2)}$ feature to do the clustering, as well as the inner product of both. The new inner product feature $\mathbb{F}$ is defined as

$$\mathbb{F}_i = \mathbb{F}'^{(1)}_i \times \mathbb{F}'^{(2)}_i$$

For central segments and marginal segments, $\mathbb{F}$ performs similarly as $\mathbb{F}^{(1)}$ or $\mathbb{F}^{(2)}$, as central segments always have fewest boundary pixels and marginal segments always have most boundary pixels. In the transition area between image center and boundary, $\mathbb{F}$ behaves more conservatively. Based on our experiment and testings, the weight of this feature is set as 0.6.

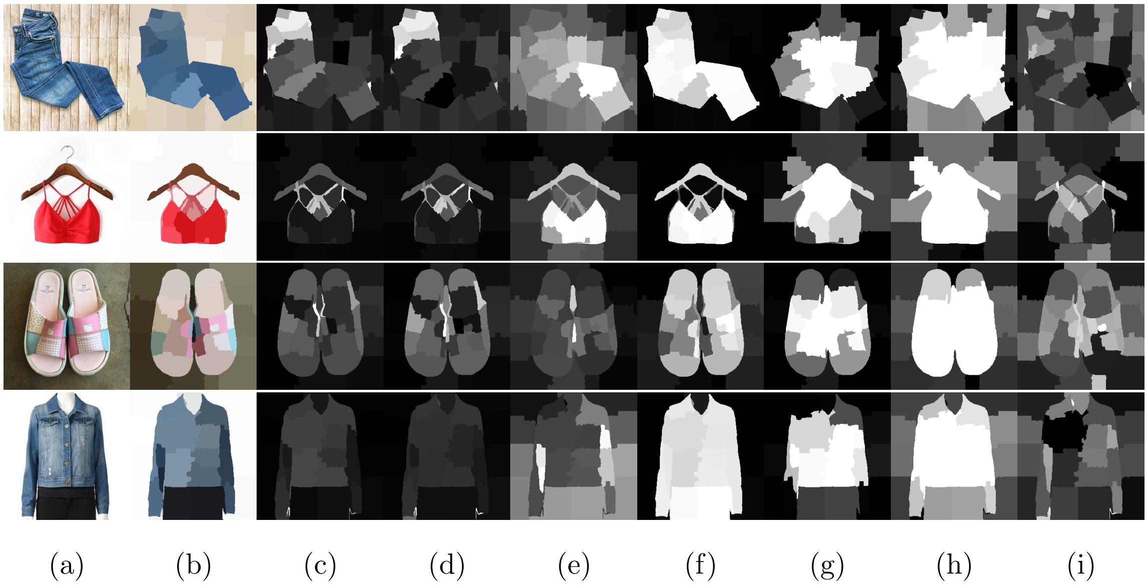


Fig. 2.9.: Features for segment grouping. (a) original image, (b) RAG segmentation result, (c) 1st order Laplacian power $\mathbb{L}^{(1)}$, (d) 2nd order Laplacian power $\mathbb{L}^{(2)}$, (e) color contrast $\mathbb{U}$, (f) distribution $\mathbb{D}$, (g) circle boundary indicator $\mathbb{F}^{(1)}$, (h) box boundary indicator $\mathbb{F}^{(2)}$, (i) standard deviation $\mathbb{E}$

2.4.4 Segment Selection

Our final goal in this module is to produce a mask that only recognizes the garment region. This can be seen as a pixel level binary classification task: every pixel in the image should be labeled as either garment or non-garment. As stated in the introduction section, pixel level image segmentation is mostly studied using supervised statistical learning especially deep neural network. However, by utilizing the features introduced above, we are able to approximate the segmentation results by clustering algorithms.

This approach seems not as ”smart” as neural networks and other supervised classification algorithm, and learning with ground truth heuristically performs better as they utilize each data point individually. However, clustering algorithms allows us to consider the correlation between segments within the given image, and there are some characteristics of this clustering task:

1. Binary clustering. Each segment can be either background or garment region
2. Few data points. Heuristically, there are about 20 segments in an image
3. Even cluster size. The garment usually has a significant amount of pixels in product portraits.

This can be used to select the clustering algorithm that fits this problem best. However due to the nature of clustering algorithms, the clustering result is less predictable and stable. Therefore, we propose a semi-supervised learning structure. First, we choose three different clustering algorithms to perform clustering, and generate three clustering results $\Theta^{(1)}$, $\Theta^{(2)}$, and $\Theta^{(3)}$. combine

$$\Theta^{(k)} = \begin{bmatrix} \theta_1 & \theta_2 & \cdots & \theta_N \end{bmatrix}$$

where N is total number of segments in the image, and for i-th segment, the prediction from k-th algorithm is $\theta_i^{(k)}$

$$\theta_i^{(k)} = \begin{cases} 1, & \text{garment} \\ 0, & \text{other} \end{cases}$$

One unanswered question is what algorithms to use. Our goal is to fully utilize the advantage of the majority vote schema, where different decisions from various clustering algorithms can be combined to render the final decision. Therefore, we aim to introduce different types of clustering algorithms in the selection schema. Here we employ three different algorithms: meanshift algorithm, Kmeans algorithm, and agglomerative clustering algorithm.

- Kmeans algorithm is widely used *centroid-based algorithm.* The objective of the kmeans is to find k cluster that can minimize the average distance between the data points and their corresponding centroids. Kmeans algorithm is one of foundamental algorithms in image processing and unsupervised learning due to its simplicity and effectiveness.

- Agglomerative algorithm performs a hierarchical clustering procedure by a bottom up approach, and it is considered as a *connectivity-based algorithm.* The linkage criteria is ward, which minimizes the sum of squared differences within all clusters [48]. Although the standard algorithm for hierarchical clustering has a time complexity of $O(n^3)$ and $O(n^2)$ memory. In our case, since the number of segments/data points in each image are usually small (about 20), the agglomerative algorithm will be a good fit for our system.

- Meanshift clustering is a *density-based algorithm* and aims to discover blobs in a smooth density of samples [49]. Due to its mode-seeking nature, meanshift tends to find local maxima where highest density of the data points will be reached. This characteristics can future help us to identify the outstanding segments (outliers) inside the image or small garment items in the image, and it works better with uneven dataset. However, mean-shift algorithm works with search windows and does not need the number of clusters beforehand. Therefore, we preset the search window bandwidth to 0.3 quantile of the dataset and run the meanshift algorithm. If the clustering result does not contain 2 clusters, we adaptively update the bandwidth so that the final clustering results would be binary.

We further show quantitative results of different clustering algorithms in later section. In Fig. 2.11, we are showing that some results from our semi-unsupervised segment selection method. Pseudo colors, red and blue, are overlaid on the image to distinguish two different labels.

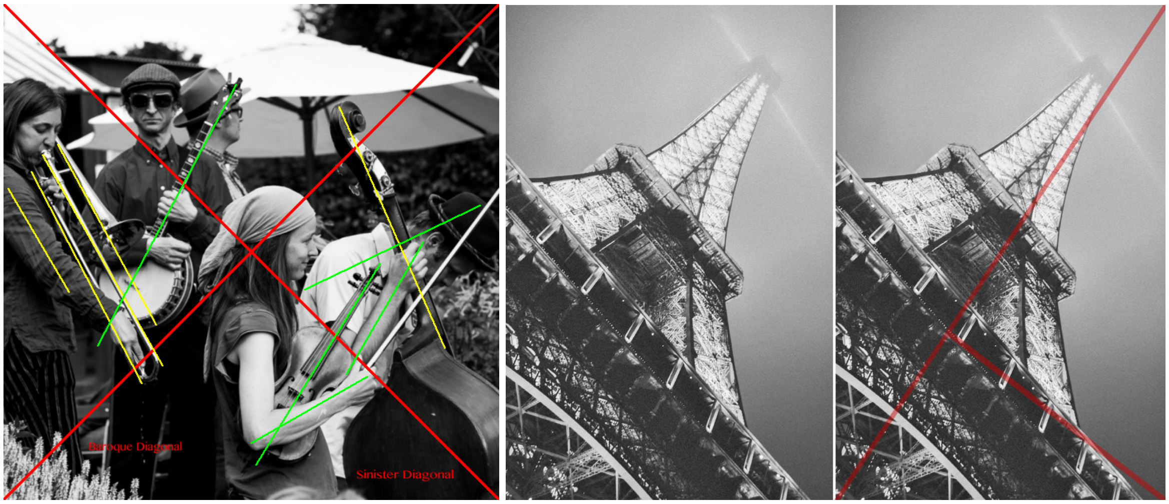
Unlike the supervised classification algorithms, the labels produced by the clustering algorithm usually only serve the purpose of separating different clusters, and they do not have a meaning. Therefore the actual labels produced by three algorithms might have different meaning and are not unified. Therefore we need to unified the label meaning and make sure each label should be matched with our definition above. We first find a pixel that can be used as a reference, which should be labeled as background, and then examine all the labels to see if the produced label matches the reference label. If not, we take the complement of the original labels as the new label.

To choose the reference pixel that is in most cases a background pixel, we investigated many portraits photographs and found out that for image composition, most important objects are aligned with or on the baroque diagonal, which is from the lower left to upper right corner. Baroque diagonal is said to provide a more pleasing and positive viewing experience, while the sinister diagonal has negative and concerning psychological implication on it. So most of the product portraits avoid arranging items on the sinister diagonal. [50] In Fig. 2.10, there are some examples given to demonstrate some examples of object alignment in portrait photography. On other fashion portrait examples we provide before it is very rare to see items that follows the sinister diagonal. Thus, we use the top left pixel as the non-garment reference pixel, and calibrate all three clustering results. Some sample results are shown in Fig. 2.11.

Then we use majority vote scheme to finalize the results. The algorithm is given in Alg.1. When disagreement exists, $\mathbb{F}^{(2)}$ is used to eliminate some border segment, making the selection more conservative.

Algorithm 1 Majority Vote

```
for $i = 1$ to $N$ do
    if $\theta_i^{(1)} = \theta_i^{(2)} = \theta_i^{(3)} = \theta_i$ then
        $\theta^* = \theta_i$
    else if $\mathbb{F}_i > 0.5$ then
        $\theta^* = 0$
    else
        $\theta^* = \sum_{k=1}^{3} \theta_i^k - 1$
    end if
end for
```



(a)

(b)

Fig. 2.10.: Examples of Baroque and Sinister diagonal analysis in photography. Most object align along or on the two diagonals in (a) [50], and based on (b) [51], baroque diagonal is more visually pleasing or important than sinister.

The final segmentation results are shown in the last column in Fig. 2.11. In general the algorithm produces relative robust result.

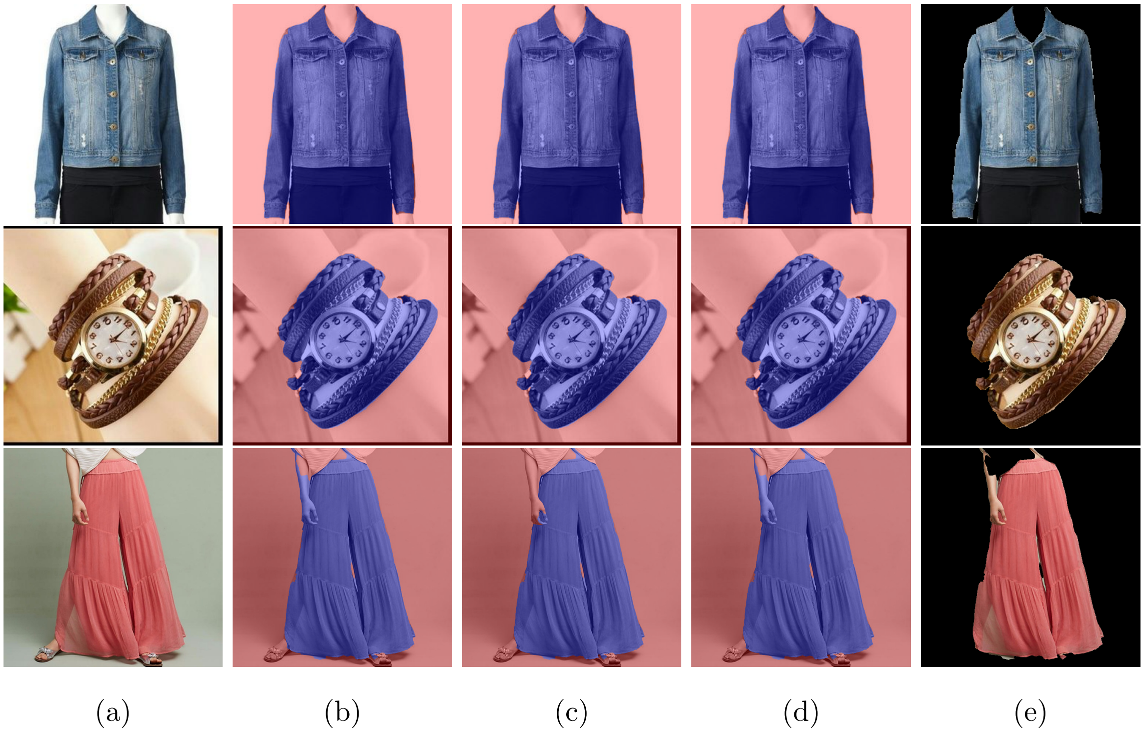


Fig. 2.11.: The clustering results and final clustering result. The column (a) is the original image; column (b) is the meanshift algorithm; column (c) is the agglomerative clustering with Ward distance; column (d) is the Kmeans algorithm result; the rightmost column are the final result

2.5 Neural Network Training

As stated before, one of crucial requirement for building a supervised semantic segmentation algorithm is the dataset. Furthermore, to train complex neural network architectures, astronomically large datasets are desired that contain hundreds of thousands images and manually labeled ground truths, and collecting data in such a large scale is very challenging.

For segmenting the fashion garment, fashion datasets that suit our need are very rare. Therefore, in this section, we adapt and repurpose a public dataset and train a Deeplab semantic segmentation architecture in hope to better process images with human bodies or multiple items.

2.5.1 Dataset Preparation and Model Training

One of datasets that recently become available for fashion segmentation research is Multi-Human Parsing (MHP) dataset by Feng et al. [52, 53]. A snapshot of MHP dataset is shown in Fig. 2.12. It consists of more than 15,000 images for training and 5,000 images for validation. All images are collected from stock photos, TV/movie productions, and internet, and the collection shows large variation of culture, region, and time/era. And for every training or validation image, there are pixel-level segmentation maps corresponding to the semantic category of the human subjects and clothing. In short, the scale of the dataset is ideal for training neural network based algorithms.

While the focus of multi-human parsing is to label every pixel in an image (with more than one persons) as a semantic category (body part, clothing item, and other objects) and the human subject it belongs to, our fashion segmentation problem is much simpler – only the semantic information of fashion and body part is needed. Therefore, some modification on the annotations are needed to fit in our work before training the network.

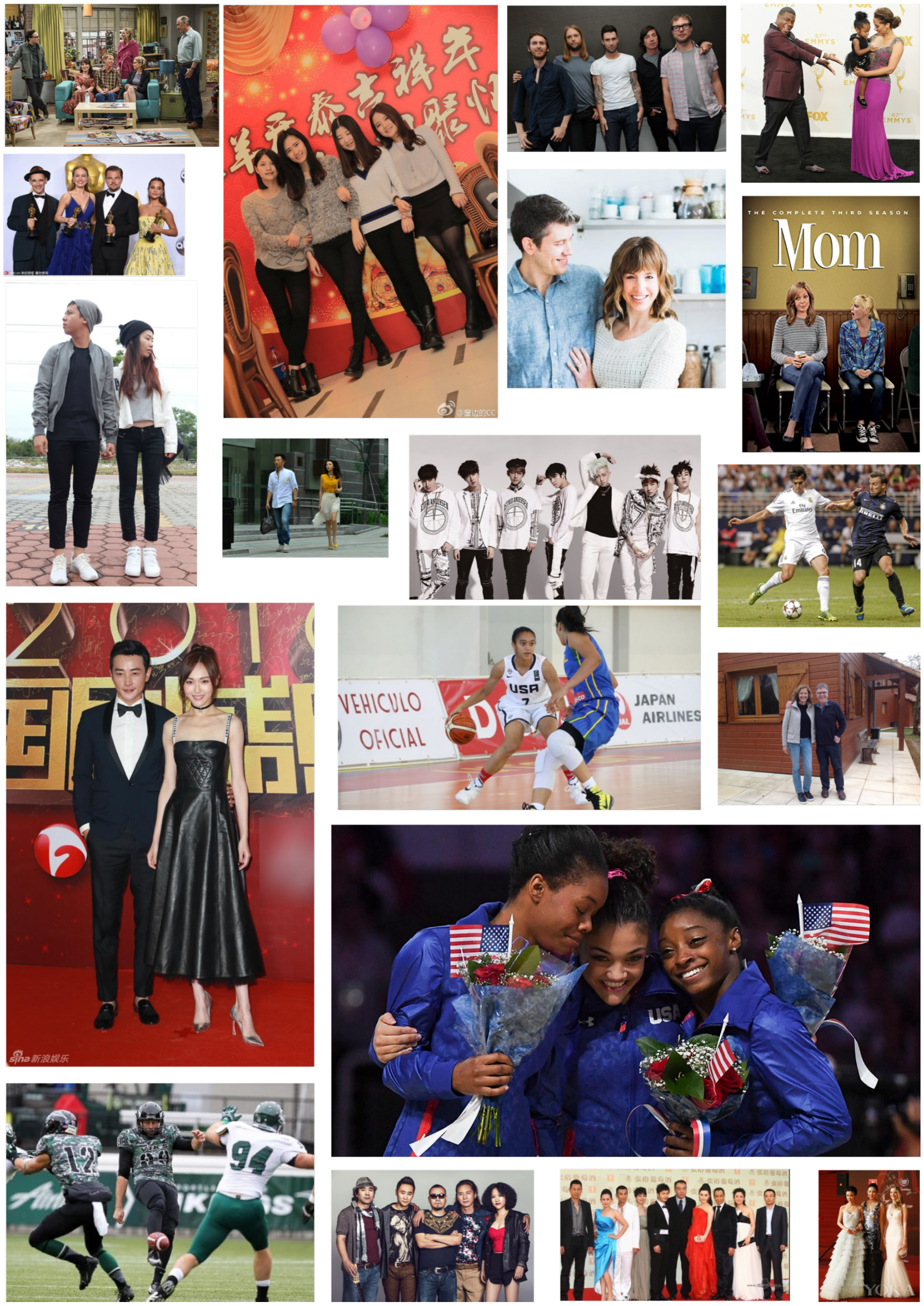


Fig. 2.12.: A snapshot of Multi-Human Parsing Dataset [52,53]. The image collection reflects a large variety of human subjects and real world scenarios, and all images contain at least two persons.

Data Adaptation

To fit the dataset into our research purposes of fashion segmentation, the following modifications:

1. **Merging multi-human segmentation ground truths into the single segmentation label.** Our research focus is to classify the clothing and body pixels regardless different individuals in the image. Therefore, we would like to incorporate all the annotations into one single segmentation map as shown in .

2. **Simplify and reduce the number of semantic categories in the segmentation.** Our goal is to reduce the unnecessary complexity of our problem. The modification contains two folds:

   - Combine subcategories into a blanket master category. For instance, we combine *left shoe* and *right shoe* as *shoes*. This allows us to deploy image augmentation techniques such as flipping to enrich our dataset.

   - Remove unrelated labels like *ball*. These items not only are not relevant to our mission, but also occur rarely in the dataset, which might further skew the training and validation process.

Our simplified dataset contains 48 semantic labels, and we further enrich our dataset variation using image augmentation techniques.

Data Augmentation

In hope of increasing the robustness of trained model and avoiding over-fitting, a series of data augmentations are deployed. Image loading operation and augmentation operations are executed by the dataset loader and the image transformation pipeline jointly during the training. The transformation sequence is as:

1. **random flip and rotation** Flipping (vertical and horizontal) and rotation are some of the most used techniques for enriching the dataset. Possibilities of

random flipping horizontally and vertically are 0.5 for each operation, and we set the rotation range as [-45°, 45°].

2. **Gaussian noise and Gaussian blur** We also further introduce gaussian noise and gaussian blur in our training set to enhance the robustness of the trained network. And previous work has shown that such training with noise is equivalent to a form of regularization in which an extra term is added to the error function [54]. For a given image, only up to one gaussian operation will be applied. In other words, an image can be free from any gaussian operation. However, an image can't be blurred and get noise at the same time.

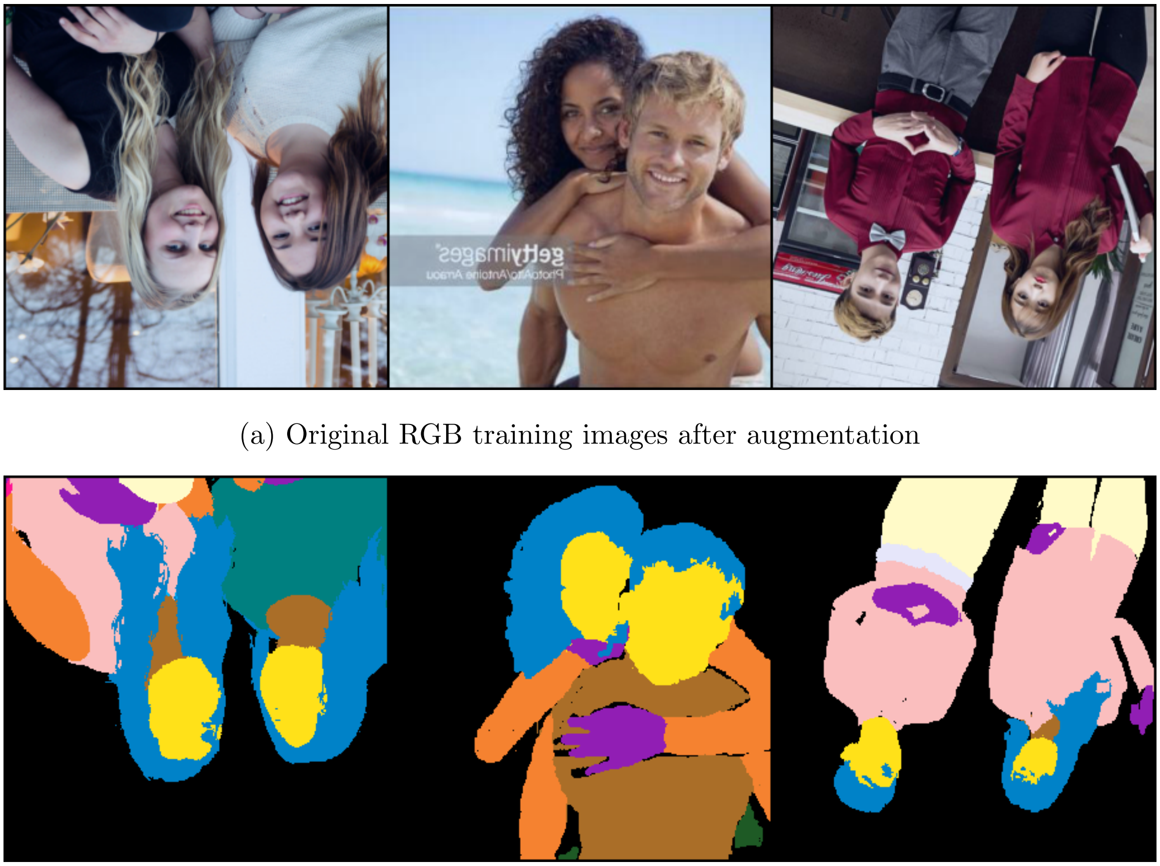
3. **image scaling** The MHP dataset contains images with different aspect ratios and image resolutions. Thus, it is desired to unify the image shape before training. First, we resize the image in such a fashion that the aspect ratio is unchanged but the shorter side of image is 300 pixel wide/tall. Then, we rescale the image with random scale factor ranging from 0.8 (zoom out with padding) and 1.3 (zoom in in the center). In this way, we can encourage the network to be more adaptive towards objects in different sizes.

4. **random crop** In addition to the image scaling, training and validation images are cropped randomly into 300×300 patches by the image transformation pipeline. Because the rescaling step is designed to confine the image dimension roughly same as the cropping size, our final cropped patches are able to capture the majority of image scene.

Fig. 2.13 shows some examples of training image patches after augmentation. The augmentation the With all image augmentation operations, we can

Model Training

Here we are using the DeepLab v3+ proposed by Chen et al. [55]. DeepLab is a series of semantic segmentation models developed at Google for understanding images under the pixel level. Compared with the traditional deep convolutional neural network segmentation models, where score maps for all semantic classes are generated at the end of the process, Deeplab networks utilizes Conditional Random Field (CRF) to reinforce the exact outlines of objects and introduce atrous convolution to assimilate larger field of view content. The networks have achieved state-of-the-art performance on public benchmarking datasets and deployed on Google's own imaging devices.



(a) Original RGB training images after augmentation

(b) Ground truth annotation after augmentation

Fig. 2.13.: Visualization of three training images after augmentation (a) and their corresponding ground truth annotation (b). All training patches are cropped into 300×300, and flipping, rotation, and blurring can be observed. The label for the ball is remapped into background in the center patch.

The newest version, DeepLab v3+, uses the encoder-decoder architecture and deploys a decoder module to improve the segmentation performance on object boundaries. It also boosts the accuracy and runtime performance by using depth-wise separable convolution in decoder and the atrous spatial pyramid pooling operations. The model achieves the state-of-art results on PASCAL VOC 2012.

For our application, we choose Deeplab v3+ using ResNet [56] as backbone. To battle the issue of the class imbalance, we both used frequency-based class weight and focal loss to emphasize the importance of under-represented classes. The balanced weight $w(m)$ for class m can be defined as

$$w(m) = \frac{1}{\log(1 + f(m))}$$

where $f(m)$ is the class frequency in the training data batch. This will significantly boost the importance of the class that has low frequency in the dataset.

Another way to emphasize the learning for hard examples is focal loss. Focal loss is proposed by Lin et al. [57] for dense object detection. Focal loss FL is defined as

$$FL(p_m) = -(1 - p_m)^\gamma \log(p_m)$$

where p_m is defined as

$$p_m = \begin{cases} p, & \text{if } y = 1 \\ 1 - p, & \text{otherwise} \end{cases}$$

indicating the confidence of the prediction of class m. When an example is misclassified and p_m is close to 0, the modulating factor is close to 1 and the loss is unaffected.As p_m is approaching 1, the weight goes down and the importance of the well-classified examples is down-played. γ is the parameter that controls the effect of focal factor. If $\gamma = 0$, focal loss is equivalent to cross entropy. When γ increases, the effect of the focal factor increases accordingly.

Table 2.3.: Training configuration

Configuration	Parameters	Note
Backbone	ResNet	Proven good accuracy
Loss	Focal loss	–
Crop Size	300×300	Poshmark image size; GPU memory constrain
Batch Size	8	Maximum under 300px crop size
Learning Rate	0.007	After tuning result
Learning Rate Scheduler	Poly	–

2.5.2 Experimental Results

We use balanced weight and focal loss to train DeepLab v3+ network with ResNet as backbone. Table 2.3 shows the configuration we used for the optimal trained model.

Our training curves are shown in Fig. 2.14. Before talking about the curves, we need to define the performance evaluation metrics first.

- **Pixel Accuracy**

  For $k+1$ classes (K target semantic labels and 1 (one) background/unlabeled), we define n_{ij} as the number of pixels where the ground truth label is i, and the prediction is j. Therefore, we can define n_{ii} as True Positive (TP), n_{ij} as False Negative (FN), and n_{ji} as False Positive (FP).

  Thus, we can define the Pixel Accuracy (PA) [58] as

$$PA = \frac{\sum_{i=0}^{K} n_{ii}}{\sum_{i=0}^{K} \sum_{j=0}^{K} n_{ij}} = \frac{\sum_{i=0}^{K} n_{ii}}{N}$$

  Where $N = \sum_{i=0}^{K} \sum_{j=0}^{K} n_{ij}$ is the total number of pixels in the image.

  However, PA does not take the class imbalance issue into account while calculating the final accuracy, and it can only show the performance for smaller classes compared with dominant classes while ignoring minority classes. Class imbalance is prevalent in many semantic segmentation problems, including ours. From the observation, we can see that classes like background, face, hair, and arms are the predominant classes our dataset: faces and background almost always exist in all image patches and occupy a significant portion of the image, while classes like heels, helmets, and ties appear not only less often but also in a relatively small size in our dataset. Therefore, we want to use Intersection over Union (IoU) [58] metrics to evaluate the model performance.

- **Mean Intersection over Union (MIoU)**

  In addition to calculating pixel accuracy, we also calculate Mean Intersection over Union [59].

  First, for each class, we calculate the IoU. For m-th class, the number of true positive pixels, also known as the intersection, $I(m)$, for class m can be written as n_{mm}. Then $\sum_{j=0}^{K} n_{mj}$ represents the number of pixels that are labeled as m in ground truth, and $\sum_{j=0}^{K} n_{jm}$ represents the number of pixels that are all predicted as m by our model.

  Therefore, the union for class m, $U(m)$, can be written as

$$U(m) = \sum_{j=0}^{K} n_{mj} + \sum_{j=0}^{K} n_{jm} - n_{mm}$$

  ,

  Note that we deducted the intersection in the calculation, as the intersection is included in both $\sum_{j=0}^{K} n_{mj}$ and $\sum_{j=0}^{K} n_{jm}$.

  Then the IoU for class m is

$$IoU(m) = \frac{I(m)}{U(m)} = \frac{n_{mm}}{\sum_{j=0}^{K} n_{mj} + \sum_{j=0}^{K} n_{jm} - n_{mm}}$$

  Thus, the Mean IoU can be derived as

$$MIoU = \frac{1}{K+1}\sum_{m=0}^{K} IoU(m) = \frac{1}{K+1}\sum_{m=0}^{K}\frac{n_{mm}}{\sum_{j=0}^{K} n_{mj} + \sum_{j=0}^{K} n_{jm} - n_{mm}}$$

- **Frequency Weighted Intersection over Union (FWIoU)** We define the frequecy of class m $P(m)$ is the ratio of number of pixels labeled as m in the ground truth to the total number of pixels.

  $$P(m) = \frac{\sum_{j=0}^{K} n_{mj}}{N}$$

  And the FWIoU [60] can be expressed as weighted sum of IoU's

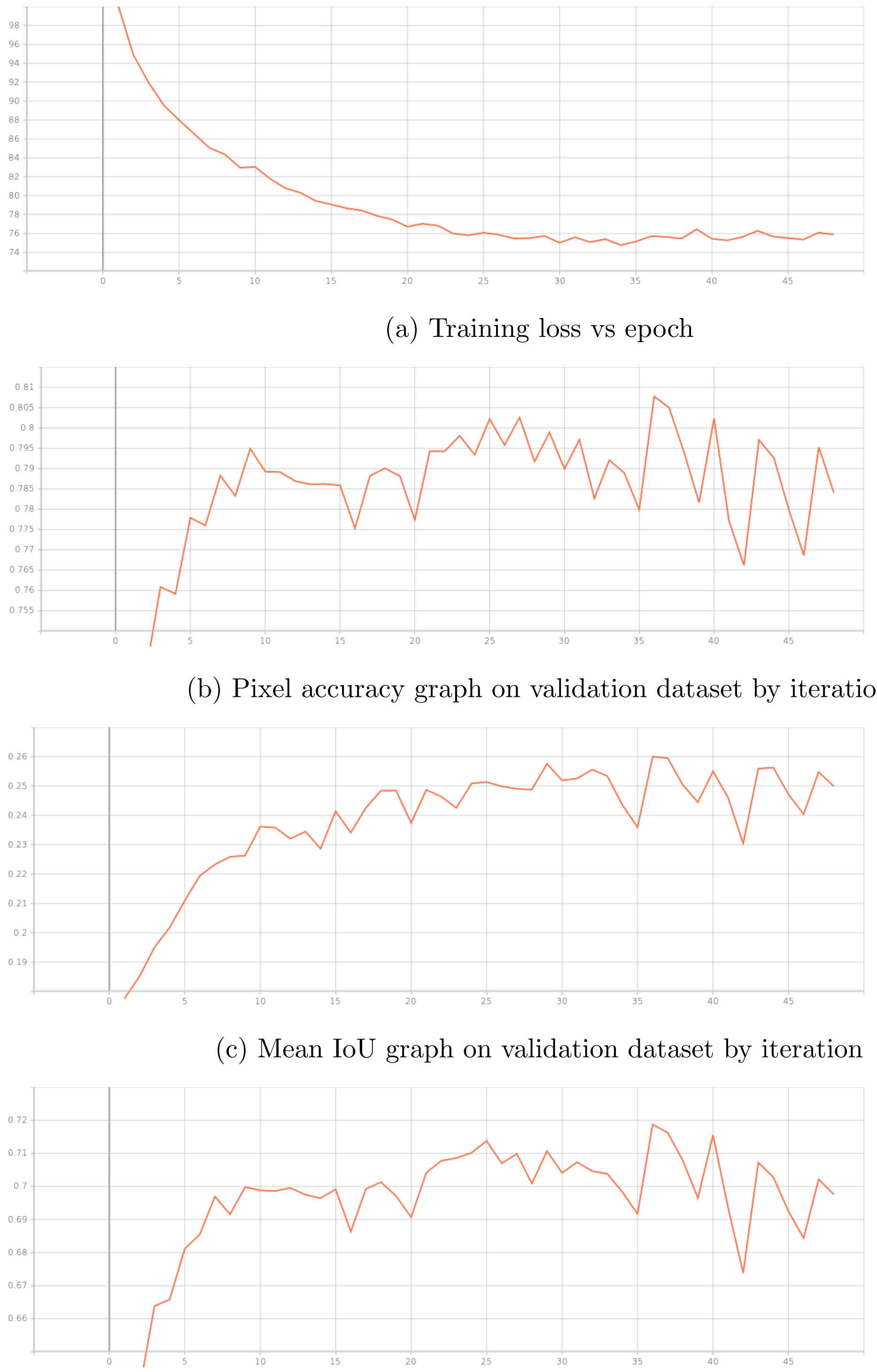
  $$FWIoU = \sum_{m=0}^{K} P(m) \times IoU(m) = \frac{1}{N}\sum_{m=0}^{K}\frac{n_{mm}\sum_{j=0}^{K} n_{mj}}{\sum_{j=0}^{K} n_{mj} + \sum_{j=0}^{K} n_{jm} - n_{mm}}$$

  Compared with MIoU, FWIoU incorporates the occurrences of semantic categories in the calculation and gives an overall accuracy.

The model performances during the training are shown in Fig. 2.14, and segmentation results are shown in Fig. 2.15 and 2.16.

1. Body parts like face, arms, hairs, and hands are easier to identify compared to rest of the fashion categories. This is due to the consistent appearance of the human body parts. However, feet are harder to detect, thanks to different foot wears and in general less coverage and attention on foot.

2. The network tends to have better prediction on male clothing. This is due to the fact that most men in the dataset are dressed relatively simple compared with women's dressing: less accessories, less frequent complicated patterns, and more predictable dressing combinations (usually jacket+shirt+dress pants, or shirt+dress pants). Note that we cannot conclude that the segmentation network is able to identify the gender of the human subjects; our observation



(a) Training loss vs epoch

(b) Pixel accuracy graph on validation dataset by iteration

(c) Mean IoU graph on validation dataset by iteration

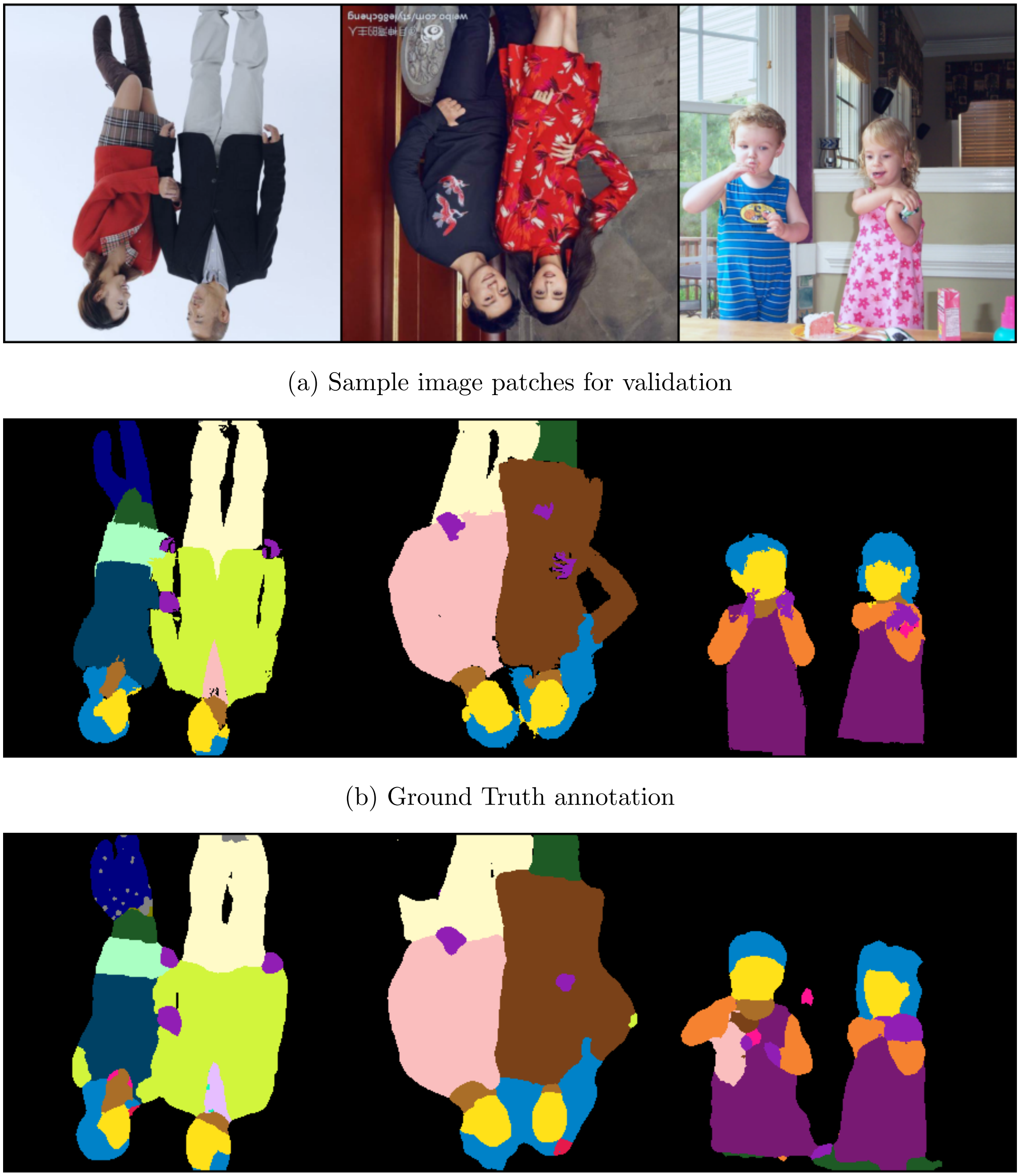
(d) FWIoU graph on validation dataset by iteration

Fig. 2.14.: The training procedure and the model performance validation set during the training process produced by Tensorboard. The network training converges at about 36th epoch.

simply suggests that the network captures the tendency of certain highly frequent dressing combinations, and these patterns happen to echo men's dressing code in the dataset.

3. Similar classes (for example, robe and dress) create confusions. Moreover, items of different categories can be worn or shown in different ways that further confusion.

4. Smaller items such as necklaces and hair accessories tend to be ignored during the segmentation. This might due to the fact that atrous convolution tends to incorporate global context instead of localization.

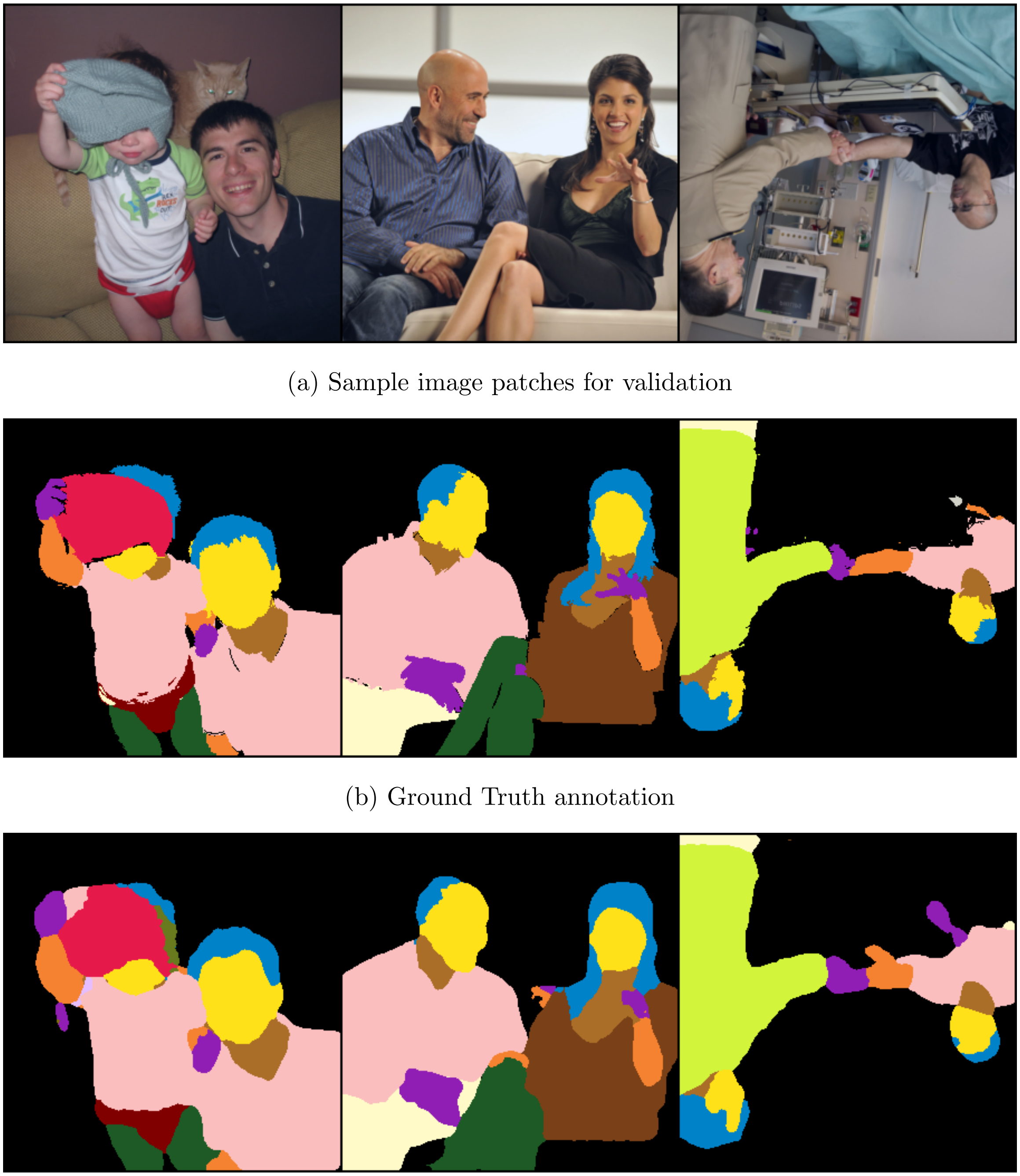


(a) Sample image patches for validation

(b) Ground Truth annotation

(c) Prediction from Deeplab model

Fig. 2.15.: First patch of examples of Deeplab model for fashion semantic segmentation. () shows three images, and ground truth annotation is shown in (). (c) shows the prediction from the deeplab model. The model is able to produce high accuracy prediction.

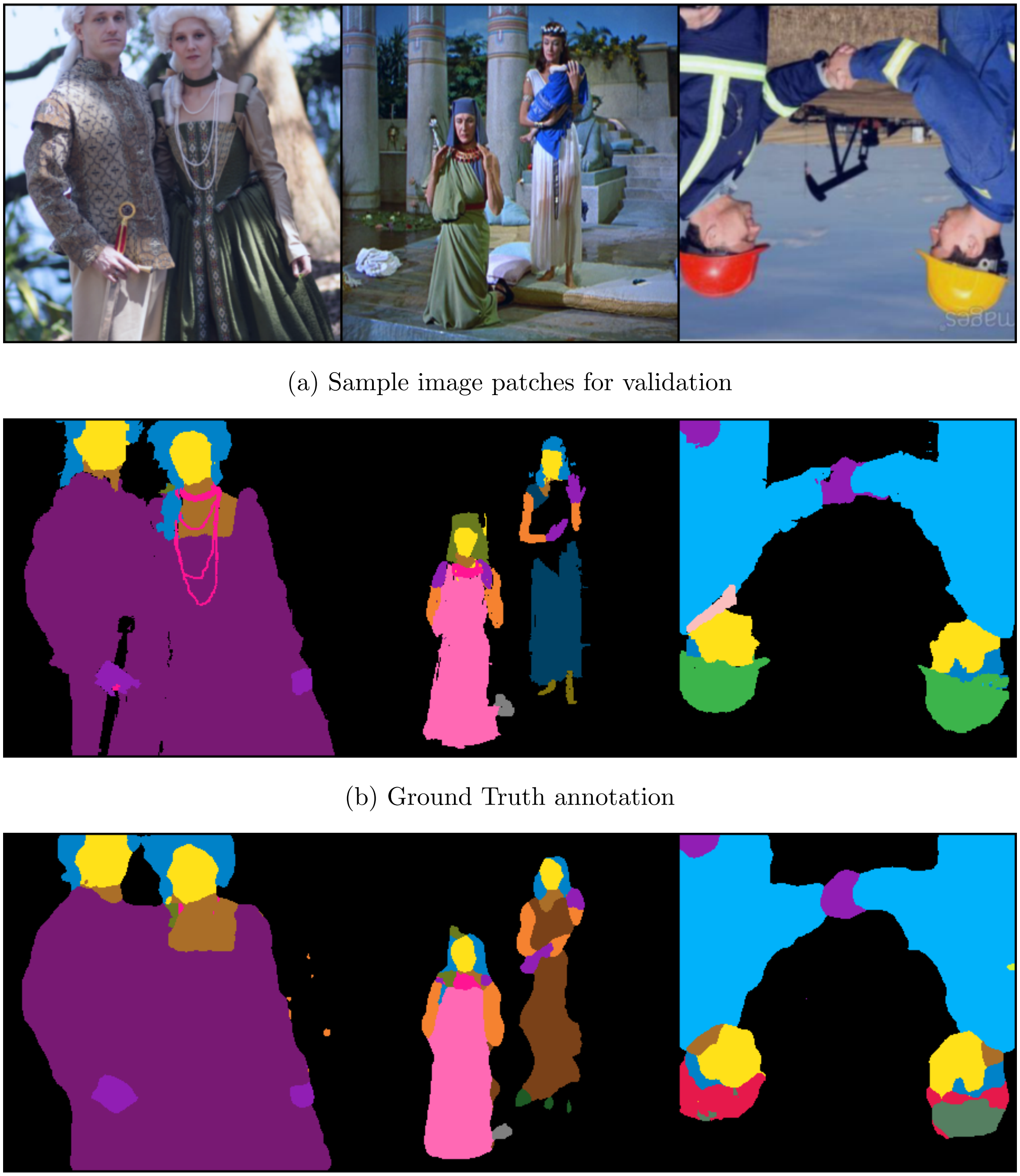


(a) Sample image patches for validation

(b) Ground Truth annotation

(c) Prediction from Deeplab model

Fig. 2.16.: First patch of examples of Deeplab model for fashion semantic segmentation. (a) shows three images, and ground truth annotation is shown in (b). (c) shows the prediction from the deeplab model. The model is able to produce high accuracy prediction.

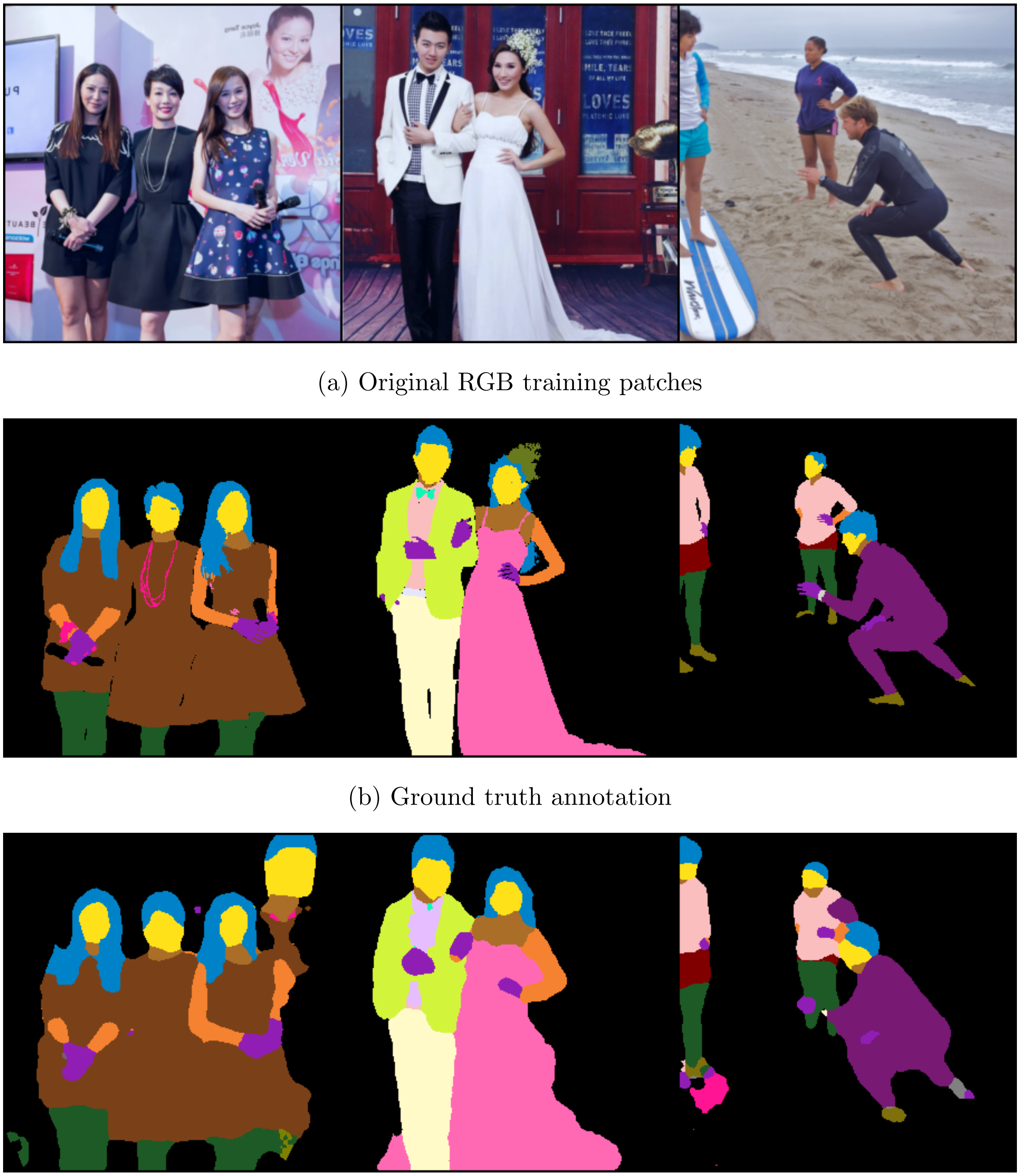


(a) Sample image patches for validation

(b) Ground Truth annotation

(c) Prediction from Deeplab model

Fig. 2.17.: Second patch of examples of Deeplab model for fashion semantic segmentation. (a) shows three images, and ground truth annotation is shown in (b). (c) shows the prediction from the deeplab model. The model is able to produce high accuracy prediction.



(a) Original RGB training patches

(b) Ground truth annotation

(c) Deeplab prediction

Fig. 2.18.: Visualization of three training images after augmentation (a) and their corresponding ground truth annotation (b). All training patches are cropped into 300×300, and flipping, rotation, and blurring can be observed. The label for the ball is remapped into background in the center patch.

2.6 Color Extraction and Fashion Color Naming

The first module of the Autonomous Garment Extraction System is to find the color from the garment region, and correspond the color value to a certain color description.

2.6.1 Color Extraction With Gaussian Mixure Model

Gaussian mixture model is a probabilistic model for representing subpopulations that complies the normal distribution within an overall population, and it is widely used in color imaging to extract the mean color from a population of color data points. Here we use Gaussian Mixture Model and expectation maximization algorithm to obtain the estimated average color(s) from given color vectors from garment region. We set the number of Gaussian distributions as 2 to generate at least two colors for each image. If these two colors are very similar to each other($\Delta E < 30$), then we can say that this garment only has one color. Sample result is shown in the Fig. 2.19. The GMM-based image color summarizer matches the human visual perception.

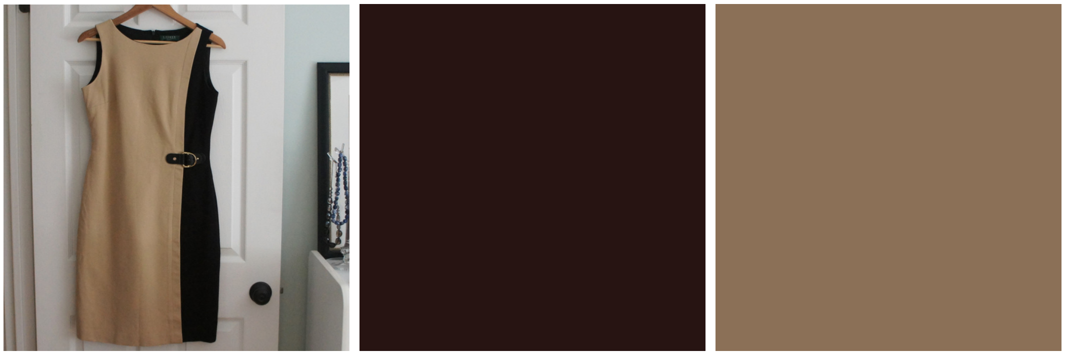
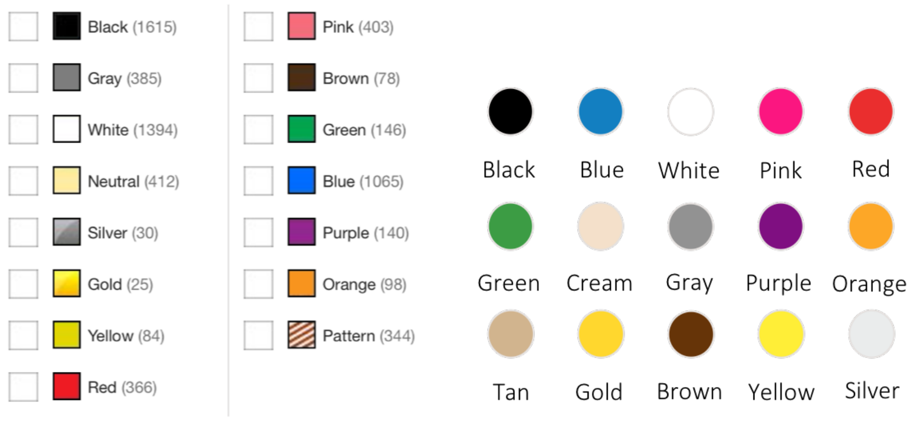


Fig. 2.19.: Sample color extraction results

Next step is to match the color coordinates with the pre-defined color palette on the fashion retail website, shown in Fig. 2.20(b).

2.6.2 Fashion Color Naming on Online Fashion Marketplace

As illustrated in the previous section, for color management in the fashion industry, unlike other industries, where color names usually are objective and given based on the color appearance, most of the fashion color naming is done by the manufacturers individually before products are released. Other than that, seasonally trending color names are also predefined and maintained every season by fashion experts. Furthermore, those names are not standard color description used in the previously mentioned scientific color naming systems. Yet it is not a standard color naming system that is universally recognized by all fashion manufacturers.



(a) (b)

Fig. 2.20.: The color palette on different fashion website

Other than the problem of the lack of a universal standard fashion color system for manufacturers, for fashion retail, especially online retailers, color naming also has many challenges that are unique to it:

1. Different websites have different fashion palette definition. For instance, in Fig. 2.20, we show two palettes from Poshmark and Neiman Marcus, respectively. For example, neutral and pattern only exist on the Neiman Marcus website.

2. Some color families have more granular color descriptions. For example, on the Poshmark website, the yellow family has yellow, gold, and orange. For comparison, blue family only has one label.

3. Color can appear different on different websites. For example, Poshmark silver is much lighter than Neiman Marcus silver.

4. Color names can also represent texture, like silver and gold on the Neiman Marcus website, which indicate not only the color hue, but also the metallic texture look.

Therefore, it is important to develop our algorithm based on the color philosophy and organization of the marketplace, and a platform/marketplace-driven color categorization algorithm is proposed.

2.6.3 Platform-driven Approach using CIELa*b* color difference

Focusing on the fashion marketplace definition, we extract the RGB values from the website's color palette, then we calculate and compare the color differences between extracted mean colors to all the reference colors. This is a very simple solution, and generally works well when the colors are bright and highly saturated, for example red and green. To further enhance the performance on other colors, colors are grouped and classified using different strategies within each group.

The new color matching method first groups all the reference colors into three kinds:

- Neutral colors – black, white, silver, and gray. These color appearances are usually consistent and have a lower variance.

- Off-neutral colors – cream, brown, tan. Off-white colors have very long chroma values, but have a moderate amount of variations.

- Saturated colors – red, yellow, green, blue, purple, orange, pink. The first impressions for these saturated colors usually are the appearances with high chroma and saturation. However the variation is bigger than other two kinds.

Then, we categorize the extracted mean color into its corresponding kinds by the following criteria:

- A color c is a neutral color if its chroma is smaller than 10.
- A color c is a saturated color if its chroma is higher than 20.

All the colors whose chroma is between 10 and 20 are the off-neutral color. Therefore, each color only has to compare with its corresponding kind of colors, instead of the entire set of reference colors. For both off-neutral color and neutral color set, we use the reference color with the smallest ΔE as final color results, and for saturated color, in order to accentuate the significance of hue, we use the color with smallest cosine similarity. The results for neutral and saturated colors are improved.

Our nearest-neighbor-based algorithm is a straightforward answer to the color matching problem. And it is very computationally easy to use, as it only needs to store n reference colors and takes $O(n)$ to produce the final answer, which is a constant. In addition, for an online fashion marketplace or any online fashion retailer, the number of color choices offered is always limited. However, this NN-based method has some limitations:

- **Accuracy.** The model generally fails to provide accurate predictions due to oversimplification. Our algorithm only uses a limited number of reference points, and each class only has one point. The algorithm also oversimplifies the complexity by drawing hard decision boundaries in Lab color space.
- **Flexibility.** The model can only work for reference points from the website, and it cannot learn users' color naming schema to further improve the color naming algorithm.

- **Accuracy.** The model generally fails to provide accurate predictions due to oversimplification. Our algorithm only uses a limited number of reference points, and each class only has one point. The algorithm also oversimplifies the complexity by drawing hard decision boundaries in Lab color space.
- **Flexibility.** The model can only work for reference points from the website, and it cannot learn users' color naming schema to further improve the color naming algorithm.

Our research also shows that human subjects tend to associate color appearance to verbal description based on their own life experience, without referencing the color palette provided by the marketplace platform. Fig. 2.21 shows three different items sold on poshmark.com, and all of them are listed as pink. However only the left one is actually closer to the pink color defined. Therefore, we want to build a data-driven color naming classification method that can learn color naming schema from the online fashion shoppers.

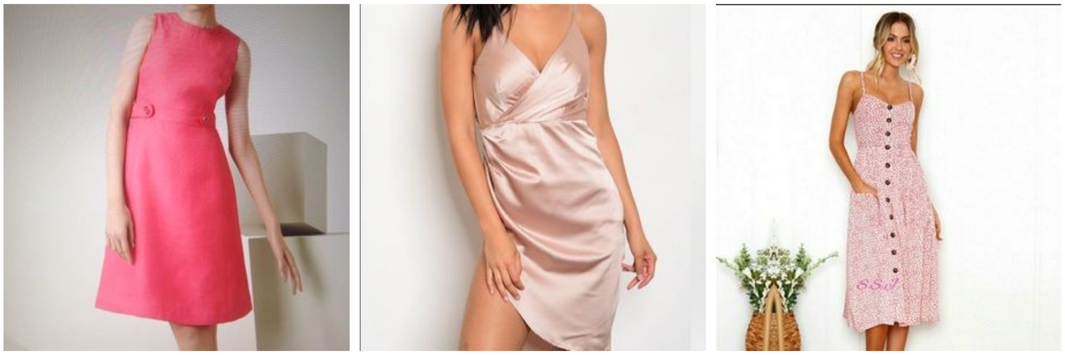


Fig. 2.21.: Examples of three pink dresses sold on Poshmark. All three dresses are labeled as pink by their corresponding sellers on the website, but color appearances are very different from the website's definition.

2.7 Data-driven Approach using Random Forest

2.7.1 Reversed Color Naming Experiment

For the fashion marketplace, it is important to constantly and actively learn color descriptions from the online fashion community instead of from predefined labels. To further study how the online fashion community names colors, we design and conduct a psychophysical experiment called the *Reversed Color Naming Experiment*. We call the experiment reversed because instead of asking human subjects to name the color in a photograph, human subjects in our experiment are supposed to choose the color from the image based on a given color name.

For this experiment, we collected more than 2,500 images from a fashion P2P website; and their colors are labeled by their sellers while making the listing. Note that 1) For our specific case, up to two color labels are allowed per listing by the website. Therefore, sellers can only choose the top two predominant colors where there are more than two colors on the garment. You can see the full color palette in Fig. 2.20(a). 2) Each listing may contain multiple images, and we assume all images within the same listing share the same color label.

We invited 6 human subjects to participate in our experiment. They are 20-30 year-old adults with normal color vision. We chose this age range to reflect the demographic of online fashion shoppers, and color vision screening was done by using Color Blindness Test [61]. Although it was not a rigid requirement, all of our human subjects were avid online fashion shoppers. Their previous online fashion shopping experience helped them understand each image and its color. Each human subject was assigned a set of fashion product images. On average, each person had 400 - 500 images to label within a week.

To keep our color experiment as simple and consistent as possible, we asked our human subjects use an Apple MacBook Pro with Retina Display to view the images and participate in the experiment. The experimental procedure can be described as follows: For a given image, we present the label(s) associated with the listing, and our human subject is expected to use the color picker to choose the color pixel(s) that is most representative of the color label. Then the selected color values are converted to CIELab and the subject moves on to the next image.

In this part, we further discuss our paradigm of using the random forest algorithm to extract human viewers' fashion color naming schema.

2.7.2 Model Overview

We use a random forest [62] to match numerical color values to verbal description. Random forest is one of most widely used machine learning algorithms due to the good accuracy, robustness and ease of use. The algorithm is one type of ensemble classification method that combines a set of simple decision tree classifiers such that each tree is trained on a series of training data sampled independently and with the same distribution for all trees in the forest.

By running many decision trees and aggregating their outputs for prediction, the random forest algorithm fixes sub-par model robustness of a single decision tree; and it can also control over-fitting. A typical random forest classification training workflow can be described as follows:

1. Sample N different sets of training data from the original dataset using bootstrapping.

2. If there are M features, max split feature m ($m << M$) is determined such that at each node, m features are selected at random out of the entire set of M features and the best split on these m is used to split the node. The value of m remains unchanged during the forest growing.

3. Given each dataset, grow a decision tree to the largest extent possible. No pruning is required.

For inference, for each data point, we pass the M-dimensional feature vector to all the decision trees grown, and the final result is determined by majority vote.

It has been shown that the classification error resulting with random forest depends on two things [62]:

1. The correlation between any two trees in the forest. The more de-correlated trees are, the more accurate the forest is.
2. The strength of each individual tree in the forest. A strong tree has high accuracy, and increasing the strength of all the individual trees helps to improve the forest accuracy.

We can adjust m to change the between-tree correlation and individual tree strength. Larger m increases both the correlation and the strength. Therefore m should be further fine tuned to ensure optimal performance.

2.7.3 Feature Design

As previously stated, the key requirement of random forest is a large pool of features. We hereby introduce all features we engineer in this work.

We developed a set of seven features for our classifiers. All of our features are based in CIELab color space. The first three features are the L, a^*, and b^* values. Furthermore, we define chroma C as

$$C = \sqrt{{a^*}^2 + {b^*}^2}, \tag{2.1}$$

and hue H as

$$H = \arctan\left(\frac{b^*}{a^*}\right) \tag{2.2}$$

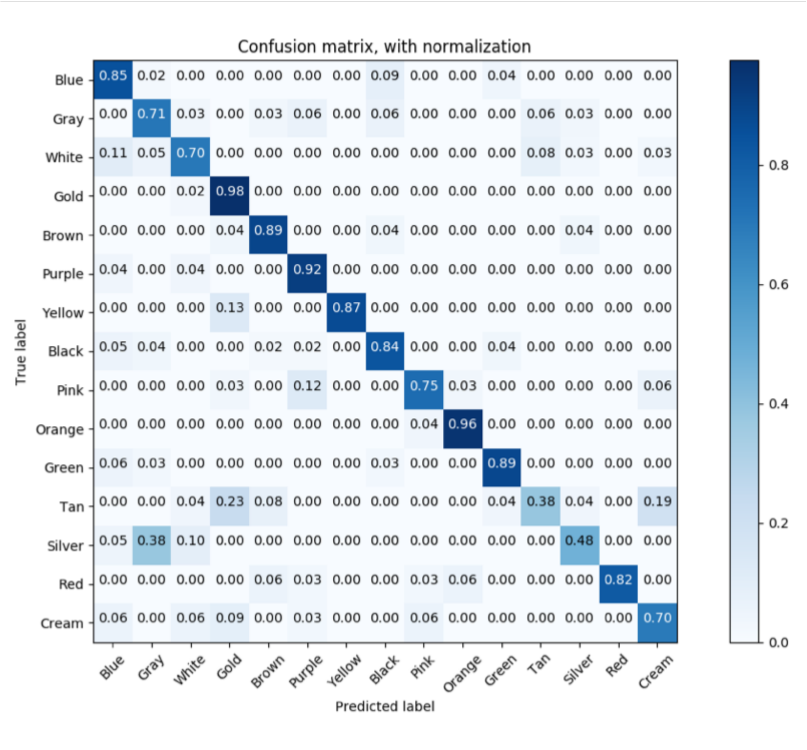
where a^* and b^* are from the CIELa*b* color coordinates. We also introduce other quadratic features like La^* and Lb^*. We will further study the feature importance later.

2.7.4 Model Evaluation and Analysis

We split our data from the reversed color naming experiment into 70/15/15 for training, validation, and testing, respectively. Then, multiple combinations of hyper parameters (the number of trees, maximum split feature m) are proposed for model selection. After validation, we grow 25 trees for our classifier; and we follow the general practice and choose m as $\lfloor \sqrt{M} \rfloor = 2$. Figure 2.22 shows the confusion matrix on the testing set for our proposed algorithm. We have the following observations from our testing:

- Overall, our algorithm performs very well with average testing accuracy of 80%. Compared with previous 60% accuracy from the previous nearest-neighbor method, a huge improvement can be gained by using random forest.
- Generally, focal colors like blue, purple, and orange have higher accuracy.
- Off-white colors (gray, cream, silver) create much confusion.
- There are also a couple of confusing pairs: pink-purple, tan-gold-brown, tan-cream, yellow-gold. These pairs can be shown to be very close to each other.

Furthermore, we study the importance of all the features. Feature importance is calculated as node impurity (here we use Gini impurity index) decrease weighted by the probability of reaching that node. The node probability can be calculated by the number of samples that reach the node, divided by the total number of samples. The higher the value the more important the feature [63]. Figure 2.23 indicates the feature importance from the most to the least important. We can see that luminance value is the most important feature of all; and all other features share relatively similar importance. We can also see that by adding non-linear features to the classification system, we drastically improve the model complexity from the nearest-neighbor based color naming method.



True label \ Predicted label	Blue	Gray	White	Gold	Brown	Purple	Yellow	Black	Pink	Orange	Green	Tan	Silver	Red	Cream
Blue	0.85	0.02	0.00	0.00	0.00	0.00	0.00	0.09	0.00	0.00	0.04	0.00	0.00	0.00	0.00
Gray	0.00	0.71	0.03	0.00	0.03	0.06	0.00	0.06	0.00	0.00	0.00	0.06	0.03	0.00	0.00
White	0.11	0.05	0.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.08	0.03	0.00	0.03
Gold	0.00	0.00	0.02	0.98	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Brown	0.00	0.00	0.00	0.04	0.89	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.04	0.00	0.00
Purple	0.04	0.00	0.04	0.00	0.00	0.92	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Yellow	0.00	0.00	0.00	0.13	0.00	0.00	0.87	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Black	0.05	0.04	0.00	0.00	0.02	0.02	0.00	0.84	0.00	0.00	0.04	0.00	0.00	0.00	0.00
Pink	0.00	0.00	0.00	0.03	0.00	0.12	0.00	0.00	0.75	0.03	0.00	0.00	0.00	0.00	0.06
Orange	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.04	0.96	0.00	0.00	0.00	0.00	0.00
Green	0.06	0.03	0.00	0.00	0.00	0.00	0.00	0.03	0.00	0.00	0.89	0.00	0.00	0.00	0.00
Tan	0.00	0.00	0.04	0.23	0.08	0.00	0.00	0.00	0.00	0.00	0.04	0.38	0.04	0.00	0.19
Silver	0.05	0.38	0.10	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.48	0.00	0.00
Red	0.00	0.00	0.00	0.00	0.06	0.03	0.00	0.00	0.03	0.06	0.00	0.00	0.00	0.82	0.00
Cream	0.06	0.00	0.06	0.09	0.00	0.03	0.00	0.00	0.06	0.00	0.00	0.00	0.00	0.00	0.70

Fig. 2.22.: Confusion matrix for testing set based on trained random forest classifier. Overall, our results are showing that the algorithm is able to produce reliable color descriptions. Some confusions can be observed between silver and gray (texture difference), tan and gold (fuzzy definition), and tan and cream (fuzzy definition).

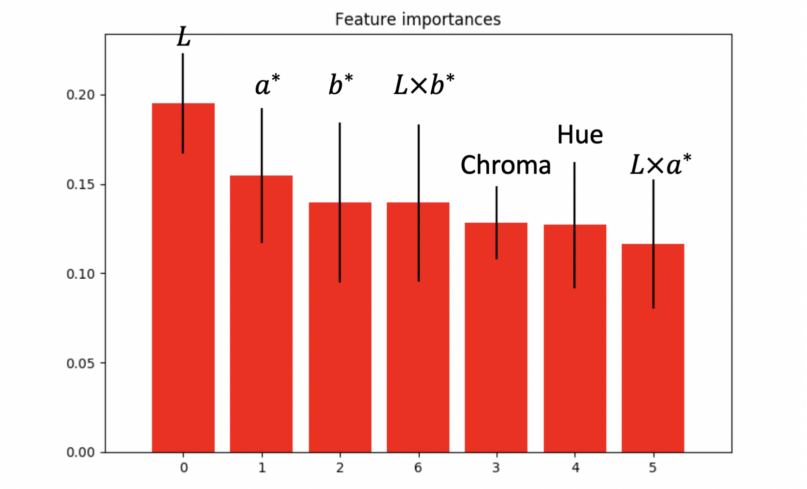


Fig. 2.23.: Feature importance for our trained random forest classifier.

2.8 Pipeline Integration

In this section, we discuss the final integration of the AGCES. In the previous sections, we develop different modules for functionalities including segmentation or semantic segmentation, color extraction, and color mapping. The final step of the system is to integrate all the modules together, to create a complete pipeline that produce color names, given a color input image.

2.8.1 Pipeline Assembly

The entire AGCES pipeline contains three modules: segmentation, color extraction, and color mapping. We show the pipeline in Fig. 2.24 with all the intermediate results. As stated before, Gaussian Mixture Model (GMM) does a good job for color extraction, and the Random-Forest-based color mapping system is developed and tested to better map color coordinates to color verbal descriptions. Our color classification model is trained using data from Reversed Color Naming Experiment and saved for further inference. And GMM model is built for each semantic label.

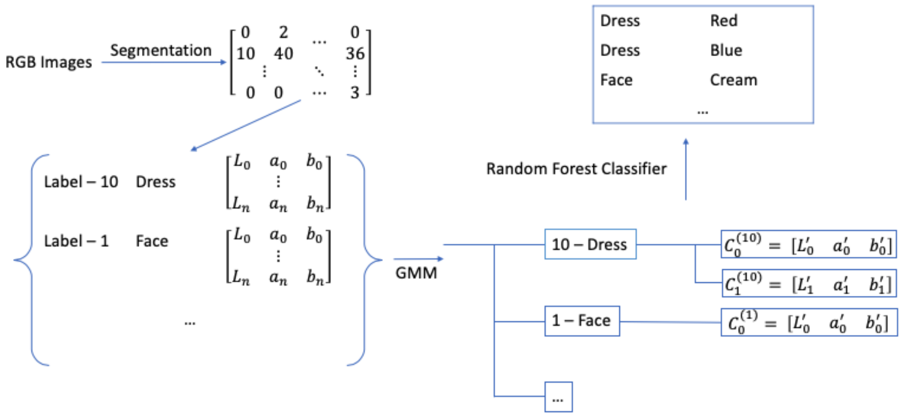


Fig. 2.24.: The final pipeline that combines segmentation module, color extraction module, and color mapping module. We show all intermediate results between each stage. Note that the numeric label is only for demonstration and is not corresponding to the true semantic segmentation in the rest of the work.

For the segmentation module, we use Deeplab segmentation module as a demonstration. The difference is that for clustering-based segmentation module, there wouldn't be semantic labels associated with the foreground.

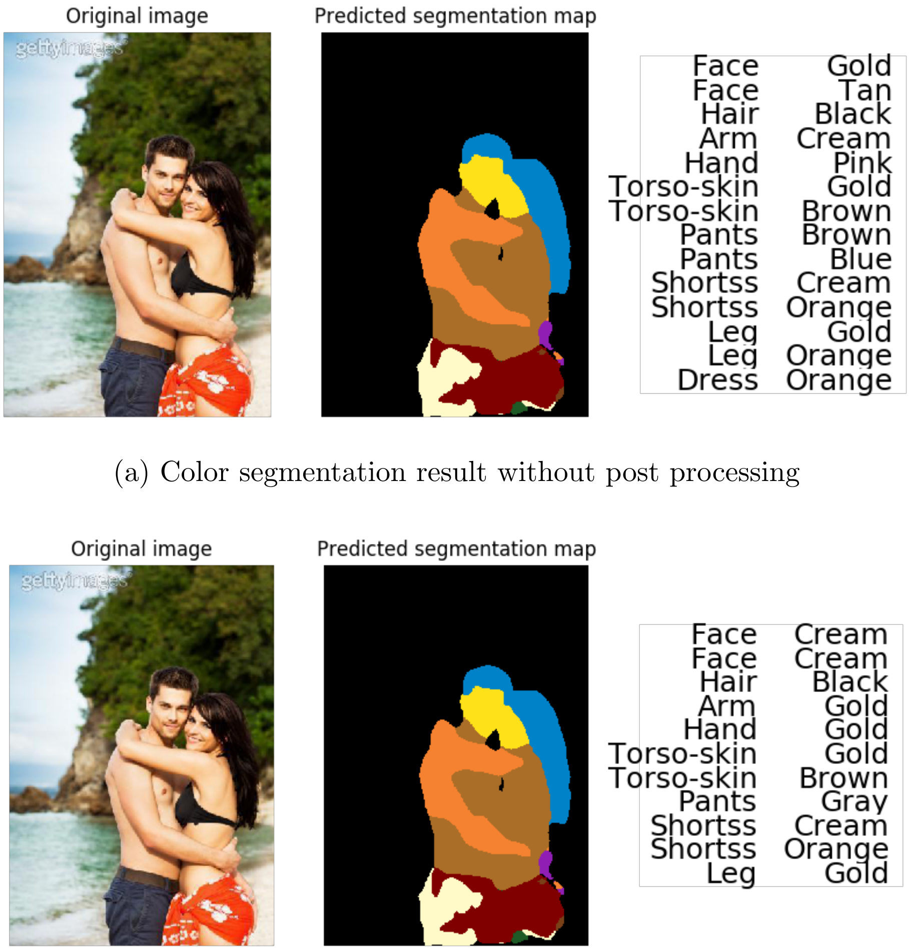
2.8.2 Segmentation Map Post Processing

As previously observed, the segmentation map tends to bleed over the object boundary, and created a "bloated" segmentation map. Also, the model will generate some small noisy segments within large segment. This can cause some problems for color extraction:

- Generally speaking, edges occur where the color of the neighboring pixels changes drastically, and the "bloated" segmentation map will include some pixels on both sides of the edges. Therefore, the color extraction results can be altered by introducing these outliers.
- Small satellite segments tend to be incorrect prediction and have irregular colors, therefore, we can boost accuracy by eliminating incorrect labels. Moreover, small segments will introduce more complexity to compute the image color.

Therefore, the segmentation map should be further processed before color extraction. Specifically, we want to shrink the segment boundaries using image morphology. We use erosion with disk mask with radius of 2 to process each semantic label. Then, we use the reduced collection of pixels for color extraction and color mapping.

Fig.2.25 shows a comparison between with and without post processing. The figure shows that the post processing eliminates some of the noisy segments (dress) and corrects some color labels, for example, face color (gold/tan v.s. cream) and leg color (gold/orange v.s. gold). Overall, we see the improvement and benefit by deploying labelmap erosion into our pipeline.



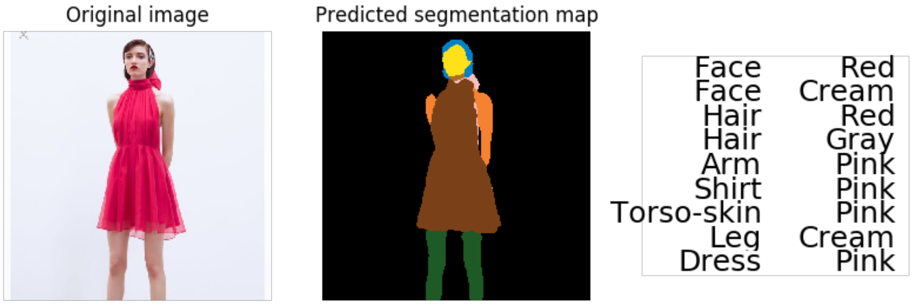
Face	Gold
Face	Tan
Hair	Black
Arm	Cream
Hand	Pink
Torso-skin	Gold
Torso-skin	Brown
Pants	Brown
Pants	Blue
Shortss	Cream
Shortss	Orange
Leg	Gold
Leg	Orange
Dress	Orange

(a) Color segmentation result without post processing

Face	Cream
Face	Cream
Hair	Black
Arm	Gold
Hand	Gold
Torso-skin	Gold
Torso-skin	Brown
Pants	Gray
Shortss	Cream
Shortss	Orange
Leg	Gold

(b) Color segmentation result with post processing

Fig. 2.25.: Comparison between with and without segmentation map post processing. (a) shows the original image (left), segmentation map from Deeplab model (middle), and object color table without post processing (right). (b) shows the same original image and segmentation map, but the object color table with segmentation map post processing (right).

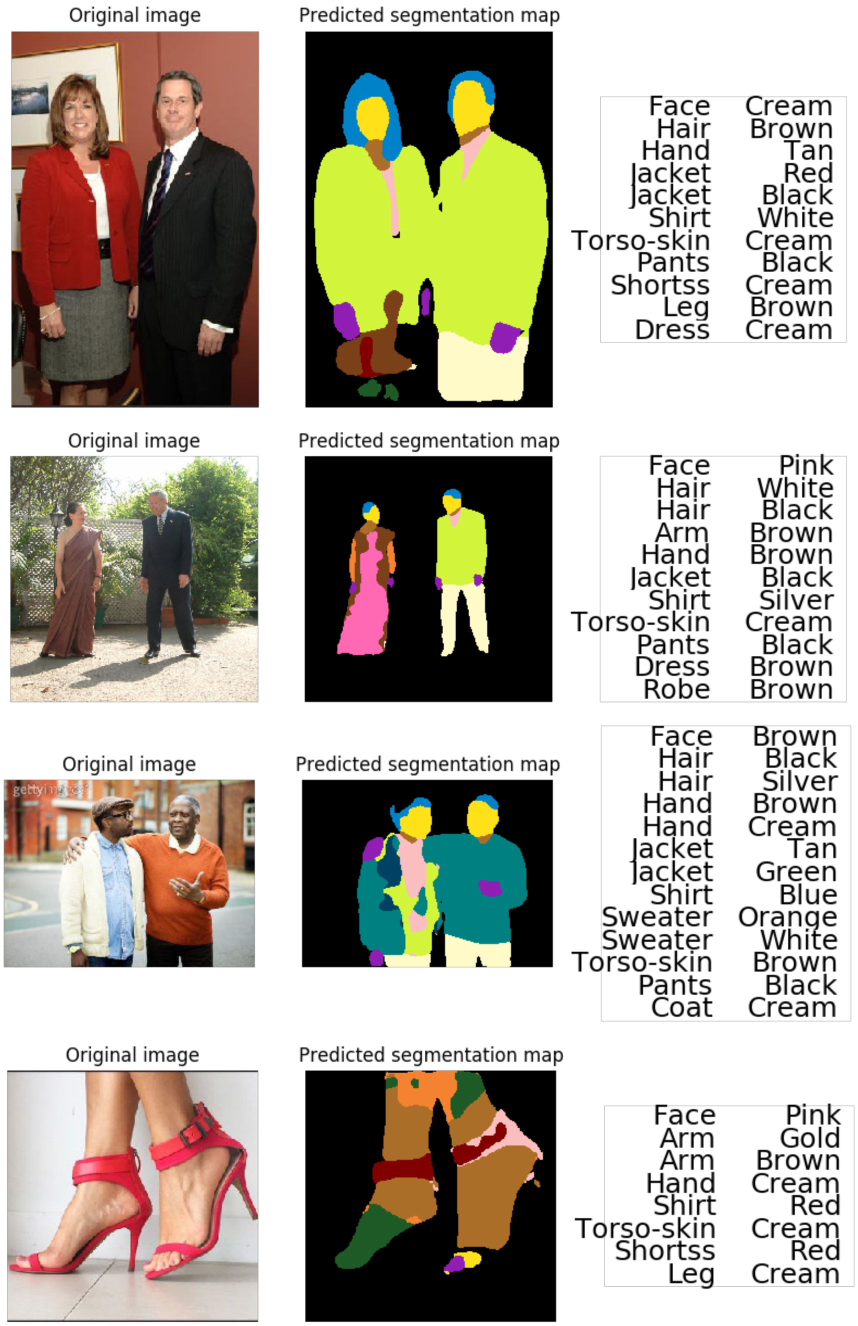


Face	Red
Face	Cream
Hair	Red
Hair	Gray
Arm	Pink
Shirt	Pink
Torso-skin	Pink
Leg	Cream
Dress	Pink

Fig. 2.26.: Example of AGCES pipeline on Poshmark product portrait with human model. We can see that our pipeline is able to extract the garment information and name the color accurately.

2.8.3 Results

In this section, we show results from the pipeline. In Fig. 2.26 and 2.27, some examples of segmentation results and object color tables. We find that our pipeline is able to extract the semantic label and find corresponding color name for fashion portraits as we defined before. But due to the limitation of semantic segmentation, the pipeline assumes all the pixels of the same label belong to the same instance. Therefore, for images where multiple persons wearing garments of different colors, the GMM model tends to summarize the average color, which is less accurate.



Original image

Predicted segmentation map

Face	Cream
Hair	Brown
Hand	Tan
Jacket	Red
Jacket	Black
Shirt	White
Torso-skin	Cream
Pants	Black
Shortss	Cream
Leg	Brown
Dress	Cream

Original image

Predicted segmentation map

Face	Pink
Hair	White
Hair	Black
Arm	Brown
Hand	Brown
Jacket	Black
Shirt	Silver
Torso-skin	Cream
Pants	Black
Dress	Brown
Robe	Brown

Original image

Predicted segmentation map

Face	Brown
Hair	Black
Hair	Silver
Hand	Brown
Hand	Cream
Jacket	Tan
Jacket	Green
Shirt	Blue
Sweater	Orange
Sweater	White
Torso-skin	Brown
Pants	Black
Coat	Cream

Original image

Predicted segmentation map

Face	Pink
Arm	Gold
Arm	Brown
Hand	Cream
Shirt	Red
Torso-skin	Cream
Shortss	Red
Leg	Cream

Fig. 2.27.: Examples of AGCES pipeline. We can see that for images with human body the results are more rebust for segmentation and able to extract color accurately. There are misclassification for some Poshmark images that only showcase the item.

2.9 Segmentation Algorithm Evaluation

To further investigate the performance of our two different methods, traditional clustering-based segmentation method and DeepLab-based semantic segmentation, on real life amateur fashion image dataset.

2.9.1 Data Collection and Segmentation

We obtained a collection of about 85 fashion images from the online fashion marketplace, namely Poshmark. The images are collected on Poshmark website, and to make sure that all the major categories (shirts, shorts, pants, dresses, sweaters, coats, bags, shoes, and other accessories) are covered evenly in our evaluation, we collected 8-9 fashion listings from the website's search results. Then, images are labeled in image semantic segmentation tool in Matlab. Note that:

- The distribution of fashion item categories in the dataset for this quantitative evaluation does not necessarily reflect the real category distribution on the website.
- Images are retrieved from the latest results from the search engine. This is to make sure that our data reflects the most recent online fashion image trends.
- When multiple images are provided in the listing, the thumbnail image is chosen, as the thumbnail is regarded as the first impression of the listings.
- The images are labeled as foreground/background, and the output is binary images.

Some of the images collected in this dataset are shown in Fig. 2.28, and their corresponding segmentation labels visualized as binary segmentation masks are shown in Fig. 2.29. It can be seen that the images collected from real life fashion marketplace can be different from our underlying assumptions for clustering and network based segmentation algorithms: the segmentation algorithm assumes input images are fashion product portraits (single item, no complicated layout and styling), and the neural network approach is trained on human parsing dataset, which does not include images without human bodies and single items. Therefore, it is important to benchmark both algorithms in the real life dataset. In the next two subsections, we will show benchmark results of both algorithms on the evaluation dataset.

2.9.2 Majority Selection Module Ablation Study

As mentioned in the previous section, in our clustering segmentation pipeline, we use the majority vote scheme to render the final decisions (on which segments are foreground and which are background) based on the judgements from three different clustering algorithms: KMeans, Meanshift, and Agglomerative algorithm. And before combining all the clustering labels, we also correct the clustering labels from each algorithm by setting the label of the top left corner pixel to 0.

To demonstrate the effectiveness of the final implemented system, it is desired to do segmentation performance comparison against our evaluation dataset. Specifically, we look into following algorithm configuration variation:

1. **Effectiveness of selected clustering algorithms and configurations.** Here, we want to compare the effectiveness of the configurations of the agglomerative clustering algorithms.

2. **Effectiveness of majority selection scheme.** To show the advantage of using three clustering algorithms instead of only one, we compare using each individual clustering algorithms with using majority vote to combine all three algorithm.

3. **Effectiveness of our clustering label correction process.** Like mentioned in the previous section, clustering algorithms only produce labels for separating data points into different groups, and the labels generated do not carry any meaning. Therefore, we need to correct the labels by assigning the label

of top-left pixel as 0, assuming the top left pixel is always background. We further investigate the effectiveness of this assumption and correction process

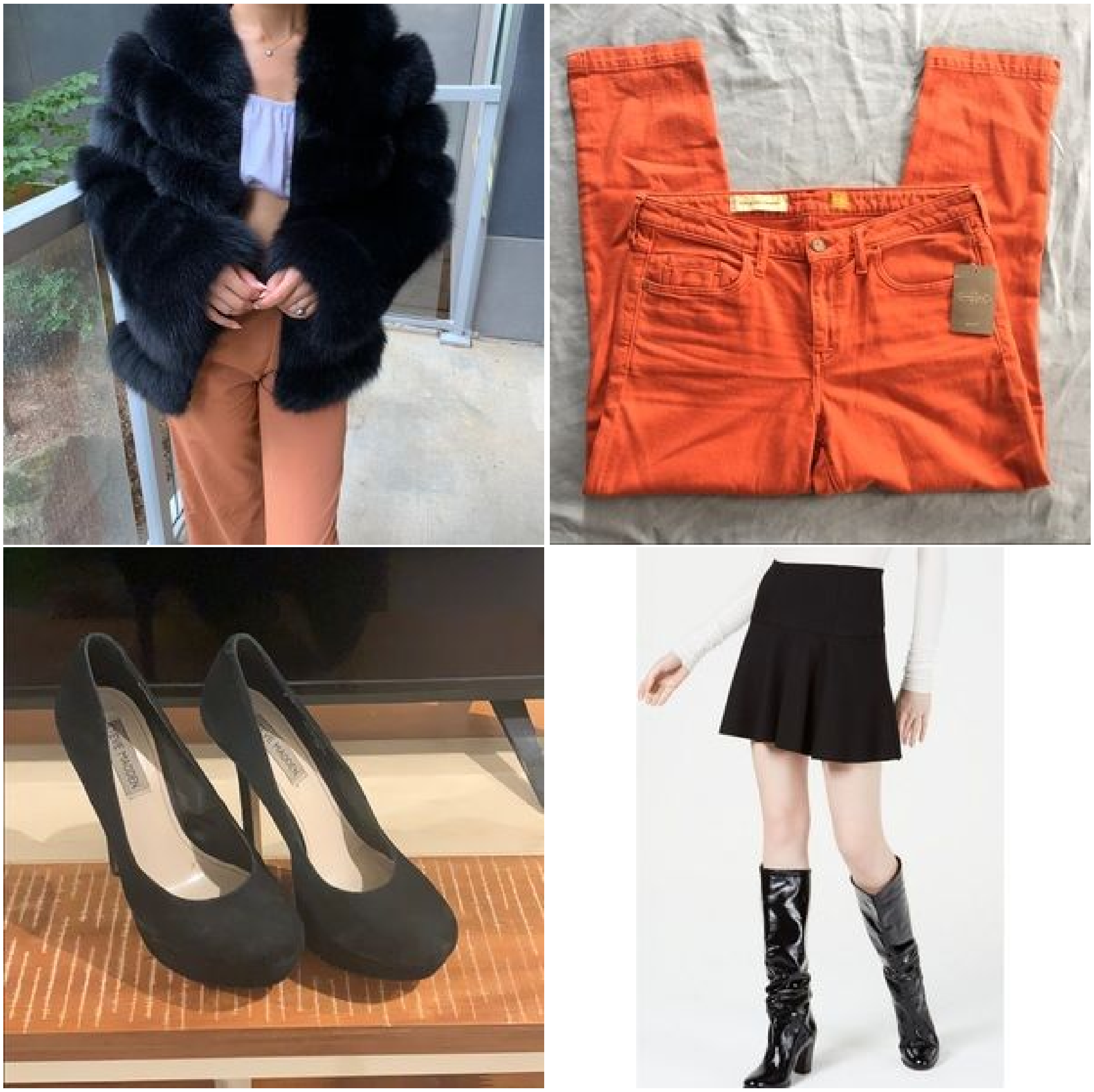


Fig. 2.28.: Examples of images collected from Poshmark website for evaluation. A number of characteristics can be found: first, the images are not necessarily only centered around the item for sale; garments sometimes are folded and styled for display, instead being worn on model. These unique characteristics can potentially bring challenges to our both clustering-based and network-based algorithms.

on kmeans. Note that, although our quantitative investigation has been only conducted to kmeans, this also applies to all other clustering algorithms.

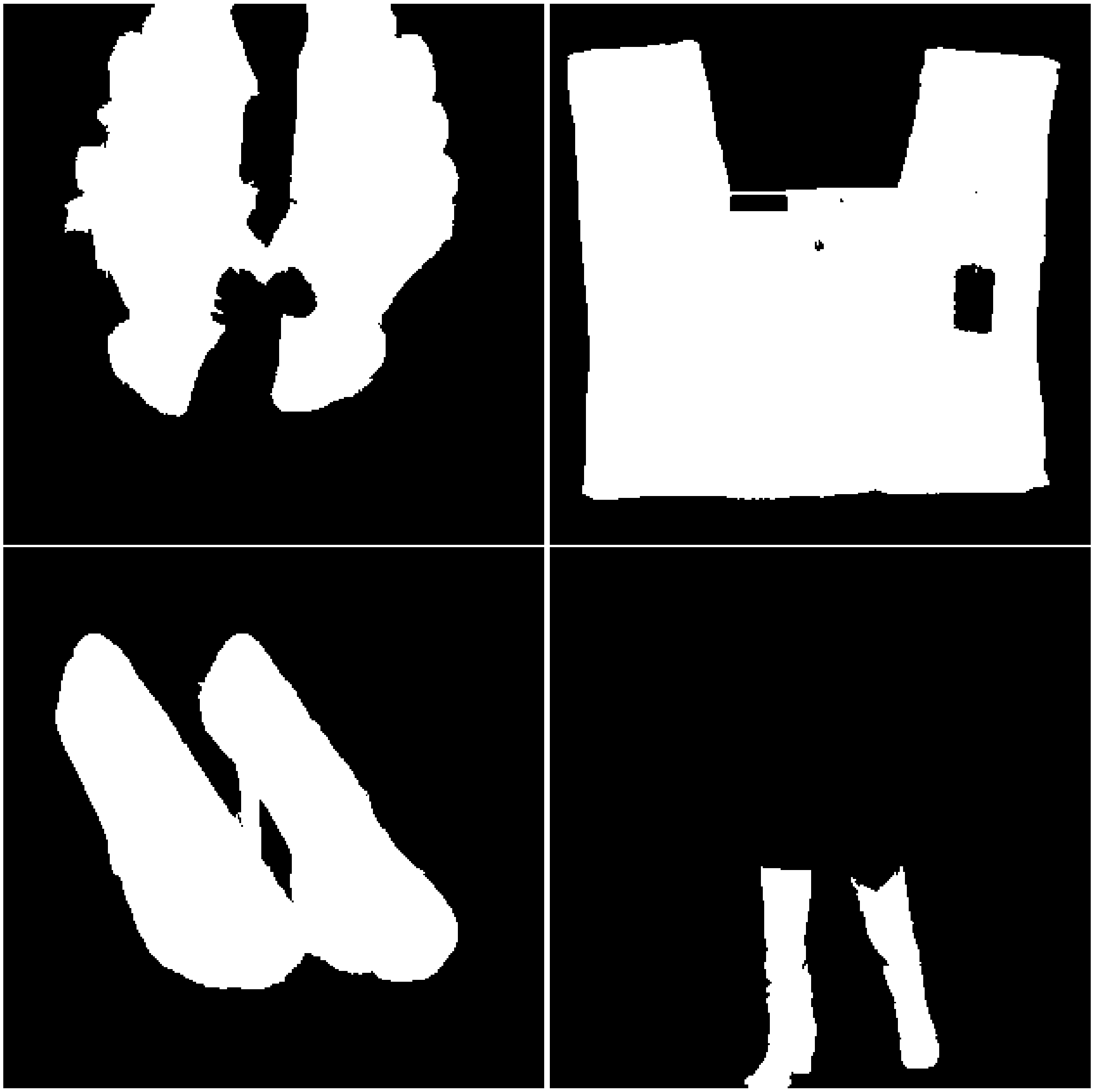


Fig. 2.29.: Binary segmentation masks for images shown in Fig. 2.28. The white pixels indicates the pixels of garment for sale in the listing, and the black pixels are ones of the background. Note that garment items other than the on sale items are considered as background in our segmentation (for example, the second and forth image).

Table 2.4.: Comparison of different algorithm configuration as segment selection method in clustering algorithm (mIoU). Here we benchmark different linkage types used in the agglomerative algorithm. We also benchmark the effectivness of clustering label correction process on KMeans. Note that the same process is used for all other configurations.

Clustering Algorithm		mIoU
Agglomerative	Ward	71.36%
	Single	45.03%
Meanshift	–	50.46%
KMeans	Corrected	71.86%
	Uncorrected	46.68%
Majority selection		72.76%

The segmentation comparison is presented in Table. 2.4. First, we show the segmentation accuracy of using agglomerative, meanshift, and kmeans clustering algorithms individually. It can be observed that our algorithm achieves higher mIoU by using three algorithms and majority selection scheme. Secondly, it can be observed that agglomerative algorithm with Ward linkage has higher segmentation accuracy across all individual clustering methods. Thirdly, the clustering correction process considerably increases the accuracy of clustering algorithms.

Fig. 2.30 further shows some examples of the difference of segmentation performances.

2.9.3 Segmentation Algorithm Benchmark

We show the segmentation comparison between clustering algorithm and neural network in Fig. 2.31 and Fig. 2.32. It can be observed that the clustering-based algorithm can generally extract the fashion image subjects accurately when the image

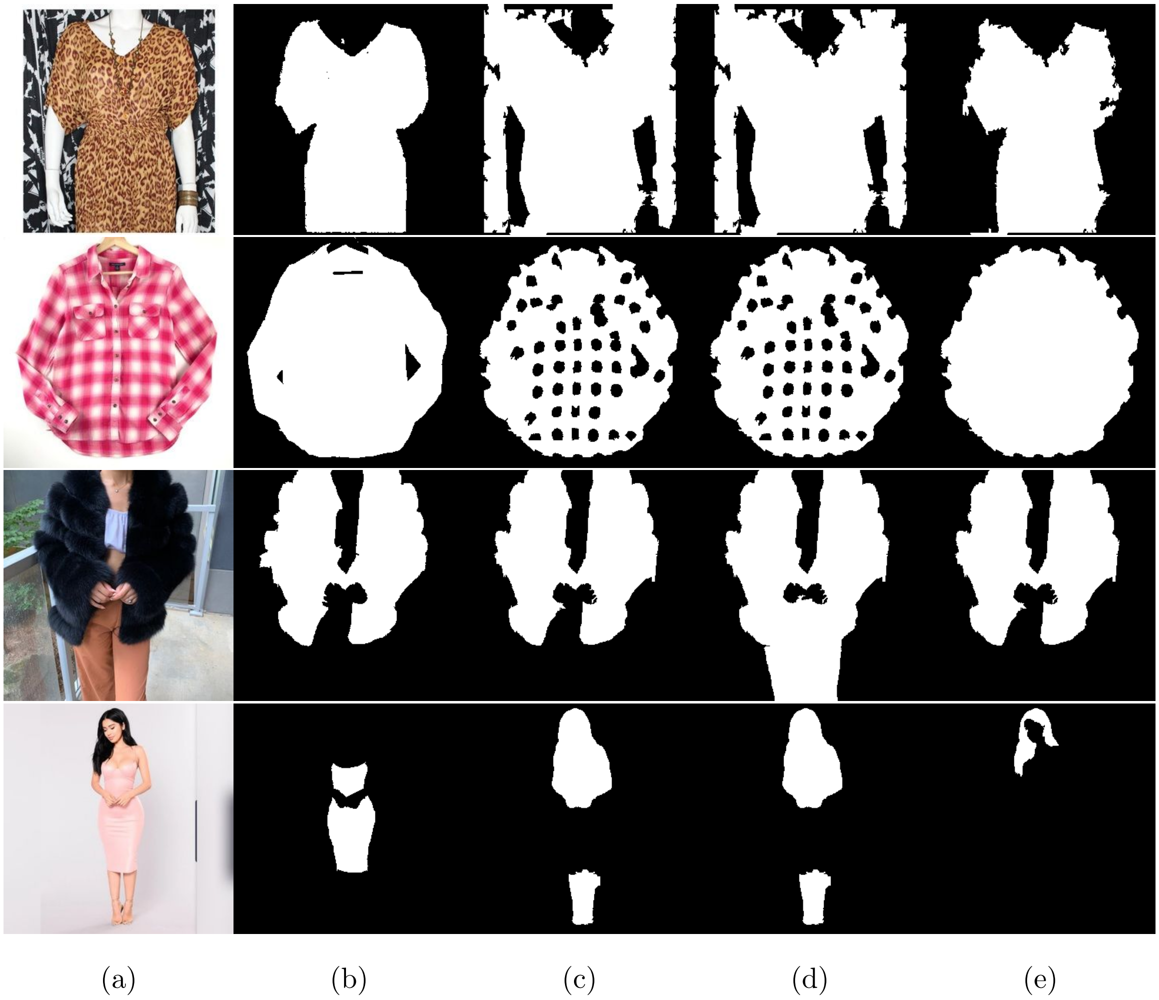


Fig. 2.30.: Additional segmentation results comparisons between agglomerative (ward), kmeans, and meanshift. (a) original images, (b) ground truth segmentation labels, (c) agglomerative algorithm, (d) kmeans algorithm, and (e) meanshift algorithm.

has a clear background. However, because the algorithm is human-agnostic and fashion-agnostic, human body and other garments can be included. On the other hand, we can see that DeepLab network works very well for images where garments are worn or demoed by humans or mannequins. However, networks are having worse performance on images of bags, shoes, and jewelries. There are several reasons:

1. Items like bags, jewelries, and shoes have lower occurrence in the training dataset. Therefore, segmentation results for these categories inherently have lower accuracy than other categories like dresses and pants.

2. In the training set, these items are usually not the focus of the image, where accessories mostly appear small and blurry. Therefore, the garment appearance in product portrait is very different from in training dataset.

Here, we run both image segmentation algorithms on the evaluation dataset. In our evaluation, we find the foreground in both ground truth mask and prediction mask and calculate the mean IoUs of each image. However, since the DeepLab network produces more sophisticated segmentation map including background, garments, and human body parts, we need to transform semantic segmentation results to binary segmentation. We combine the body part segmentation labels with background, and treat all clothing garments labels as foreground.

Furthermore, to investigate algorithms' performance difference between different types of the images, we tag the images in the dataset in which garments are modeled or displayed by human model or mannequins, and calculate mIoUs on modeled images and non modeled ones. Theoretically, images with modeled garments tend to have more than one items, which can be challenging for algorithms to separate different items and skin colors.

We show results in the Table 2.5. Our Poshmark evaluation dataset consists of 30 modeled images and 55 non modeled ones. We can see that, for fashion product portraits, clustering algorithm achieves a higher accuracy than deep lab segmentation that is trained on human parsing dataset due to the lack of human body cue. On the

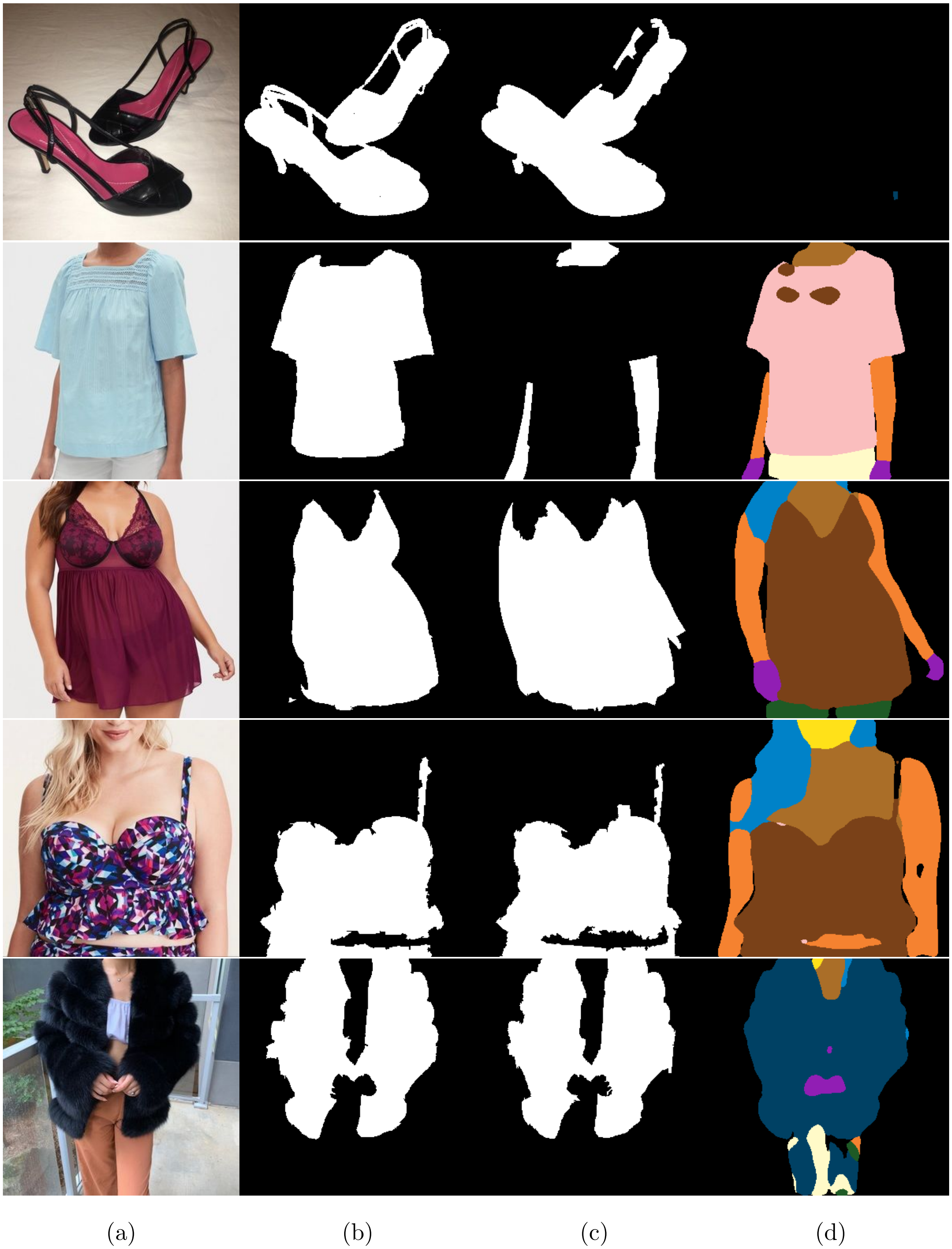
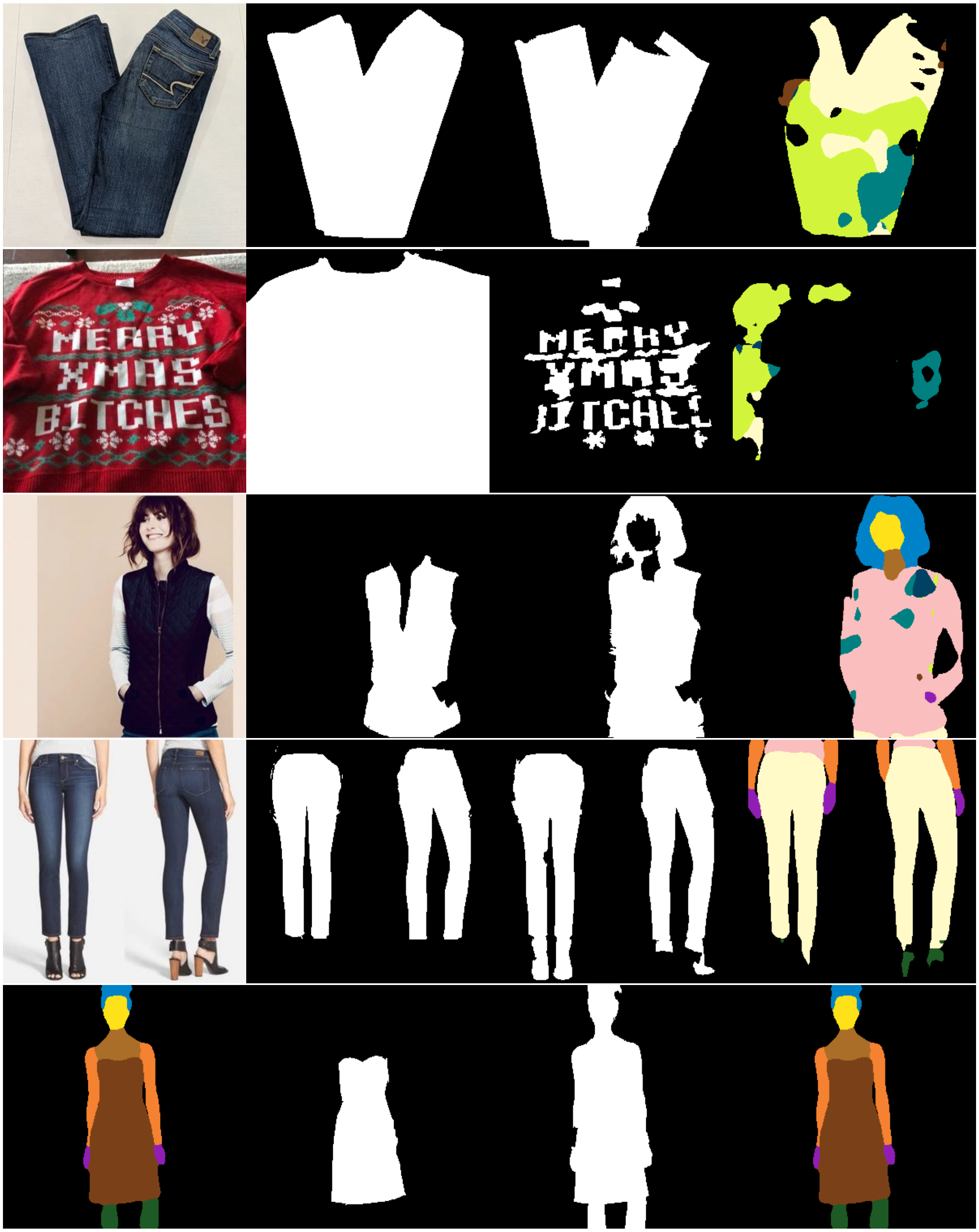


Fig. 2.31.: Segmentation results comparisons between two algorithms. (a) original images, (b) ground truth segmentation labels, (c) clustering segmentation algorithm, and (d) prediction from neural network.



(a) (b) (c) (d)

Fig. 2.32.: (Continued) Segmentation results comparisons between two algorithms. (a) original images, (b) ground truth segmentation labels, (c) clustering segmentation algorithm, and (d) prediction from neural network.

Table 2.5.: Average segmentation accuracy comparison between clustering algorithm and trained DeepLab network (mIoU).

	Clustering	MHP Network
Non modeled	80.41%	49.68%
Modeled	58.73%	78.09%
Overall	72.76%	60.07%

other hand, semantic segmentation algorithm proves to be more accurate for separating garment items from the background for modeled images. These two algorithms show unique advantages in different image types, and overall, because of the composition of the dataset, the clustering algorithm outperforms the semantic segmentation network trained on MHP.

2.10 Conclusion and Future Work

In this project, we propose a system that autonomously extracts the garment from the fashion product portrait photographs and assigns the color of the garment with a verbal descriptive name. This system contains three modules: first module takes the input image and partitions the image into perceptually meaningful segments; second module utilizes the designed the features to group all the garment segments and others; the last module extracts the mean colors using Gaussian Mixture Model, and matches the color coordinate with corresponding color names predefined on the fashion market website. We evaluate the performance of segmentation module case-by-case, and that of color extraction module by statistical performance measurement. Our segmentation module can achieve reasonably good semantic segmentation result for fashion product portraits, and our color extraction model achieves over 70% accuracy based on our dataset.

Fashion color naming has been a crucial yet under studied problem for the online fashion market. In this work, we explore a new adaptive approach to match color values to descriptive color names. A reversed color naming experiment is proposed to collect color naming data; and a new random-forest based color classification system is used.

Our random forest classifier shows great improvement over our previous model and achieves high accuracy of color name prediction. Furthermore, the algorithm has the potential of further learning new fashion color vocabulary in the future if new trainable data is given.

The accomplished work we mentioned above helps to identify the garment region in fashion product portrait images, and also produce accurate color naming on the extracted garment region. To further improve our system practicality and capability, the following dissertation research is enumerated below:

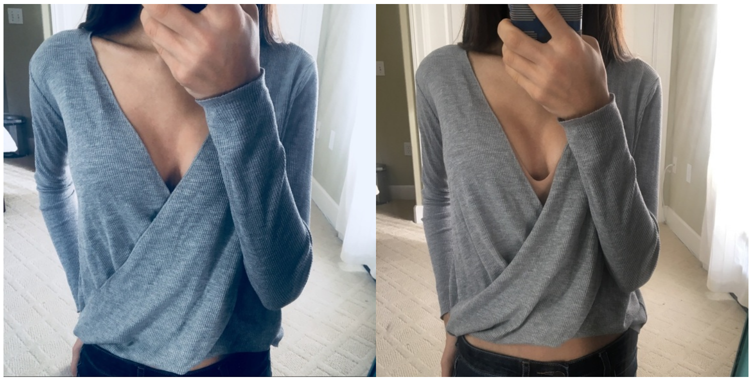
- **Image color calibration.**

  As we stated in the previous section, many of the sellers on the P2P fashion marketplace do not possess sufficient professional photography skills and equipment to produce color-wise accurate images consistently. We provide an example in Fig. 2.33. Although the color reproduction accuracy does not necessarily deteriorate the segmentation results, it significantly effects the color conclusion drawn by the algorithm. To further improve the accuracy of color extraction, two facets should be considered:

1. For single image, white balance should be calibrated and restored if necessary.
2. Further improve semantic segmentation to process fashion images with various view points, with/without human model, and instance segmentation.
3. If the listing provides more than one images, potential color calibration cross multiple images can be implemented.

- **Adaptive and active learning for ever-changing fashion trends.**
  As mentioned before, a very interesting aspect of fashion is that there are always new trends and styles coming up in the fashion world.

  1. New fashion color names can be learned dynamically from the fashion community and expert. Another option is to match different garments that share the same color.
  2. New fashion item styles can also be recognized actively. This can further enhance our semantic segmentation performance by eliminating some ambiguities.



(a) (b)

Fig. 2.33.: Example of the variation of color editing done by user. 2.33a and 2.33b are for the same listing, and these two shots are for the same item. Because of the color editing done by the seller, it is hard to tell the real color of the item. Source: Poshmark.com

3. NEURAL NETWORK APPLICATIONS ON INKJET PRINTING QUALITY AND FORENSICS

3.1 Overview

In Chapter 2, we conduct a thorough investigation on fashion image analysis. Like other mostly studied image understanding problems, the mission of our fashion image understanding is to recognize human subjects and fashion objects and to extract information from real world images. Our research shows different approaches to incorporating machine learning methodologies in different complexity scales (both traditional machine learning algorithms and neural networks), achieving high image segmentation and color extraction accuracy with limited or no suitable data sources.

Chapter 3 focuses on image understanding problems on inkjet printing analysis, where the explicit semantic content is not the end goal of research project. Inkjet printing analysis aims to learn the printing quality and extract unique dot pattern of inkjet printer that can be used to identify the source machine for forensics and anti-counterfeiting purposes. Neither the print quality nor the dot pattern is a localized feature; implicit image quality metrics and printing forensics are related to texture and pattern that occurs throughout the image.

Despite the huge difference on the nature of two problems mentioned above, they share challenges of data collection. For fashion analysis, collecting data requires less of domain expertise, and it can be done by a larger pool of human subjects that have had related experience before. Therefore, data collection campaigns can be conducted on a large scale using crowd sourcing platforms like Amazon's Mechanical Turk [64]. However, for problems like medical imaging and inkjet imaging, image data and ground truth collection are usually done by domain experts like medical doctors and image scientists, and it is unrealistic to find a large pool of experts to do a huge amount of data labeling with a fairly tight budget. Hence, datasets for such uses are usually much smaller.

To circumvent training large neural networks for every specific use, our research suggests that pretrained neural network that have learned complex features from millions of images from datasets like ImageNet [3] or COCO [2] can be used as human vision feature generators for print image analysis. This allows us to use pre-trained model and directly use features generated by the network in lieu of transfer learning or network fine tuning. We illustrate the implementation on inkjet printing quality analysis in Section 3.2, and printer forensics in Section 3.3.

3.2 Image Quality Assessment Using Computer Vision

3.2.1 Introduction

Image quality assessment is very important for electronic imaging systems. It allows us to predict the quality of the image, so that it can be maintained, controlled and possibly enhanced before production or further processing [65]. Hence, a reliable image quality paradigm is crucial in the development of image processing systems.

There have been many efforts developing image quality assessment methods and tools in the imaging community. In general, the methodology of image quality assessment can be divided into two schools: subjective methods [66], where human viewers are involved to evaluate the quality of images, and objective methods [67], where the numerical metrics are calculated from the image. Theoretically, subjective methods tend to be more accurate and reliable, as images are ultimately viewed by human viewers. However, subjective methods are not practical due to time and labor cost. Thus, objective image quality metrics are highly desired in the imaging industry to predict the quality of an image as close as possible to the subjective assessment.

Traditional objective image processing-based image quality assessment pipelines [68] for image noise use filter-based techniques to focus on the structure within the certain (pre-determined) frequency range, and analyze the image linearly channel by channel. However, such approaches based on linear combination of filters may not be best suited for the task of modeling human perception of printed image noise.

Neural network based computer vision models have shown great potential to mimic human vision in many fields, and increasing research efforts are being put into using computer vision to do image quality assessment [69]. In this paper, we use computer vision (CV) models to build appearance-based metrics to evaluate the visual image quality of printed images, in particular, the micro uniformity [70] in a print. We aim to characterize the visual noise that is perceived by human viewers from printed images. Our model leverages recent developments in deep neural networks and machine learning to mimic noise perception of the human vision system. Our experiment

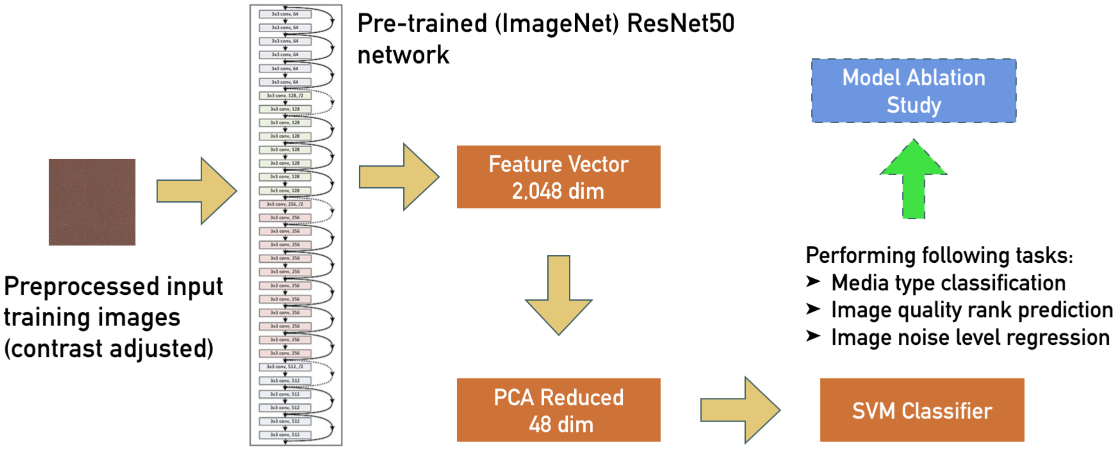

Pre-trained (ImageNet) ResNet50 network
Model Ablation Study
Feature Vector 2,048 dim
Preprocessed input training images (contrast adjusted)
Performing following tasks:
➤ Media type classification
➤ Image quality rank prediction
➤ Image noise level regression
PCA Reduced 48 dim
SVM Classifier

Fig. 3.1.: Overall framework of computer vision based image quality analysis system.

shows that the proposed model achieves high accuracy for both subjective and objective micro uniformity assessment tasks. Furthermore, to demonstrate the viability of the proposed model, we show the effectiveness of the off-the-shelf neural network features for decision making in various image quality assessment tasks.

3.2.2 Data Acquisition

Our dataset consists of 75,000 RGB image patches (224x224x3) from 600 dpi scans of full page CMYK halftone inkjet prints on 6 different papers. These 6 media types and their respective printing conditions were chosen to produce nearly identical colors but with various levels of graininess noise according to human perception and analytical image analysis. In the context of this chapter, paper type is just a convenient label for various levels (classes) of image noise.

The digital image patch used in this experiment is shown in Fig. 3.1. Note that the dimension of the image is chosen to fit in the input layer of the Residual Neural Network. Also, the contrast of the scanned images are adjusted and normalized to control the environment and reinforce our pipeline to learn from the noise and nothing else.

3.2.3 Machine Learning Based Image Quality Assessment

The modeling framework can be described as follow: We use a pre-trained Residual Neural Network ResNet50 [56], to retrieve a 2,048 dimension feature vector from an input RGB image. We then use the Principal Component Analysis (PCA) [71] to further reduce the feature dimensions to 48. Finally, the reduced feature is input to a support vector machine that has been trained for various objective and subjective image quality assessment tasks such as media type classification, visual noise metric, and IQ noise ranking.

Media Type Classification

As noted previously, the media type is a surrogate for different levels of image noise. Due to the different physical and chemical attributes of the media and the physics of ink-media interaction, the printed images have different IQ noise appearance on different media. Therefore, we aim to use discriminant features extracted from a Convolution Neural Network (CNN) to pinpoint the source media of image.

We train a multi-class Support Vector Machine (SVM) classifier to predict the correct media types for given images. We split the dataset into 80% and 20%, and train the classifier on the 80%. We show the results in Fig. 3.2. It shows that our model achieves higher than 95% accuracy across all paper types. This suggests that the noise patterns for each paper type are distinct and the CV model successfully learns to recognize them. Compared with traditional IQ metrics, where only real values are produced as the representation of the quality, our model can further approach the human vision system by reflecting the type of source based on distinct image visual appearance.

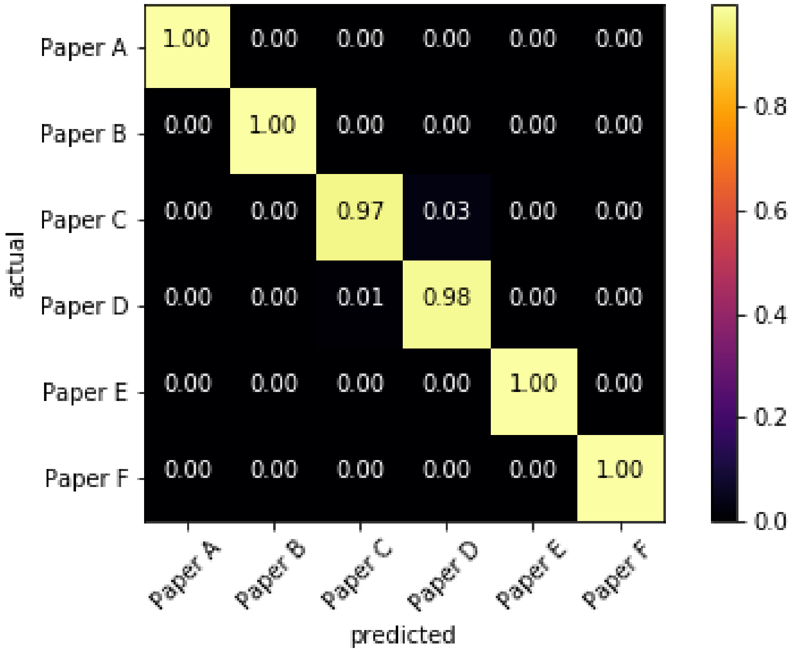


Fig. 3.2.: Confusion matrix on test data for media classification.

Image Quality Ranking

In addition to the media classification capability, the proposed model also shows ability to perform pair wise image quality comparison, and it can produce an overall subjective IQ rankings through out all the studied images.

First, we had a human expert rank the images in order of preference from 0, the best image quality, to 5, the worst image quality. We verified that the ranking is generally consistent for images of the same media type, so we assigned the same rank to all images of a media type. Then, we trained a set of five binary SVM classifiers, and combined them to produce discrete number from 0 to 5 as image quality ranking. Fig. 3.3 shows the final prediction accuracy. Overall ranking prediction is higher than 95%, except for paper C which appears have some confusion with paper D. Coincidentally, papers C and D are from the same manufacturer.

Note that although the rankings are given by experts by paper types, hence the rankings are directly associated with the paper types, there are differences between media classification and image quality ranking prediction. For media type classification, we consider every class to be independent, and there is no correlation assumed between classes. However, the pair wise ranking prediction is ordinal regression, and the algorithm needs to find the correct order based on image quality and find the right boundaries between two classes that have similar image quality performance. Fig. 3.3 also shows the difference between media type classification and ranking prediction. Comparing Fig. 3.2 and Fig.3.3, we can see that ranking prediction has more confusion between papers C and D, but little confusion can be observed in the media type classification.

Image Noise Level Prediction

Finally, in the context of objective image quality analyses, we consider the ability of our image quality assessment model to quantitatively predict the noise level. We

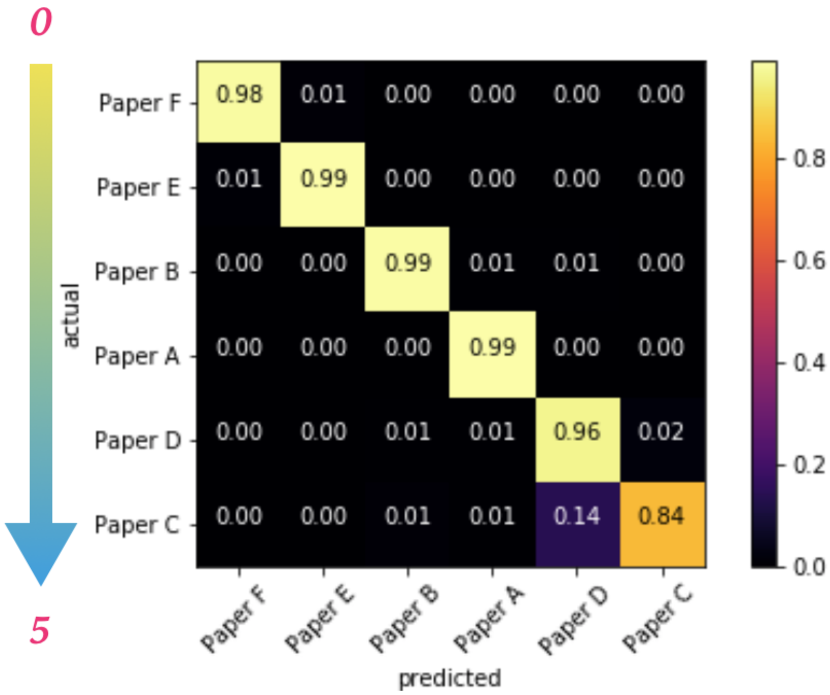


Fig. 3.3.: Confusion matrix on test data for image quality ranking. Here we keep all paper types in order of image quality. 0 means the best image quality, and 5 means the worst image quality.

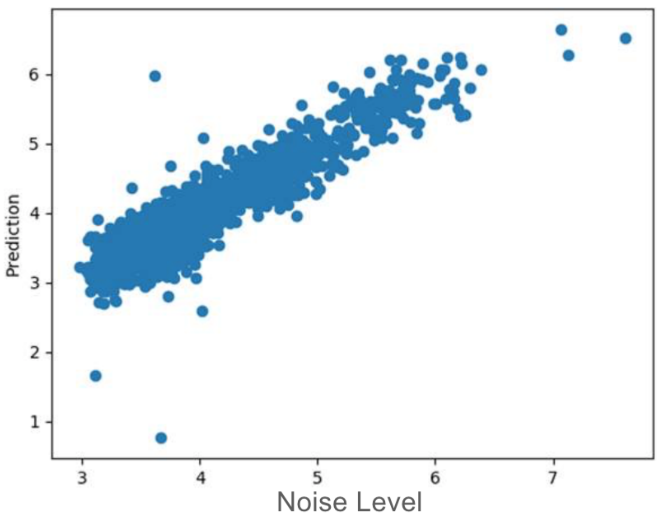


Fig. 3.4.: Plot of ground truth noise measure vs prediction from noise level regressor. The x-axis is the ground truth, and y-axis is the prediction from our regression model.

use an industry standard image quality tool to measure the visual high frequency noise level from patches, and trained a SVM regression model to predict it.

The regression results on test data are shown in Fig. 3.4. A linear correlation between ground truth and prediction can be observed. The R-Squared score of our regression model is about 0.86, suggesting a good agreement between objective metric and metric inferred from the neural network model.

3.2.4 Model Ablation Study

To validate and demonstrate the viability of our model, we aim to further examine the proposed model. Although, explaining and examining the machine learning classifier, including traditional machine learning methods and deep learning methods, has been drawing more and more attention recently, most of the classifier explainers found in the recent literature [72, 73] are focused on explaining spatially localized features for generic image recognition or text classification tasks. However, this is not applicable in our study. All the images in our experiment look roughly the same; the distinct features of interest is the noise (i.e. texture), which is not local. Therefore, we propose two approaches to explain our model: (a) by checking the separability of features generated by the pre-trained neural network and (b) by determining the confidence level change of the SVM classifier as the noise structure is altered.

Effectiveness of Neural Network Vision Features

Although the IQ assessment pipeline achieves high accuracy using the ResNet50, further testing on the effectiveness of the neural network features is still desired for following reasons:

- **Networking repurposing:** It has been shown that the hierarchical combination of convolution layers in the neural networks is capable of extracting localized features. However, for image quality assessment of noise, no localized semantic features are included in the images.
- **Dataset difference:** Our pre-trained network was trained on ImageNet [1] dataset. ImageNet contains millions of images labeled with semantic labels, and it is designed to train and benchmark classifiers to do image and object recognition. Therefore, no information regarding image quality is given during the network training.

Therefore, the effectiveness of neural network features needs to be validated for IQ assessment. Here, we propose that neural network features are effective if the extracted neural network visual feature can separate the data points by their visual appearance differences.

To demonstrate the separability of the neural network features, we plot data points by first and second PCA components, and the visualization is shown in Fig. 3.5. We can see that datapoints of the same media type are well clustered, and some

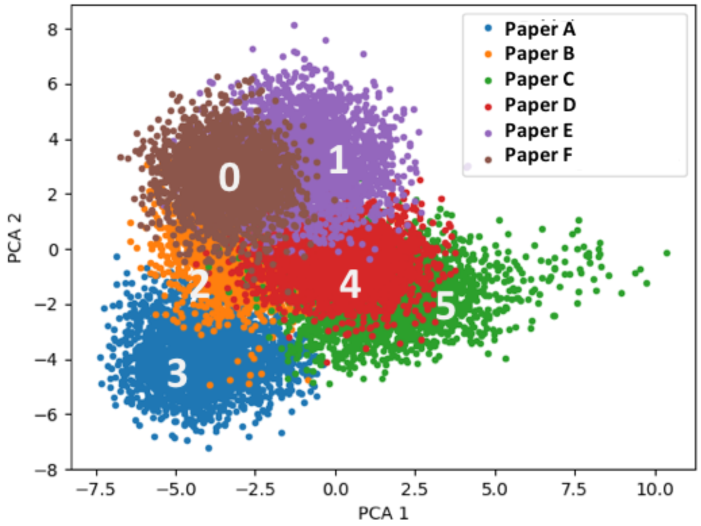


Fig. 3.5.: Plot of data points by 1st and 2nd PCA components. Data points are color coded by media types, and image quality rankings from human expert are marked on the different point clouds.

separability between different clusters can be observed. In fact, one can argue as more latent features (e.g. PCA components) are included, the point clouds will become completely separable from each other. Hence, near perfect classification results can be expected.

To further enhance the between class separability of the neural network features, we use Linear Discriminant Analysis (LDA) [74] to process the neural network features and visualize the first two LDA components in Fig. 3.6. Unlike the PCA, the LDA uses the class labels to separate the data points in orthogonal spaces. We can see that a good separability can be achieved using just 2 LDA components. The only overlap observed occurs between papers C and D, which correlates with our previous observation in image quality rankings.

Therefore, we can conclude that neural network features from pre-trained ResNet are effective in discriminating between visual appearances of images required for image quality assessment tasks.

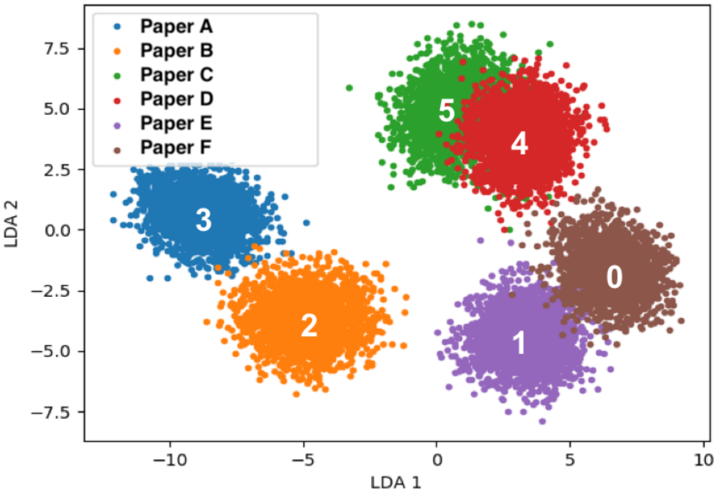


Fig. 3.6.: Plot of data points by 1st and 2nd LDA components. Data points are color coded by media types, and image quality rankings from human expert are marked on the different point clouds.

Classification Explaining

We now consider the decision making of the SVM classifier. Unlike most of the classifier explainers, which find localized stimulating regions within an image corresponding to a class label, we attempt to find the relevant frequency ranges in the image that allows the classifier to make its prediction.

Our approach is described as follow: First, we filter original images with a low pass filter. Here we chose filters with cut-off frequency ranging from 0 to 10 cycle/mm. Then, the filtered images are fed into our pipeline to predict the different media types. We monitor the confidence of the true class from the SVM classifier as the cut-off frequency is changed.

Fig. 3.7 shows the confidence level versus cut-off frequency. The following observations can be made:

1. The confidence of classifier is very low when only near-DC component is included in the image. This is not surprising because that the near-DC component should look very similar on all different media types, which is a constant tone patch of contrast adjusted color.

2. The confidence of classifier is almost at 1 when the cut-off frequency is greater that 10 cycles/mm suggesting there no discriminating features of frequency greater than 10 cycles/mm .

3. The confidence level jump occurs when the cut-off frequency is between 4-6 cycles/mm. Each media type has a distinct response corresponding to the noise frequencies that the classifier has learned to discriminate the media type.

Note that, in Fig. 3.7, the confidence curve representing paper type B does not go down to near 0 when only near DC component is included. This is caused by the nature of the multi-class classifier. Given an arbitrary image, the classifier is designed to assign one of six labels. Therefore, one of the six labels is selected as

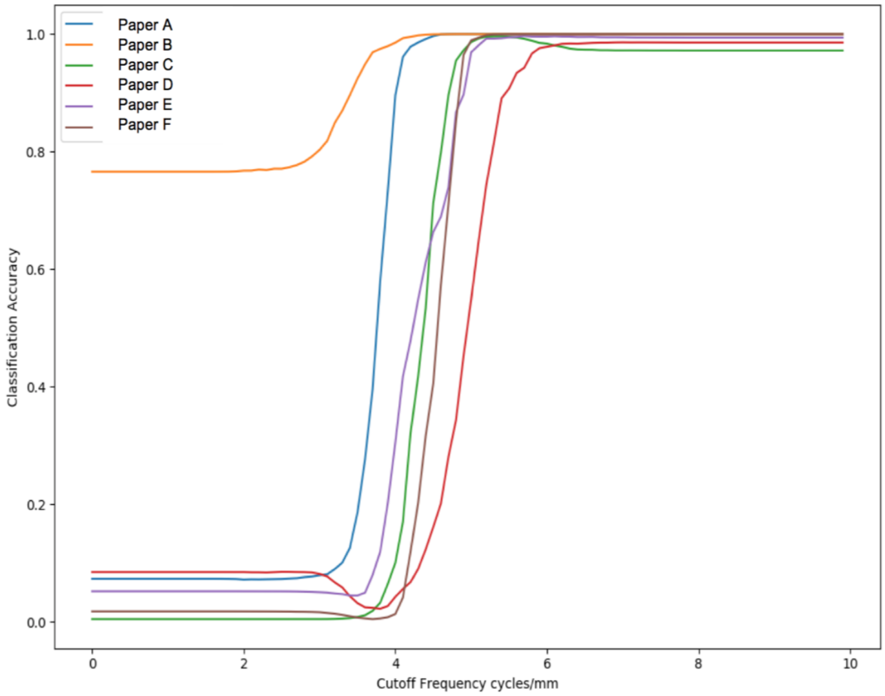


Fig. 3.7.: The plot of media type classifier's confidence response corresponding to frequency.

default classifier when the classifier is not able to produce any confident decision. In our case, paper B is the default classifier label.

To summarize, we demonstrated that the neural network feature utilized by our image quality assessment model is able to recognize the image quality differences within the test images, and our pipeline makes the classification decision based on the high frequency noise between 4 and 6 cycles/mm. We note that these frequencies are in the visual noise high frequency (VNHF) range typically associated with graininess noise.

3.2.5 Conclusion

We propose a neural network based image quality assessment model to characterize graininess noise in the printed images. Our pipeline utilizes a pre-trained residual neural network to process the images with various levels of image noise, and extracts features for both subjective and objective image quality assessment tasks, including media type classification, IQ ranking prediction, and IQ noise level prediction. The proposed model achieves high accuracy on the aforementioned IQ assessment tasks without additional neural network training and other processing. Compared with traditional filter based IQ metrics, our model is able to better approximate the human vision system and successfully assess images based on visual appearance.

Furthermore, we propose a new methodology for explaining and examining the IQ assessment classifier. First, We show that visual features extracted from neural networks are effective and highly discriminating for visual appearance and image quality. Then, we show that our trained classifiers are responding to image noise frequencies between 4 6 cycles/mm, which one often attributes to graininess.

3.2.6 Acknowledgments

We thank PARC and Xerox Corp. for supporting this research, and following people for their insights and inspiring discussions on machine learning, image quality and human vision system: Wencheng Wu and Beilei Xu of the University of Rochester, and Mojgan Rabbani of Xerox Corp..

3.3 Intrinsic Signatures for Forensic Identification of SOHO Inkjet Printers

In a 2006 report, The U.S. Secret Service estimated that 1 in 10,000 currency notes in circulation is a counterfeit[1]. In Europe, Small Office Home Office (SOHO) inkjet printers now account for over 50% of the production of counterfeit currency notes. Authorities charged with tracking counterfeit currency to its source have a range of resources at their disposal. Even if these tools do not definitively identify the particular unit that was used to produce a counterfeit note, any information that they provide can prove to be a valuable aid to the investigation.

There are many possible approaches to forensic printer identification [75]. Some of these methods require labor intensive effort by a highly trained observer. Examples include inspection of prints under a microscope [76], chemical analysis of the inks used to print the suspect currency [77], and detection of spur marks from the gears used to advance the media through the printer [78]. Other methods are based on image analysis, including analysis of the structure of printed character glyphs [79,80], analysis of page geometric distortion [81], analysis of halftone dot structure [82], and analysis of the memory contents of the suspect printing device, combined with analysis of the printed page [83].

In this section, we focus on the development of intrinsic printer features for SOHO inkjet printers that are based on the analysis of the printer dot structure in highlight regions. In contrast to Ref. [82] above, which considers only laser electrophotographic printers that use periodic, clustered-dot halftoning patterns, here we consider the spatial arrangement and size of individual ink drops in dispersed-dot, aperiodic (stochastic) halftone patterns.

We also introduce the Printer Identification System (PIS) that can autonomously identify the source machine based on a given print sample. This system is made possible by the power of recent developments in Deep Neural Networks (DNNs). A

[1] For the full report, go to: `https://www.federalreserve.gov/boarddocs/rptcongress/counterfeit/counterfeit2006.pdf`

number of effort has been made to combine DNN and halftoning. Moon *et al.* [84] characterized the Inkjet printer model using Deep Neural Networks. Ferreira *et al.* [85] used a combination of Neural Netwokr classifiers and external classifiers to do laser printer classification on low resolution scanned images. Here, we use Residual Neural Network (RNN) [56] to classify high-resolution inkjet print captures.

3.3.1 Experiment Design and Sample Acquisition

The goal of this paper is to develop features that can serve as intrinsic signatures for SOHO inkjet printers. As mentioned above, inkjet printers use dispersed-dot aperiodic (stochastic) halftoning algorithms. We specifically choose to analyze the halftone dot patterns in highlight regions, since such regions most clearly illustrate the spatial pattern of dots (each dot corresponds to a single inkjet drop), and the size of these dots.

To support this research project, we purchased 16 inkjet printers ranging in price from $30 USD to $90 USD. This printer set consisted of 9 different models from 4 major SOHO inkjet printer manufacturers, namely HP, Cannon, Epson, and Brother. In addition, for three of these models, we purchased three units of each model in order to explore unit-to-unit variations within the same printer model. One of the printers was dead on arrival, so all of our experiments were based on 15 printers. As a means of identification, each of the units was assigned an alphabet ranging from A to H, among which Printers C, D, E are from the same manufacturer, and Printers F, G, and H are from another manufacturer.

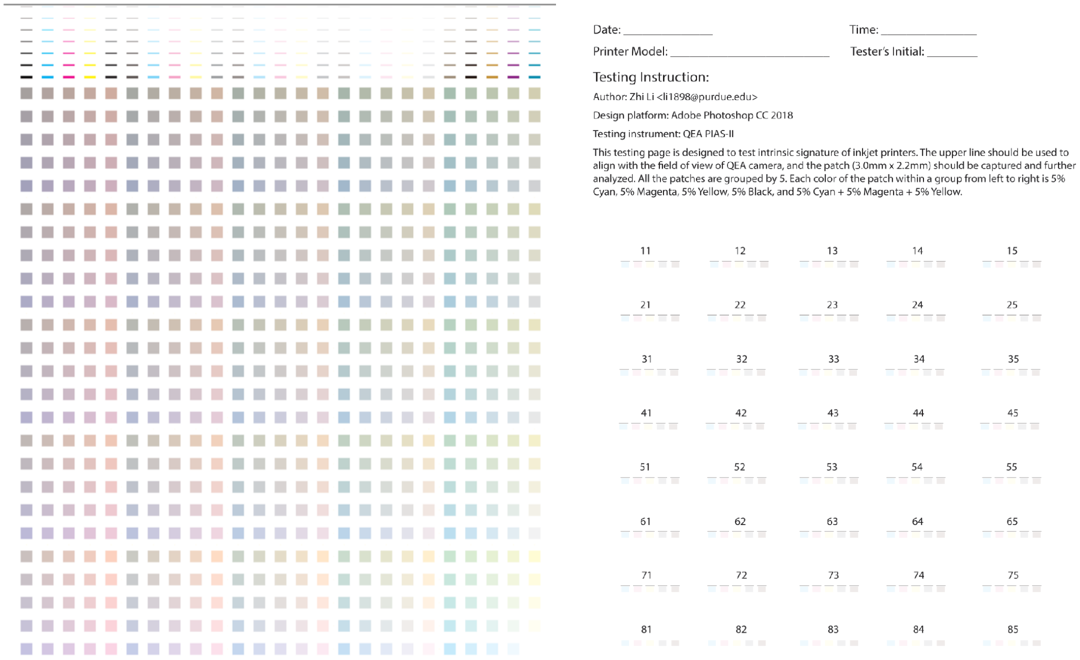
Test Page Design

For our paper, we designed a test page consisting of constant-tone patches with gamma-corrected absorptance levels of 21%, 15%, 10%, 5%, and 0% for each of three colorants (CMY). Our analyses are entirely based on the constant-tone dot patterns. We printed the test page with each of the 15 target printers, and captured images of selected regions using a QEA PIAS-II camera (Resolution 7663.4 dpi with $3.2mm \times 2.4mm$ field of view (1024×768 pixels). We found that when the cyan, magenta, and yellow level is 5%, the dots are dense enough so that their spatial relationship is prominent, yet dot coalescence is reasonably less frequent. For the simplicity of this paper, the 5% cyan, 5% magenta, and 5% yellow patches will be referred as *Triple 5* patches from now on.

Another two vital settings in our experiments are the printer driver settings and the media used. All the pages are printed in the best print quality mode, and the resolution is set to 600 dpi. We choose 600 dpi because this is the standard resolution for SOHO inkjet printers. Also more intrinsic features of the dot spatial relationships can be shown by controlling all the prints to have the same resolution. As for the media, since we target the currency counterfeiting issue, it is desired to study the printer behavior with similar media. Note that some countries have switched to polymer banknotes in an effort to combat counterfeiting and reduce costs. Printing on this special media is beyond our research area. Paper-based currency like the U.S. Dollar is what we try to simulate. According to the Bureau of Engraving and Printing[2], US paper currency is made up of 75% cotton and 25% linen. Therefore, in our experiment, we acquired two types of linen paper from Envelopes.com and Southworth. We also used Boise Multi-Use Copy paper in our experiment, as it represents plain paper that is daily used and most accessible among all the paper types.

Our second half of the experiments is for the classification. This test page consists of 40 repetition of a group of test targets. These test targets are 5% cyan, 5% magenta, 5% yellow, 5% black and *Triple 5*, respectively. These patches are designed to be about $3mm \times 2mm$ to fit in the field of view of the QEA PIAS-II. The bar on top of the patches is designed to help the user align the field of view of the camera with the lattice of printer-addressable points.

[2] For more information on U.S. currency and its paper and ink, see: `https://www.moneyfactory.gov/hmimpaperandink.html`



(a)

(b)

Date: ______________ Time: ______________

Printer Model: ________________________ Tester's Initial: ________

Testing Instruction:

Author: Zhi Li <li1898@purdue.edu>

Design platform: Adobe Photoshop CC 2018

Testing instrument: QEA PIAS-II

This testing page is designed to test intrinsic signature of inkjet printers. The upper line should be used to align with the field of view of QEA camera, and the patch (3.0mm x 2.2mm) should be captured and further analyzed. All the patches are grouped by 5. Each color of the patch within a group from left to right is 5% Cyan, 5% Magenta, 5% Yellow, 5% Black, and 5% Cyan + 5% Magenta + 5% Yellow.

Fig. 3.8.: Designed test pages. (a) is the Phase I test page that is used to select a valid target; (b) is the phase II test page that contains same tone color patches replicated across the page.

3.3.2 Printer Characterization

We propose four different features to characterize the dot patterns from the captures shown in Fig. 3.9 Row 1: Dot Size, Dot Density, Average Distance to the Nearest Dot, and Nearest Dot-Sector Density Function. All our analyses are based on separate color channels, so we will do color separation first.

Colorant Separation

A colorant channel separation is desired to study the printhead characteristics individually. The following processes are used to obtain different channels:

Media pixel elimination. As shown in Fig. 3.9 Row 1, paper pixels are represented as white pixels in the captures, therefore we are able to identify the media/white pixels by examining a modified measure of color saturation S in CIELab color space

$$S = \frac{c^{*2}}{L^{*}}, \tag{3.1}$$

where c^{*2} is the square of the chroma value, and $c^{*2} = a^{*2} + b^{*2}$. We use c^{*2} instead of c^{*} in saturation calculation to balance the significance of the chroma and the lightness when the Lightness value is small. Since we only have yellow, magenta, and cyan ink, if a pixel is inked, it can only be one of, or a combination of these three primary colors (it is unlikely to have all three ink drops overlap together in such a light color, as shown in Fig. 3.9 Row 1). Therefore, as shown in Fig. 3.10, the saturation for inked pixels is higher, and other un-inked pixels have relatively low saturation value. Hence, we can eliminate the media pixels by thresholding S. Our experiment shows that when the threshold is 3, the separation performance is optimal.

Ink separation. To separate different ink colors, we measure the hue angle h_{ab} for every inked pixel.

$$h_{ab} = \arctan(\frac{b^{*}}{a^{*}}). \tag{3.2}$$

The quadrant is determined by the signs of the a^{*} and the b^{*}, and all the angles are calculated in radians. As shown in Fig. 3.11, the majority of the color pixels have the hue angle between [-2, 2], where three peaks can be identified: (from left to right) cyan ink peaks around -2, magenta ink peaks around -0.3, and yellow peaks around 1.5. Therefore we can separate the colored pixels by determining the closest peak. Some results are shown in Fig. 3.9. Although it is rare to see in these light color patches, ink overlapping and coalescence still can happen. We define the color pixels whose hue angles are between two adjacent peaks as overlapping pixels; and they count to both ink maps. Note that we are viewing the unwrapped histogram in Fig. 3.11. The real hue angle is distributed as a closed ring. Therefore, the cyan peak should also be adjacent to the yellow peak.

After the procedure illustrated above, we also would like to filter out noisy pixels. Firstly, insignificant clutters of pixels can be found in the captured images that are too small to be considered as an individual ink drop. Thus, we only consider dot-clusters that have more than 30 pixels. Secondly, pixels near a capture boundary should be eliminated from further study, as the full picture of their surroundings is unclear. Therefore, we only consider dots that are at least 50 pixels away from any of the boundaries. Then, we can perform connected component analysis to determine the different drops. Once we have a list of drops, data can be gathered about the drops such as their size, location, and compactness.

Dot Statistics

The print head and the ink used are critical aspects of the inkjet printer: print heads control the size of the ink drops and other behaviors of the jets, and the chemical nature of the ink determines the appearance of the prints and the ink spread on the paper, hence changing the dot shape. However, for a certain printer model, the print heads deployed and the ink selection are usually fixed, leading to consistent dot statistics as a unique feature shared within the same model. Therefore, we studied the dot statistics of the halftone image microscopic structure. And for each ink, we created the following features: Dot Area and Dot Density.

Dot Area

For Dot Area, we use the number of camera pixels to represent the size of the printed dot. Since we are using the same magnification throughout all the captures, the number of pixels is proportional to actual dot size measured in mm^2. The dot size comparison over all the models is shown in Fig. 3.12. It can be seen that the printers models from same manufacturer, for example, Printers C, D, and E, have similar dot sizes. Another three printers, Printers F, G, and H are also from the same manufacturer; and therefore their yellow dots are much bigger than are those for the other colorants. Note that Printer F is from a different series than Printer G and H; and the size of the dots from Printer F is quite different than the size of the dots from Printers G and H.

Dot Density

Another feature that we find to be distinctive is the dot density, or dot pixel-cluster count per capture. As shown in the Fig. 3.9, different printer models have not only very different overall dot densities (Row 2 in Fig. 3.9), but also different dot densities in the three different channels (Rows 3, 4, and 5 in Fig. 3.9). Hence, we evaluate all the captures and plot the bar chart on dot counts in Fig. 3.13. As seen in the figure, the dot densities of all the printers are very different: Printer B has overall the smallest number of dots across all 8 models and 3 colorants. Printers F, G, and H all have smaller numbers of yellow dots, but the portions of cyan, magenta and yellow dots are quite different; Printers A, C, D, ad E have relatively similar proportions of dots across the three colorants with nearly the same densities for cyan and magenta. But the overall dot densities for Printer E are significantly larger than are those for Printers A, C, and D.

Spatial Distribution

Other than looking into the dot statistics, it is also very important to characterize the spatial dot distribution within each colorant. We propose two metrics to characterize the spatial distribution between the dots.

Average Distance to the Nearest Dot (ADND)

Average Distance to the Nearest Dot (ADND) is a measure of how close together the dots are. For each dot that is fully surrounded by its neighbors, we find out the distance between this dot and its nearest dot. Then, we are able to determine the ADND by average all the distances obtained in the capture image. As shown in Figure 3.14, we can see that Printers F, G, and H have a very distinct dot spatial arrangement compared to the others. The yellow dots are much sparser compared with other colorants. This is also consistent with the data shown in the Figs. 3.12 and 3.13, where we see that the yellow dots are much larger, and fewer in number for Printers F, G, and H, than for the other printers. Meanwhile Printers C, D, and E have similar spatial distributions, as they all from the same manufacturer.

Small ADND values also might also indicate a phenomenon that we call *dot pairing*. Dot pairing occurs when dots of the same colorant frequently occur in close, often nearly horizontal, proximity to each other. Note that unlike a major dot and its satellite dots, which is another distinct phenomenon that can be observed in high resolution captures, dot pairing is more consistent and the two dot sizes are relatively similar. It happens more frequently among the printers with smaller drop size. A set of examples are given in Fig. 3.15. As can be seen in the figure, cyan dots or magenta dots are paired together horizontally. The effect of dot pairing on the visual appearance of the print is not investigated in this study. But this phenomenon is very distinct and could be a good indicator of the printer model.

Nearest Dot-Sector Density Function (ND-SDF)

The ADND metric draws a picture of how individual dots are spaced in the print: the biggest circle, centered at the centroid of the dot, which does not have any other dots inside of it. This concept of the dot placement measurement does not take into account the dot alignment or any other directional information. Therefore, we propose the second measure Nearest Dot-Sector Density Function (ND-SDF). Inspired by the phenomenon of dot pairing, the ND-SDF metric is designed to capture the orientation of the dot and its nearest neighboring dot. The algorithm is described as below:

1. Set up 7 sectors as shown in Fig. 3.16.(a);
2. For color channel c, calculate S_c, the set of all the dots that are not near a bounary of the capture image field of view;
3. For each dot $d \in S_c$, search for the nearest dot in one of the sectors shown in Fig. 3.16.(a). If such a dot d' exists, do:
    (a) Calculate the elevation or depression angle α between d and d';

(b) Register α to the corresponding sector i by adding 1 to count;

4. Normalize each individual sector value by dividing by the sum of all sector values.

Note that to avoid repetition, only dots in Quadrants I and IV are considered during the calculation. An example ND-SDF calculation is given in the Fig. 3.16.(b). The cyan ND-SDF histogram is shown in Fig. 3.17. We choose to show the cyan ND-SDF, as the cyan dots are more likely to pair together than other colorants. We can see that both Printers A and F have very high values in Bins -1, 0, and 1. This means that most of the nearest dot pairs accumulate around the same height. Our findings also correspond to what can be observed in Fig. 3.15.

3.3.3 Printer Identification System (PIS)

In the previous section, some hand-crafted features are engineered and designed to capture the characteristics of the microscopic structures from different prints. Although handcrafted features are intelligible to human examiners, there are some drawbacks:

1. It is rather tricky to model dot behaviors, as the inkjet imaging pipeline and inkjet marking engine technology is rather complex, and also stochastic rather than deterministic.
2. Handcrafted features cannot cover all the features seen and processed by the human viewers.
3. When future new printers join in the study, more features might need to be designed.

Therefore, a machine-learning based Printer Identification System (PIS) that can capture features autonomously is preferred for an anti-counterfeiting effort. Recent years have witnessed the rapid development of Deep Neural Networks (DNNs), and many DNN image object recognition applications have been used in our daily life. Hence, we aim to exploit the DNN's object recognition power to build a printer model classifier. We choose the Residual Network (RNN) [56], as it is one of the most popular and accurate networks in terms of image recognition[3]; and RNN also avoids some problems for network training. Here, we use ResNet50, a 50 layer version of the Network. Compared with ResNet50, other variations (101 layers or 152 layers) have deeper structures, which means that they are more prone to overfitting, while producing marginal accuracy gain.

First, steps should be taken to transform the data collected from the second phase experiment on 6 printers to data that can be used by the network. For each printer, we acquire more than 40 *Triple 5* patches using PIAS-II, and cut the each capture into 224 × 224 smaller images which is the acceptable input size of the network. As illustrated in the previous section, it is desirable to avoid the dot patterns around the image boundaries. Thus, we only collect 12 images from each original capture as shown in Fig. 3.18. Therefore, more than 3,500 image patches are collected as the training and testing dataset. Figure 3.19 presents some examples of the preprocessed input images.

Then, the pre-trained ResNet50 model is used to extract the features from the input images. Since all the weights in the model have been pre-trained using ImageNet [1], the model is well adapted to recognize both high level object details, as well as low level image features. However, original ResNet50 model outputs an label of an object in real life, for instance car, cat, dog, etc.; and this does not correspond to the desired labels in our task. Hence, we extract all the features from the second last layer and train another classifier, which is much simpler, lighter weight, and equally effective. The ResNet50 model produces 2,048 features for a single input image. But these features should not be directly used for the classification task for the following two reasons:

[3] ResNet50 won 1st place in the ILSVRC 2015 classification competition with top 5 error rate of 3.57%. For more information: `http://image-net.org/challenges/LSVRC/2015/index`

1. The dimension of the feature domain directly effects the computational complexity of the training process. With more features to take into account, more time is needed to train the classifier.

2. Using the full size feature sets will lead to overfitting. Considering the limited number of data points we have, a smaller feature set can help improve the generalization performance.

Thus, we use Principle Component Analysis (PCA) to reduce the feature space to 48 dimensions. We only use the first 48 features for training. Note that before conducting PCA, we shuffle and split the data into training and testing sets, and PCA model is fitted to the training set. Then, the same model is used to transform the testing set.

The final step is classification. We use Support Vector Machine (SVM) as the classifier model, and Radial Basis Function (RBF) as the feature space kernel. The confusion matrix of the testing results is shown in Fig. 3.20. The results here are promising: most of the prediction on the testing set is correct, especially for Printer G, H and E all the predictions are correct. Theoretically overfitting is avoided by using SVM classifier and PCA dimension reduction.

3.3.4 Conclusion

In recent years, SOHO inkjet printers play an important role in modern office productivity, but they are also being used as one of the major counterfeiting tools around the world. In this paper, we investigate the intrinsic signatures of currently popular SOHO inkjet printers, And we develop an identification system to predict the deployed printer by examining a printed document. We acquired 8 printer models from 4 major SOHO printer manufacturers. By examining the dot patterns at a microscopic level, we are able to design four intrinsic inkjet printer features that can capture both dot features and spacing features: Dot Size, Dot Density, Average Distance to Nearest Dot and Nearest Dot-Sector Density Function. We find that the printers made by the same manufacturers share more similarity than ones from other manufacturers. Our features are also able to capture phenomena like dot pairing. Finally, we use a Deep Neural Network to extract high dimensional intrinsic features from the collected prints, and fit a Principle Component Analysis model to reduce the feature set to 48 features. A SVM-based classifier is trained on the reduced feature set, and our testing results show the overall prediction accuracy is higher than 95%, and for some printer models the accuracy approaches 100%.

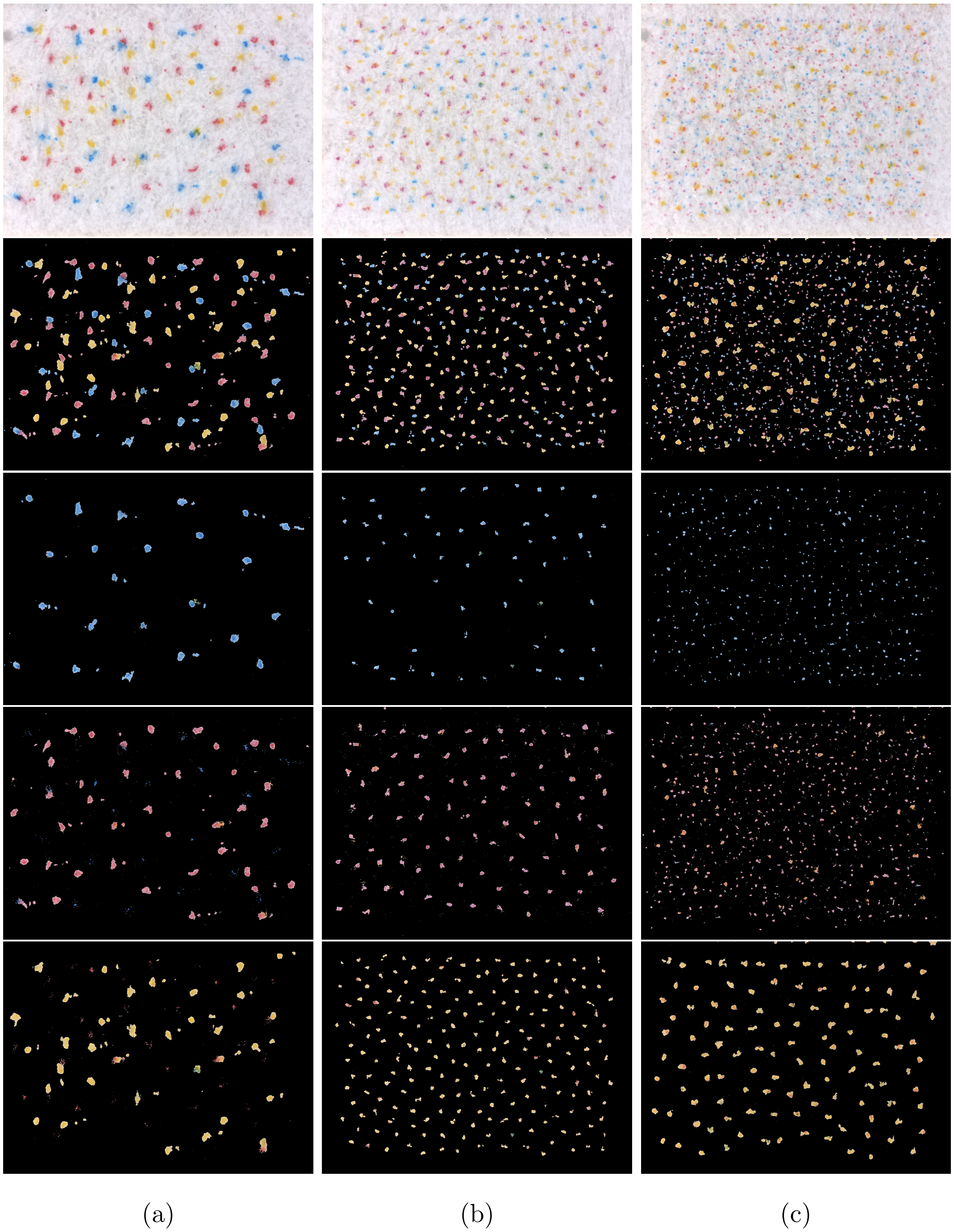


Fig. 3.9.: Sample separation results. The first row shows the original captures. The second row shows the images after eliminating the white media pixels. And the remaining three rows show the cyan, magenta and yellow channels, respectively. Column (a) shows the results for Printer B, Column (b) shows separation results for printer D, Column (c) shows the results for Printer H.

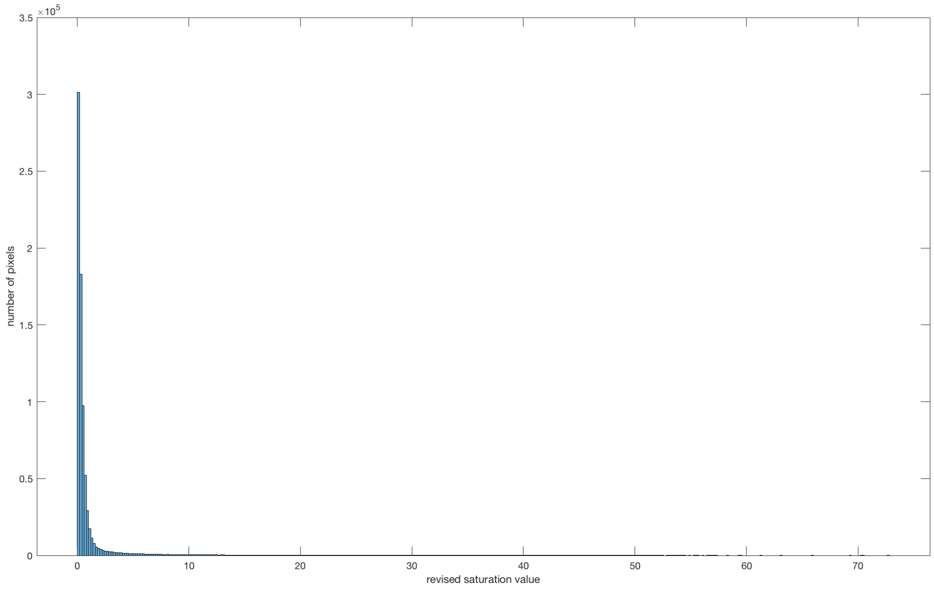


Fig. 3.10.: Distribution of pixels in a typical captured images according to saturation. The saturation has a peak around 3 and a long tail as the saturation value increases. Note that the observation applies across all the printers.

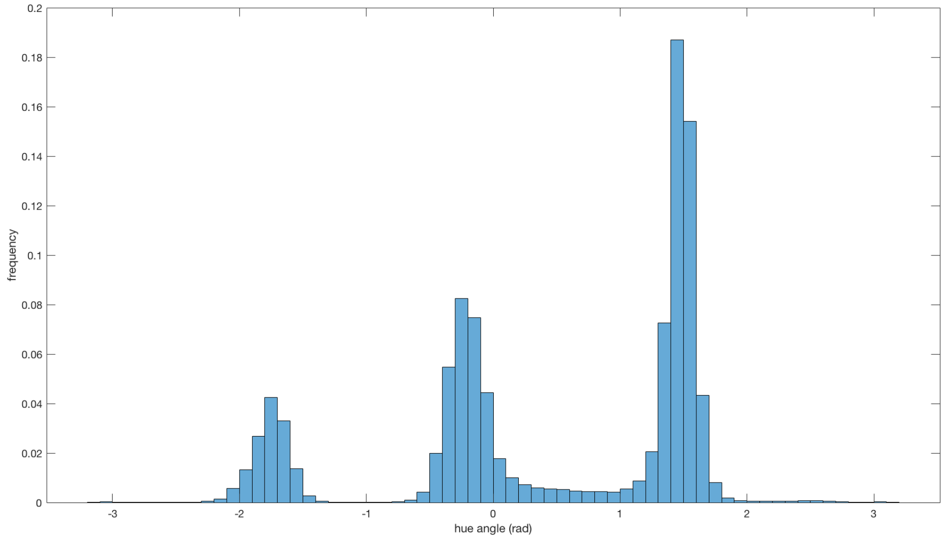


Fig. 3.11.: Distribution of the pixels in a typical captured images according to the hue. The histogram has three peaks around -1.7, -0.3, and 1.5 (rad). Note that the observation applies across all the printers.

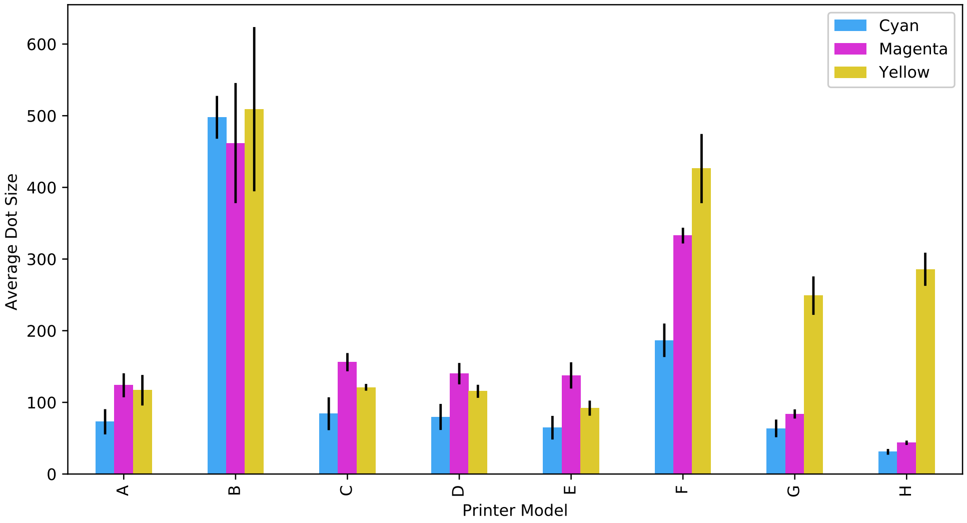


Fig. 3.12.: Average cyan, magenta, and yellow dot sizes comparison over all printer models.

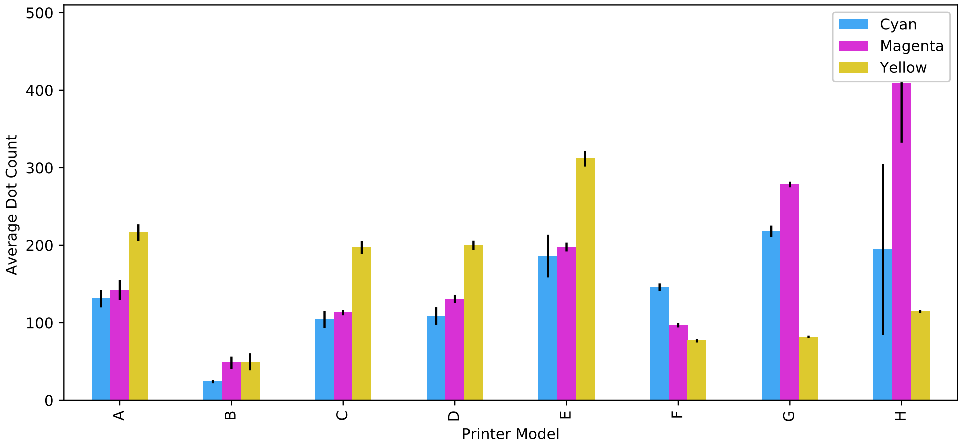


Fig. 3.13.: Average cyan, magenta, and yellow dot counts (number of dot pixel-clusters contained within the capture images) comparison over all printer models. The error bars show +/- 1 standard deviation.

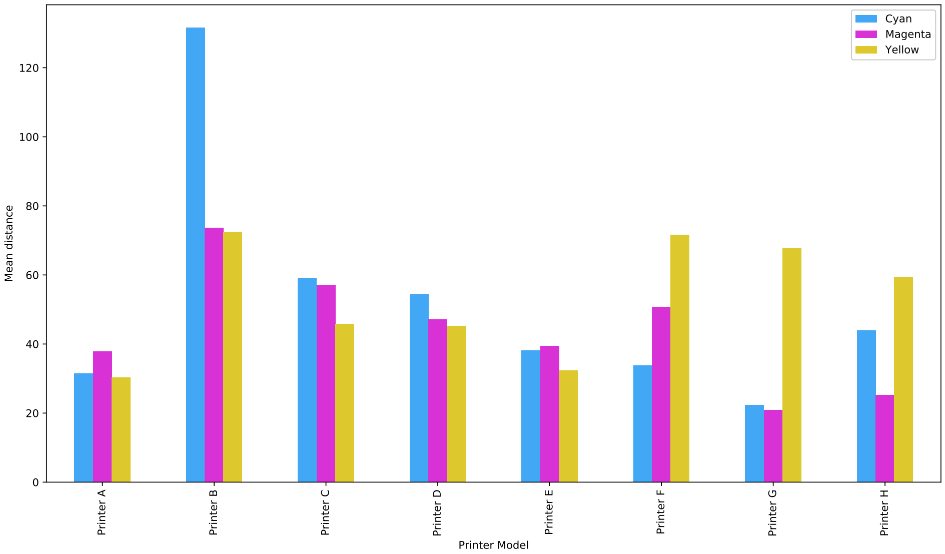
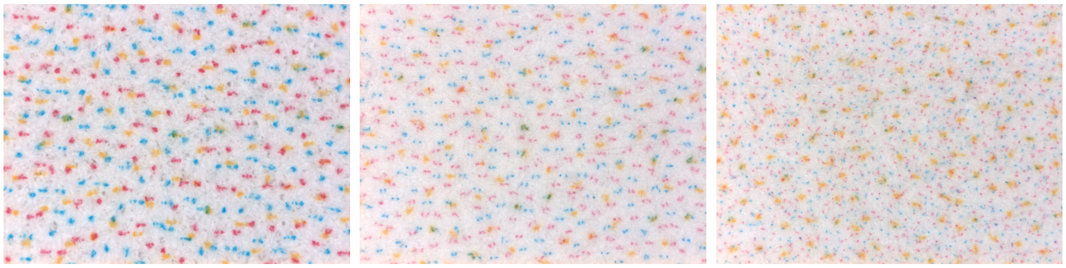
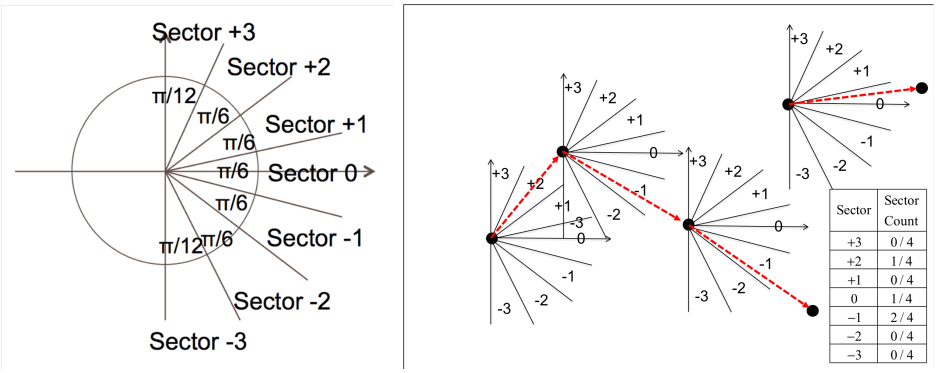


Fig. 3.14.: Comparison of Average Distance to the Nearest Dot (ADND) for the cyan, magenta, and yellow dot separations over all the printer models.



(a) Printer F (b) Printer G (c) Printer H

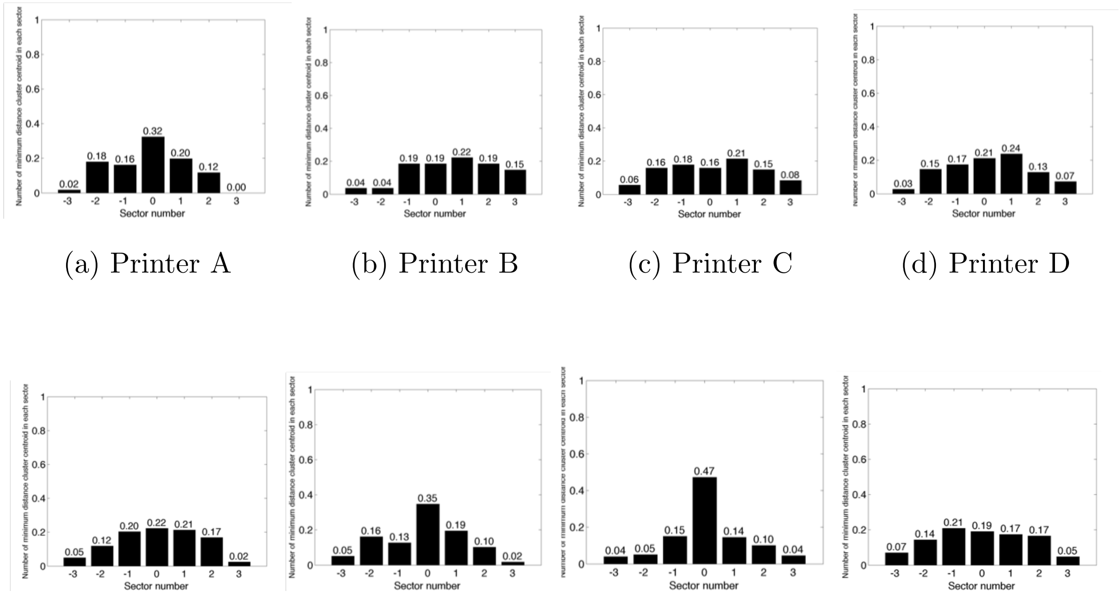
Fig. 3.15.: Examples of dot pairing. Printer F has the most frequent dot pairing phenomena, Printer G has less, and Printer H has the least.



Sector	Sector Count
+3	0 / 4
+2	1 / 4
+1	0 / 4
0	1 / 4
−1	2 / 4
−2	0 / 4
−3	0 / 4

(a) Sector Design (b) ND-SDF Example

Fig. 3.16.: Nearest Dot-Sector Density Function (ND-SDF) concept



(a) Printer A

(b) Printer B

(c) Printer C

(d) Printer D

(e) Printer E

(f) Printer F

(g) Printer G

(h) Printer H

Fig. 3.17.: ND-SDF of cyan dots across all 8 printer models.

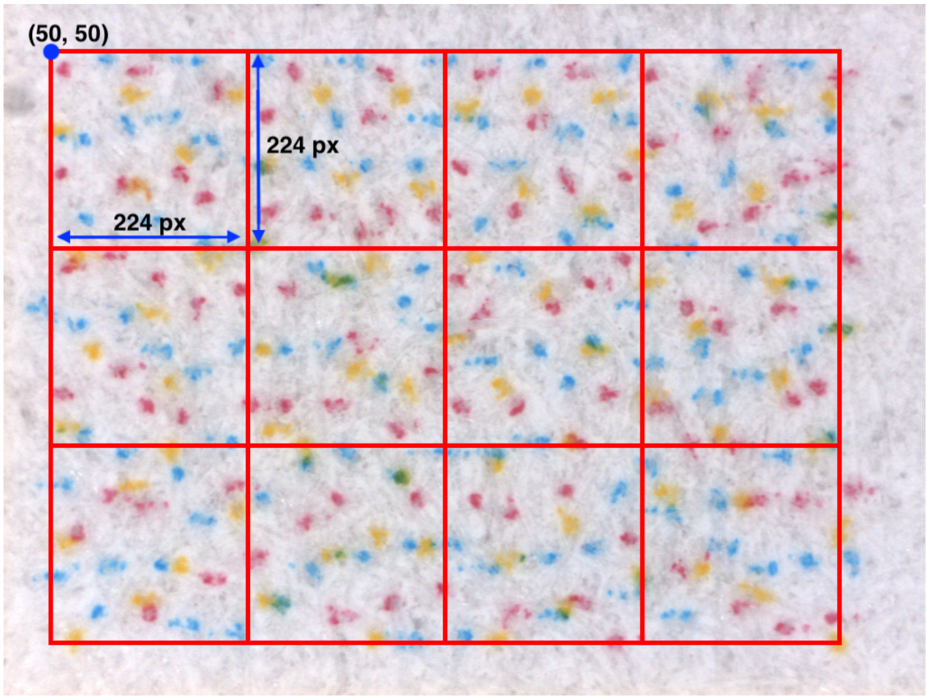
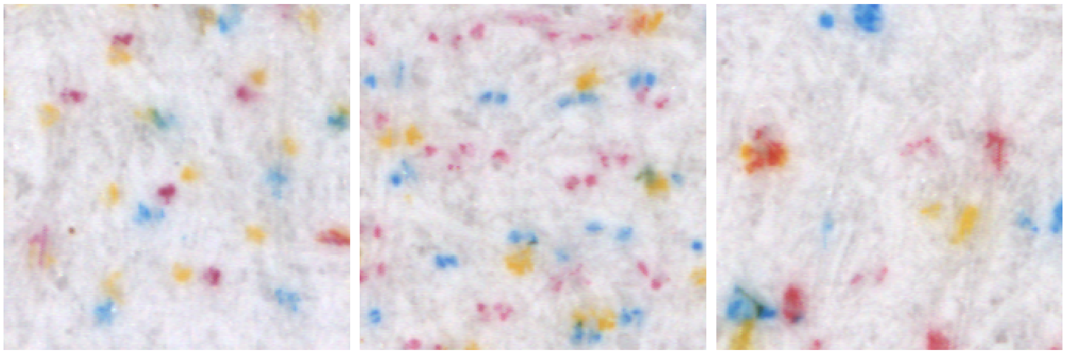


Fig. 3.18.: Example of retrieving 224 × 224 images from original capture.



(a)

(b)

(c)

Fig. 3.19.: Sample ResNet50 input images. All the images are cropped to size 224 × 224. Image (a) is from Printer D, image (b) is from Printer G, and image (c) is from Printer B.

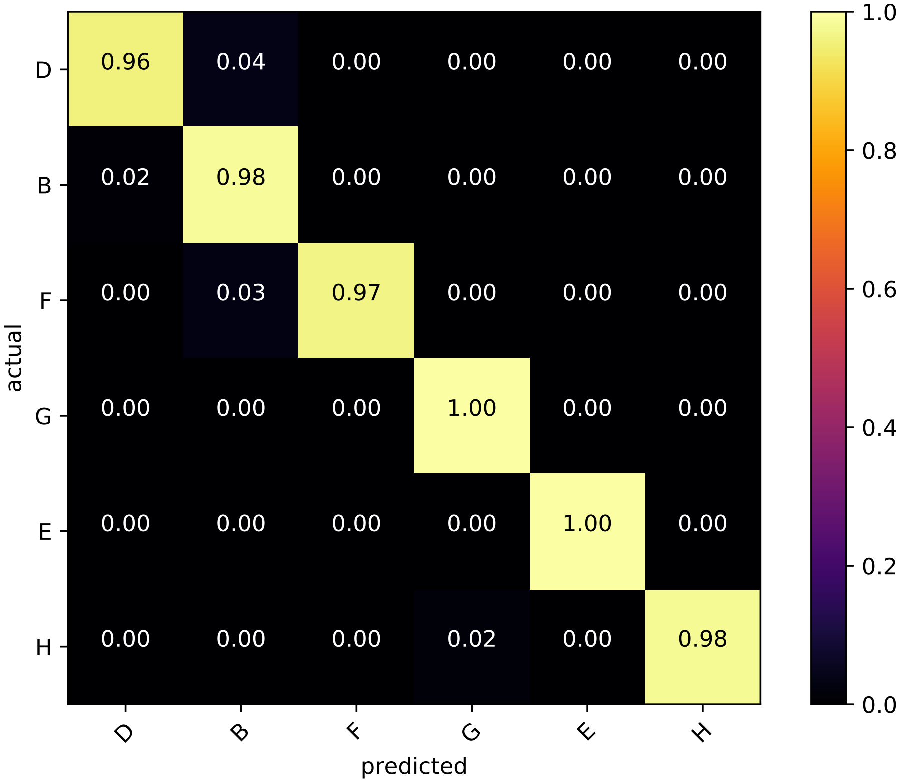


Fig. 3.20.: The confusion matrix of the classification results.

4. CONCLUSION

Machine learning algorithms, especially neural networks, have been widely used in image understanding problems, thanks to its supreme learning (it can learn to solve complex tasks), versatility (same architecture can be applied to different problem sets), and acceleration capability (special hardwares are developed for fast runtime inference). However, due to the unique nature of the problem, finding suitable data or learning from limited data is a constant challenge. In our work, we present several different approaches to develop machine learning based image understanding applications for fashion and inkjet image analysis.

In Chapter 2, we studied and investigated the fashion garment color naming problem. We introduced Autonomous Garment Color Extraction System (AGCES) that utilizes both unsupervised learning and supervised semantic segmentation neural network to extract garment component. Our segmentation model is capable of processing fashion images with human included. We also proposed the Reversed Color Naming Experiment. Our research shows that our trained DeepLab model achieve over 80% validation pixel accuracy and almost 80% Frequency Weighted Intersection over Union accuracy. Finally, our final AGCES integration is able to produce close color prediction for different fashion items and body parts in the image. Overall, our AGCES has been proven to be promising and effective for the color garment extraction task.

In Chapter 3, we aim to apply machine learning algorithms to analyze inkjet print quality and printer intrinsic signature for anti-counterfeiting purposes. In lieu of extracting semantic informations from real life images, we apply neural network architecture to learn non-localized texture information from test targets in printing laboratories. Our research shows that using Residual Neural Network that is trained on mega datasets like ImageNet can perceive image quality and texture features. Thus robust pipelines for noise analysis and printer identification are built without intense training. Our pipelines can achieve high accuracy to perform inkjet image analysis. In addition, a novel model examination methodology is proposed to further validate the effectiveness of the neural network backbone that is trained on completely irrelevant data. Overall, the data-driven computer vision approach presents great value and potential to improve future inkjet imaging technology, even when the data source is limited.

In conclusion, the major contribution of this work can be summarized as following:

- Our fashion image analysis is focused on understanding the semantics of the online fashion images.
  - Developed a novel clustering-based AGCES pipeline for online fashion garment color extraction.
  - Adapted and repurposed public available datasets with pixel-level segmentation map. Trained a semantic segmentation neural network for identifying and segmenting clothing using human presence.
  - Designed and conducted Reversed Color Naming Experiment to collect association between fashion color names and color appearance on marketplace images.
  - Trained random-forest-based color naming algorithm to utilize the color naming data collected from Reversed Color Naming Experiment. We achieved high accuracy.
- Our image/print quality and forensics analysis utilizes the power of pre-trained neural network.
  - Developed a computer vision pipeline for inkjet print quality and forensics analysis using neural network as a backbone without intense training.
  - Modeled noise level and print quality with the print quality pipeline, and our model is able to capture the print quality and characters of high frequency noise on scanned printed test images.

- Our SVM-based Printer Identification System is able to predict the source machine by examining the dot patterns with high accuracy.
- Conducted an ablation study to study the neural network backbone's behavior towards different frequency components of test images.

REFERENCES

REFERENCES

[1] R. Martinez, D. Smith, and H. Trevino, “ImageNet: a global distributed database for color image storage, and retrieval in medical imaging systems,” pp. 710–719, June 1992.

[2] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft COCO: common objects in context,” 2014.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[4] J. Deng, “ILSVRC 2016 object localisation: introduction, results,” *2nd ImageNet and COCO Visual Recognition Challenges Joint Workshop*, 2016. [Online]. Available: http://image-net.org/challenges/talks/2016/ILSVRC2016_10_09_clsloc.pdf

[5] M. H. Kiapour, X. Han, S. Lazebnik, A. C. Berg, and T. L. Berg, “Where to buy it: Matching street clothing photos in online shops,” in *Proceedings of 2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 3343–3351.

[6] G. Csurka, D. Larlus, F. Perronnin, and F. Meylan, “What is a good evaluation measure for semantic segmentation?” in *Proceedings of British Machine Vision Conference*, vol. 27, 2013.

[7] R. Mohan, “Deep Deconvolutional Networks for Scene Parsing,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Nov. 2014.

[8] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. G. Rodríguez, “A review on deep learning techniques applied to semantic segmentation,” *CoRR*, vol. abs/1704.06857, 2017. [Online]. Available: http://arxiv.org/abs/1704.06857

[9] A. Ess, T. Mueller, H. Grabner, and L. V. Gool, “Segmentation-based urban traffic scene understanding,” in *Proceedings of British Machine Vision Conference 2009 (BMVC 2009)*, 2009.

[10] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *Proceedings of 2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 3354–3361.

[11] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[12] Z. Yi, A. Criminisi, J. Shotton, and A. Blake, "Discriminative, semantic segmentation of brain tissue in mr images," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2009*, G.-Z. Yang, D. Hawkes, D. Rueckert, A. Noble, and C. Taylor, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 558–565.

[13] L. Ladick, C. Russell, P. Kohli, and P. H. S. Torr, "Associative hierarchical crfs for object class image segmentation," in *Proceedings of IEEE 12th International Conference on Computer Vision*, Sept 2009, pp. 739–746.

[14] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *International Journal of Computer Vision*, vol. 81, no. 1, pp. 2–23, Jan 2009. [Online]. Available: https://doi.org/10.1007/s11263-007-0109-1

[15] F. Schroff, A. Criminisi, and A. Zisserman, "Object class segmentation using random forests," in *Proceedings of British Machine Vision Conference*, 2008, pp. 54.1–54.10, doi:10.5244/C.22.54.

[16] J. Verbeek and W. Triggs, "Scene Segmentation with CRFs Learned from Partially Labeled Images," in *Proceedings of NIPS 2007 - Advances in Neural Information Processing Systems*, J. C. Platt, D. Koller, Y. Singer, and S. Roweis, Eds., vol. 20. Vancouver, Canada: MIT Press, Dec. 2007, pp. 1553–1560. [Online]. Available: https://hal.inria.fr/inria-00321051

[17] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, "ImageNet large scale visual recognition challenge."

[19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, vol. abs/1409.4842, 2014. [Online]. Available: http://arxiv.org/abs/1409.4842

[20] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," vol. abs/1411.4038, 2014. [Online]. Available: http://arxiv.org/abs/1411.4038

[21] S. Yan, Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, "Unconstrained fashion landmark detection via hierarchical recurrent transformer networks," in *Proceedings of the 2017 ACM on Multimedia Conference*, ser. the 2017 ACM. ACM Press, pp. 172–180.

[22] L. Bossard, M. Dantone, C. Leistner, C. Wengert, T. Quack, and L. Van Gool, “Apparel classification with style,” in *Proceedings of the 11th Asian Conference on Computer Vision - Volume Part IV*, ser. ACCV’12. Berlin, Heidelberg: Springer-Verlag, 2013, pp. 321–335. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-37447-0_25

[23] H. Chen, A. Gallagher, and B. Girod, “Describing clothing by semantic attributes,” in *Proceedings of 2012 IEEE European Conference on Computer Vision*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 609–623.

[24] Q. Chen, J. Huang, R. Feris, L. M. Brown, J. Dong, and S. Yan, “Deep domain adaptation for describing people based on fine-grained clothing attributes,” in *Proceedings of 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015, pp. 5315–5324.

[25] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang, “Deepfashion: Powering robust clothes recognition and retrieval with rich annotations,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[26] Z. Liu, S. Yan, P. Luo, X. Wang, and X. Tang, “Fashion landmark detection in the wild,” in *European Conference on Computer Vision (ECCV)*, 2016.

[27] M. H. Kiapour, K. Yamaguchi, A. C. Berg, and T. L. Berg, “Hipster wars: Discovering elements of fashion styles,” in *Proceedings of 2014 IEEE European Conference on Computer Vision*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 472–488.

[28] E. Simo-serra, S. Fidler, F. Moreno-noguer, and R. Urtasun, “Neuroaesthetics in fashion: Modeling the perception of fashionability,” in *In Proceedings of 2014 IEEE Computer Vision and Pattern Recognition*, 2014.

[29] K. Yamaguchi, T. L. Berg, and L. E. Ortiz, “Chic or social: Visual popularity analysis in online fashion networks,” in *Proceedings of the 22nd ACM International Conference on Multimedia*, ser. MM ’14. New York, NY, USA: ACM, 2014, pp. 773–776. [Online]. Available: http://doi.acm.org/10.1145/2647868.2654958

[30] S. Liu, Z. Song, G. Liu, C. Xu, H. Lu, and S. Yan, “Street-to-shop: Cross-scenario clothing retrieval via parts alignment and auxiliary set,” in *Proceedings of 2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 3330–3337.

[31] J. Fu, J. Wang, Z. Li, M. Xu, and H. Lu, “Efficient clothing retrieval with semantic-preserving visual phrases,” in *Proceedings of the 11th Asian Conference on Computer Vision (ACCV) - Volume Part II*, ser. ACCV’12. Berlin, Heidelberg: Springer-Verlag, 2013, pp. 420–431. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-37444-9_33

[32] Y. Kalantidis, L. Kennedy, and L.-J. Li, “Getting the look: Clothing recognition and segmentation for automatic product suggestions in everyday photos,” in *Proceedings of International Conference on Multimedia Retrieval (ICMR) (ICMR 2013).* Dallas, TX: ACM, April 2013.

[33] K. Yamaguchi, “Parsing clothing in fashion photographs,” in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ser. CVPR ’12. Washington, DC, USA: IEEE Computer Society, 2012, pp. 3570–3577. [Online]. Available: http://dl.acm.org/citation.cfm?id=2354409.2355126

[34] K. Yamaguchi, M. H. Kiapour, and T. L. Berg, “Paper doll parsing: Retrieving similar styles to parse clothing items,” in *Proceedings of 2013 IEEE International Conference on Computer Vision*, Dec 2013, pp. 3519–3526.

[35] H. Xiao, K. Rasul, and R. Vollgraf. (2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms.

[36] A. Mojsilovic, “A computational model for color naming and describing color composition of images,” *IEEE Transactions on Image Processing*, vol. 14, no. 5, pp. 690–699, May 2005.

[37] B. Berlin and P. Kay, *Basic Color Terms: their Universality and Evolution.* Berkeley and Los Angeles: University of California Press, 1969.

[38] P. Kay and C. K. McDaniel, “The linguistic significance of the meanings of basic color terms,” *Language*, vol. 54, no. 3, pp. 610–646, 1978. [Online]. Available: http://www.jstor.org/stable/412789

[39] E. R. Heider, “Universals in color naming and memory.” *Journal of experimental psychology*, vol. 93, no. 1, p. 10, 1972.

[40] Y. Y. Makoto Miyahara, “Mathematical transform of (r, g, b) color data to munsell (h, v, c) color data,” pp. 1001 – 1001 – 8, 1988. [Online]. Available: https://doi.org/10.1117/12.969009

[41] S. Tominaga, “A colour-naming method for computer color vision,” in *Proceedings of the 1985 IEEE International Conference on Cybernetics and Society*, vol. 573, 1985, p. 577.

[42] J. M. G. Lammens, “A computational model of color perception and color naming,” Ph.D. dissertation, Buffalo, NY, USA, 1995, uMI Order No. GAX95-09126.

[43] T. Belpaeme, “Simulating the formation of color categories,” in *Proceedings of the 17th International Joint Conference on Artificial Intelligence - Volume 1*, ser. IJCAI’01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 393–398. [Online]. Available: http://dl.acm.org/citation.cfm?id=1642090.1642144

[44] L. Pressman, “Fashion color trend report London fashion week autumn/winter 2018,” Feb 18AD. [Online]. Available: https://www.pantone.com/fashion-color-trend-report-london-autumn-winter-2018

[45] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk, “SLIC superpixels compared to state-of-the-art superpixel methods,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 8. 2274 – 2282, 2012, a previous version of this article was published as a EPFL Technical Report in 2010: http://infoscience.epfl.ch/record/149300. Supplementary material can be found at: http://ivrg.epfl.ch/research/superpixels.

[46] J. Wang and J. Allebach, “Automatic assessment of online fashion shopping photo aesthetic quality,” in *Proceedings of the 22nd IEEE International Conference on Image Processing (ICIP)*, Sept. 2015, pp. 2915–2919.

[47] Y. P. F. Perazzi, P. Krahenbuhl and A. Hornung, “Saliency filters: Contrast based filtering for salient region detection,” in *Proceedings of 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, ser. 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2012, pp. 733–740.

[48] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference and prediction*, 2nd ed. Springer, 2009. [Online]. Available: http://www-stat.stanford.edu/~tibs/ElemStatLearn/

[49] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 603–619, 2002.

[50] K. Skjrven. Street photographer’s toolbox: Baroque and sinister diagonals. [Online]. Available: https://streetphotographerstoolbox.wordpress.com/2013/01/06/baroque-and-sinister-diagonals/

[51] C. Knight. Fstoppers: The ultimate guide to composition - part one: Just say ’no’keh. [Online]. Available: https://fstoppers.com/architecture/ultimate-guide-composition-part-one-just-say-nokeh-31359

[52] J. Zhao, J. Li, Y. Cheng, L. Zhou, T. Sim, S. Yan, and J. Feng, “Understanding humans in crowded scenes: Deep nested adversarial learning and a new benchmark for multi-human parsing,” *arXiv preprint arXiv:1804.03287*, 2018.

[53] J. Li, J. Zhao, Y. Wei, C. Lang, Y. Li, T. Sim, S. Yan, and J. Feng, “Multi-human parsing in the wild,” *arXiv preprint arXiv:1705.07206*, 2017.

[54] C. M. Bishop, *Neural Networks for Pattern Recognition.* New York, NY, USA: Oxford University Press, Inc., 1995.

[55] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” in *Proceedings of European Conference of Computer Vision (ECCV) 2018*, 2018.

[56] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2016, pp. 770–778.

[57] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollr, “Focal loss for dense object detection,” in *Proceedings of 2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 2999–3007.

[58] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (VOC) challenge,” *Int. J. Comput. Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010. [Online]. Available: http://dx.doi.org/10.1007/s11263-009-0275-4

[59] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[60] W. Wang and U. Neumann, "Depth-aware CNN for RGB-D segmentation," in *The European Conference on Computer Vision (ECCV)*, September 2018.

[61] J. Clark, "The Ishihara test for color blindness." *American Journal of Physiological Optics*, 1924.

[62] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001. [Online]. Available: https://doi.org/10.1023/A:1010933404324

[63] S. Ronaghan, "The mathematics of decision trees, random forest and feature importance in Scikit-learn and Spark." Medium, May 2018.

[64] M. Buhrmester, T. Kwang, and S. D. Gosling, "Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data?" *Perspectives on Psychological Science*, vol. 6, no. 1, pp. 3–5, 2011, pMID: 26162106. [Online]. Available: https://doi.org/10.1177/1745691610393980

[65] K. Thung and P. Raveendran, "A survey of image quality measures," in *2009 International Conference for Technical Postgraduates (TECHPOS)*, Dec 2009, pp. 1–4.

[66] "Recommendation 500-10: Methodology for the subjective assessment of the quality of television pictures," ITU-R Rec. BT.500, 2000.

[67] R. Zeman, W. C. Kress, D. R. Rasmussen, E. K. Zeise, G. Chiu, K. Donohue, and D. Hertel, "Update on incits w1.1 standard for perceptual evaluation of micro-uniformity," in *Proceedings of SPIE - The International Society for Optical Engineering*, 12 2003.

[68] E. N. Dalal, D. R. Rasmussen, F. Nakaya, P. A. Crean, and M. Sato, "Evaluating the overall image quality of hardcopy output," in *Proceedings of Image: Processing, Quality, Capture, Systems (PICS) Conference*, Portland, OR, USA, 1998.

[69] L. Kang, P. Ye, Y. Li, and D. Doermann, "Convolutional neural networks for no-reference image quality assessment," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[70] B. Mishra and R. Rasmussen, "Microuniformity: An image quality metric for measuring noise," in *Proceedings of Image: Processing, Quality, Capture, Systems (PICS) Conference*, 2000, pp. 75–78. [Online]. Available: http://www.imaging.org/IST/store/epub.cfm?abstrid=1643

[71] K. P. F.R.S., "Liii. on lines and planes of closest fit to systems of points in space," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 2, no. 11, pp. 559–572, 1901. [Online]. Available: https://doi.org/10.1080/14786440109462720

[72] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: ACM, 2016, pp. 1135–1144. [Online]. Available: http://doi.acm.org/10.1145/2939672.2939778

[73] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," in *Deep Learning Workshop, International Conference on Machine Learning (ICML)*, 2015.

[74] G. J. McLachlan, "Discriminant analysis and statistical pattern recognition," 1992.

[75] P.-J. Chiang, N. Khanna, A. K. Mikkilineni, M. V. O. Segovia, J. P. Allebach, G. T. C. Chiu, and E. J. Delp, *Printer and Scanner Forensics: Models and Methods*, H. T. Sencar, S. Velastin, N. Nikolaidis, and S. Lian, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[76] B. Hobbs and M. T. Kebir, "Non-destructive testing techniques for the forensic engineering investigation of reinforced concrete buildings," *Forensic Science International*, vol. 167, no. 2, pp. 167 – 172, 2007, selected Articles of the 4th European Academy of Forensic Science Conference (EAFS2006) June 13-16, 2006 Helsinki, Finland.

[77] A. Rippert, "Overview on chemical analysis methods for soho inkjet inks," *SOHO InkJet Counterfeit Analysis Center Symposium*, 2018.

[78] S. Georgescu, "Inkjet printers and their footprints on the paper/media spur marks analysis," *SOHO InkJet Counterfeit Analysis Center Symposium*, 2018.

[79] E. J. D. Aravind K. Mikkilineni, Nitin Khanna, "Forensic printer detection using intrinsic signatures," *Proceedings of SPIE, Media Watermarking, Security, and Forensics III*, vol. 7880, pp. 7880 – 7880 – 11, 2011. [Online]. Available: https://doi.org/10.1117/12.876742

[80] J. Aronoff and S. Simske, "Effect of scanner resolution and character selection on source printer identification," *Journal of Imaging Science and Technology*, vol. 55, no. 5, Sept. 2011. [Online]. Available: http://dx.doi.org/10.2352/J.ImagingSci.Technol.2011.55.5.050602

[81] Y. Wu, X. Kong, X. You, and Y. Guo, "Printer forensics based on page document's geometric distortion," *Proceedings of the 2009 16th IEEE International Conference on Image Processing (ICIP 2009)*, pp. 2909 – 12, 2009. [Online]. Available: http://dx.doi.org/10.1109/ICIP.2009.5413420

[82] H. Wu, X. Kong, and S. Shang, "A printer forensics method using halftone dot arrangement model," Piscataway, NJ, USA, 2015, pp. 861 – 5. [Online]. Available: http://dx.doi.org/10.1109/ChinaSIP.2015.7230527

[83] S. Fahd, M. Iqbal, M. Arif, and M. Javed, "Integrated model: Statistical features, memory analysis for scanner and printer forensics," *2016 4th International Symposium on Digital Forensics and Security (ISDFS)*, pp. 74 – 7, 2016. [Online]. Available: http://dx.doi.org/10.1109/ISDFS.2016.7473521

[84] P. Moon, C. E. Kim, D. Kim, J. Moon, and I. Yun, "Ink-jet printing process modeling using neural networks," *Proceedings of the IEEE/CPMT International Electronics Manufacturing Technology (IEMT) Symposium*, 2008. [Online]. Available: http://dx.doi.org/10.1109/IEMT.2008.5507800

[85] A. Ferreira, L. Bondi, L. Baroffio, P. Bestagini, J. Huang, J. A. Dos Santos, S. Tubaro, and A. Rocha, "Data-driven feature characterization techniques for laser printer attribution," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 8, pp. 1860 – 1873, 2017. [Online]. Available: http://dx.doi.org/10.1109/TIFS.2017.2692722

VITA

VITA

Zhi Li is a fifth year doctoral student and teaching/research assistant in the School of Electrical and Computer Engineering at Purdue University, West Lafayette. His primary research focuses on machine learning applications on image understanding problems, namely fashion image analysis, with limited data. He has also been involved multiple research projects such as fashion textural and imagery analysis. Zhi worked for Xerox in summer 2018 as research engineering intern and investigated using neural network to analyze and synthesize inkjet images. And he joined Midea as computer vision engineer intern in spring 2019 and worked on validating and improving generative models. Beyond academics, Zhi was a member of Purdue University Choir from 2014 to 2019, and Eta Kappa Nu (HKN) Beta Chapter since 2016. He served as the HKN volunteer director in 2018 Spring.