GARMENT TEXTURE CLASSIFICATION BY ANALYZING LOCAL TEXTURE DESCRIPTORS

MD. SHAFIUZZAMAN
BSSE 0322

A Thesis
Submitted to the Bachelor of Science in Software Engineering
Program Office of the Institute of Information Technology,
University of Dhaka in Partial Fulfillment of the
Requirements for the Degree

BACHELOR OF SCIENCE IN SOFTWARE ENGINEERING

Institute of Information Technology
University of Dhaka
DHAKA, BANGLADESH

© MD SHAFIUZZAMAN, 2014

GARMENT TEXTURE CLASSIFICATION BY ANALYZING LOCAL TEXTURE DESCRIPTORS

MD SHAFIUZZAMAN

Approved:

Signature ***Date***

Supervisor: Dr. Mohammad Shoyaib

Co-Supervisor: Emon Kumar Dey

To my *mother and father,*
Who are always there to support me.

Abstract

Now-a-days fashion industries are investing lots of efforts to identify the current fashion trend. As a result, a new research area has been emerged named as 'Fashion Trend Forecasting'. Usually, a fashion forecaster predicts the colors, fabrics and styles that will be presented on the runway and in the stores for the upcoming seasons. It has created an interesting application field for image analysis and retrieval, since hundreds of thousands images of clothes constitute a challenging dataset to be used for automatic segmentation strategies, color analysis, texture analysis, similarity retrieval, clothing classification and so on. This thesis proposes a novel approach for automatic segmentation, color and texture based retrieval and classification of garments in fashion stores databases, exploiting texture and color information. The garment segmentation is automatically initialized by 'Grab-Cut Algorithm' and then it is performed by modeling skin colors with Gaussian Mixture Models. For color similarity retrieval and classification color centiles are calculated from normalized cumulative channel histograms and combined with Local Binary Pattern (LBP) features for texture classification. An extensive survey has been conducted to identify the best suited LBP variants. Finally, the proposed method has been validated under a free-to-use dataset publicly available for scientific purposes.

Acknowledgments

I would like to express my gratitude to my supervisor *Dr. Mohammad Shoyaib*, Associate Professor, IIT, University of Dhaka for his useful comments, remarks and engagement through the learning process of this thesis. His aspiring guidance, invaluably constructive criticism and friendly advice guide me to the right way. Furthermore, I would like to thank *Emon Kumar Dey*, Lecturer, IIT, University of Dhaka for introducing me to the topic as well as for his support on the way. His truthful and illuminating views on a number of issues related to this thesis supports me throughout the entire process.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Internet shopping has grown incredibly in last few years. To meet with the demand of the customers fashion industries are in search of some solutions that can help them to forecast the upcoming fashion. Identifying the current fashion trend may be one of the better solutions regarding this problem as it can predict the future direction. This task largely depends on the retrieval of colors and styles. Image processing and understanding, in particular, could be beneficial in this context. In fact, it can improve the quality of the manual annotations of the operators, as well as accelerate the process itself.

Successful retrieval of color and style from garment texture is a challenging task, because these textures are not uniform due to variations in orientation, scale, or other visual appearance. Furthermore, shadows and wrinkles are often part of the garment textures and they are also designed with complex patterns and multiple colors. The upcoming chapters provide a strategic plan to surpass this issues. This chapter mainly focus on the motivation, objective and scope of this thesis.

1.1 Issues related to Garment Texture Classification

A correct automatic classification of garment texture has the potential for dramatically improving the user experience as well as the industrial process, but at the same time a strong effectiveness is mandatory. Inconsistent categorization has a direct impact on the perception quality of the system. As mentioned earlier garment textures consist of complex patterns and varieties of colors, we have to consider some special cases dealing with the garment textures. For example, to deal with orientation variations, we have to use rotation invariant feature descriptor. It is really hard to identify the desired patterns as garment textures are full of shadows and wrinkles. So, we have to ensure higher accuracy of the descriptor. So, identifying the best suited descriptor for garments texture classification is one of the major contribution of this thesis. Besides, colors of the textures also need to consider as the classification will be error prone unless color distribution is measured [1].

1.2 Research Questions

As stated in the previous section, the nature of 'Garment Texture' raises the following research question:

- Are the existing feature descriptors suitable for classifying 'Garment Textures' or a new feature descriptors is needed?

To be more specific:

- Are the existing variants of the texture descriptors efficient enough to identify the complex patterns of 'Garment Textures'?
- If the existing descriptors are not efficient enough, then what modifications are necessary to define an efficient framework for "Garment Texture Classification"?

The main objective of this research is to answer the questions mentioned above and thus providing a solution for efficient 'Garment Texture Classification' system.

1.3 Scope of the thesis

This thesis address the problem of automatic segmentation, color retrieval and classification of fashion garments. Depending on the images a background removal is performed using 'Grab-Cut Algorithm' [2]. Skin removal is used to extract the garment portion only. Local Binary Pattern (LBP) [3] and color centiles [4] are used to identify the features. A Random Forest classification [5] on these features is used to classify the design category. To summarize, we combine the image segmentation techniques with a powerful texture and color description technique to create a complete fashion images analysis system. The scope of this thesis can be described as follows:

- Our method proposes an image segmentation framework to describe the non-interesting parts, such as skin and additional garments and creates a segmentation by removing them.
- We use a color descriptor that provides discriminative summary of the color distribution of the region of interest.
- We identify the best suited texture descriptor for garment texture classification.
- We use 'Random Forest' classifier to classify the textures based on color and texture features.
- We evaluate our overall method on a publicly available large dataset.

1.4 Organization of the Thesis

In Chapter 2, some preliminaries of the image segmentations are discussed along with a comparative analysis among the feature descriptors. A basic concept about the classifier used in this thesis also reviewed in this chapter. Although there is a good volume of literature addressing texture classification methods, to the best of our knowledge very few literature specifically addresses garment texture classification. However, those approaches have been discussed in Chapter 3. In Chapter 4, a complete framework is proposed to specifically address "Garment Texture Classification". Chapter 5 evaluates the framework introduced in Chapter 4. Finally, Chapter 6 concludes the thesis with a discussion about the proposed framework and future research directions.

This chapter provides a glimpse of the overall thesis. In following chapters, issues discussed in this chapter will be discussed in detail.

Chapter 2

Background Study

The recent emergence of multimedia databases and digital libraries has created new opportunities for researcher to use traditional image processing techniques to new areas of interest. In this thesis, some traditional image processing techniques are combined together to propose a complete framework for 'Garment Texture Classification'. In this chapter, we will focus on the preliminary studies that were reviewed for the thesis.

2.1 Texture analysis and classification

The image of a garment surface is not uniform but contains variations of intensities which can be identified as certain visual texture pattern. For this reason, analyzing the garment textures may provide some identical information to classify them. Classification refers to as assigning a physical object or incident into one of a set of predefined categories. In texture classification the goal is to assign an unknown sample image to one of a set of known texture classes. For example, Figure 2.1 shows 8 texture classes from the Brodatz album [6]. Effective texture classification in images has been an important topic of interest in the past decades, since it can be widely used in many applications for classification, detection or segmentation of images based on local spatial variations of intensity or color. A successful classification, detection or segmentation requires an efficient description of image texture. So, the main challenge of texture classification is to find the fittest descriptor. There are two reasons behind this challenge: On one hand, large intra-class divergence in appearance, such as illumination, color, rotation and scale, makes it extremely difficult to model the texture images of the same class; On the other hand, the wide range of various texture classes increases the difficulty of distinguishing them.

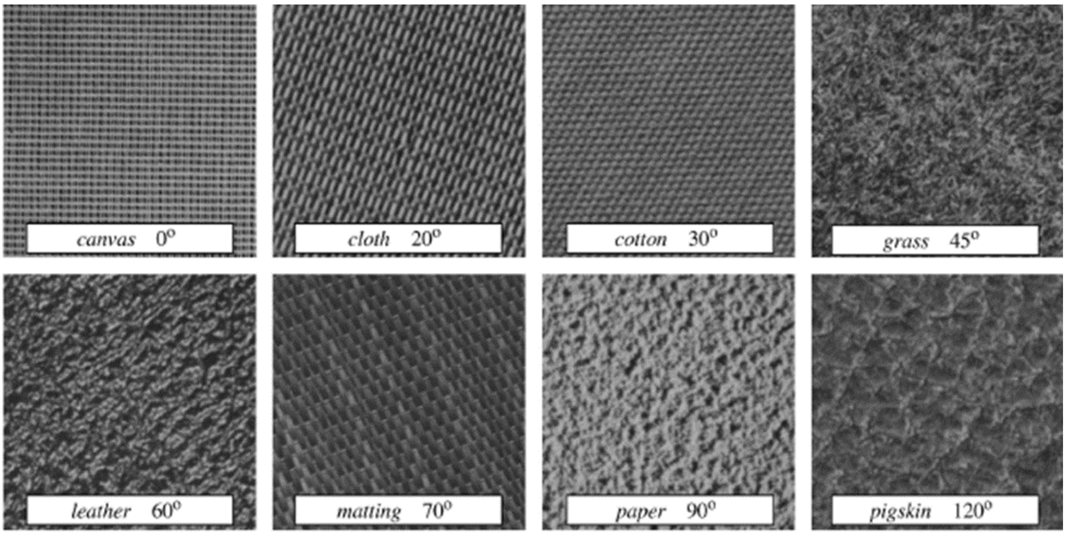

Figure 2.1: Sample Textures from Brodatz album [6]

2.2 Texture Descriptors

Proper feature representation is a crucial step in a texture classification system because a good feature simplifies the classification framework. Texture features can be categorized into two groups - sparse and dense representations [7]. For sparse feature representations, descriptors identify structures such as corners and blobs. Scale-Invariant Feature Transform (SIFT) [8], Speeded Up Robust Feature [9], Local Steering Kernel [10], Principal Curvature-Based Regions [11], Region Self-Similarity features [12], Sparse Color [13] and the sparse parts-based representation [14] are most significant texture descriptors which identify the sparse features. Dense features are extracted at fixed locations densely in a detection window. Various feature descriptors such as Wavelet [15], Haar-like features [16], Histogram of Oriented Gradients (HOG) [17], Extended Histogram of Gradients [18], Feature Context [19], Local Binary Pattern (LBP) [3], Geometric-blur [20] and Local Edge Orientation Histograms [21] are used to identify dense features. As they extract feature using a fixed window, they are also called local feature descriptors. These local descriptors are gaining popularity as they describe objects richly compared to sparse feature descriptors.

Among all the descriptors discussed above, LBP is the most popular texture classification feature. There are several reasons behind this. Firstly, LBP focus on relative intensities instead of the exact intensities. Thus, LBP is less sensitive to illumination variations. Secondly, it considers patch-wise location information instead of exact location information. Thus, LBP is robust to alignment error. Lastly, LBP features can be extracted efficiently for real-time image analysis. Analyzing those points, we have decided to use LBP as our feature descriptor for the ‘Garment Texture Classification’ problem.

The objective of LBP is to describe the surroundings of a pixel. It was originally proposed by Ojala *et al.* [3] in 1996. The basic LBP operator takes a 3-by-3 surroundings of a pixel and generates a binary 0 if the neighbor of the center pixel has smaller intensity than the center pixel otherwise it codes a binary 1. For each given pixel, a binary number is obtained by concatenating all these binary values in a clockwise direction, which starts from the one of its top-left neighbor. The corresponding decimal value of the generated binary number is then used for labeling the given pixel. The derived binary numbers are referred to be the LBPs or LBP codes. Figure 2.2 shows an example of LBP codes.

83	75	95
91	95	141
91	99	100

Sample Image

→

0	0	1
0		1
0	1	1

Binary Code: 00111100

→

	60	

Decimal Code: 60

Figure 2.2: LBP Codes

Formally, given a pixel at (x_c, y_c), the resulting LBP can be expressed in decimal form as follows:

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(i_p - i_c) 2^P \quad (1)$$

where i_c and i_P are respectively, gray-level values of the center pixel and P surrounding pixels in the circle neighborhood with a radius R and function $s(x)$ is called as threshold function and defined as:

$$s(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \quad (2)$$

LBP considered 8 surrounding pixels. However, the LBP operator is not bound to describe only the eight closest pixels. Further developments of the operator support more pixels, cover larger areas and use other thresholds. Moreover, some drawbacks of the basic LBP are identified such as its sensitivity to noise and lack of a mechanism to recover the corrupted patterns. Later, many variations of LBP proposed to mitigate these drawbacks. The main objective of this thesis was to identify the LBP variant that is suited for Garments Texture Classification. So a comprehensive study has been done on LBP variants to identify the best one.

Basic LBP considers 3x3 block of an input image. But sometimes a 3x3 block cannot capture the dominant features. To solve this problem the operator is generalized by applying different sizes of neighborhoods [22] which allows any radius and any number of sampling points in the neighborhood. Figure 2.3 shows some examples of this extension.

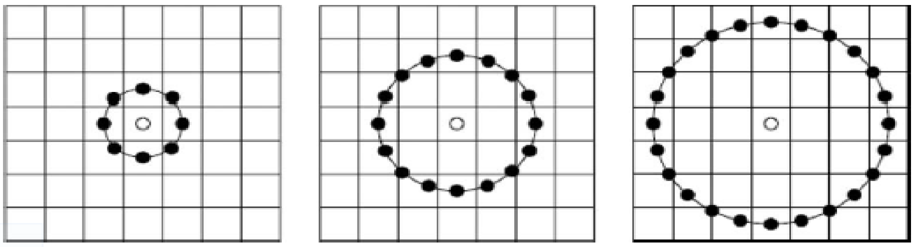

Figure 2.3: Extended LBP. (8, 1), (16, 2) and (24, 3) LBP respectively [22]

Another limitation of basic LBP is that it is not rotation invariant. If the input image rotates then LBP value also changes except for the patterns with only 1's or only 0's. To remove this problem, a rotation invariant LBP is proposed in [23]. They proposed to perform a circular bitwise right shift until the minimum value is achieved. An example of this rotation-invariant LBP is illustrated in Figure 2.4.

00111100 $\longrightarrow$ 00011110 $\longrightarrow$ 00001111

Figure 2.4: Rotation Invariant LBP. 2 bits right shift is made to achieve the rotation invariant LBP

As the minimum value is considered, an image will always provide the same codes irrespective to its any angle of rotation. Rotation invariant LBP also decreases the number of labels used in basic LBP. For example, the number of labels with the neighborhood of 8 pixels is 256 for the basic LBP, but only 36 for Rotation invariant LBP.

Although LBP is simple and robust to illumination variations, performance degrades when there are noises in the input image. To mitigate this problem first approach was proposed by Ojala *et al.* [22] which found some patterns contain more important information than others. These types of pattern are called uniform patterns. Uniform pattern contains at most two bitwise transitions from 1 to 0 or 0 to 1. For instance, LBP calculated in Figure 2.2 (00111100) is a uniform pattern as it has 2 transitions, whereas 11001001 (4 transitions) and 01010011 (6 transitions) are not. The non-uniform patterns are accumulated into a single bin which yields an LBP with less than 2^p labels.

Jin *et al.* [24] pointed that in some circumstances LBP miss the structure of local information. For example, only 256 patterns can be obtained from a LBP (8, 1) operator among all 511(2^9-1) patterns. They proposed an Improved LBP (ILBP) by comparing all the pixels including center pixel with the mean intensity of all pixels.

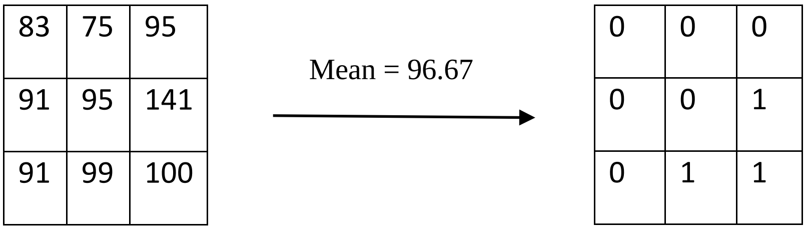

Figure 2.5: Improved LBP.

LBP collects information from all the local regions of an image. But information gathered from all regions may not be equally important for specific application. Without treating all the patterns equally, Ahonen *et al.* [25] set weights for each local region based on the importance of the information it contains.
Li *et al.* [26] proposed the Multi-Block LBP (MB-LBP) that compares the average intensity of the central sub-region with its neighboring sub-regions. Figure. 2.6 shows an example of MB-LBP, where each sub-region consists of six pixels.

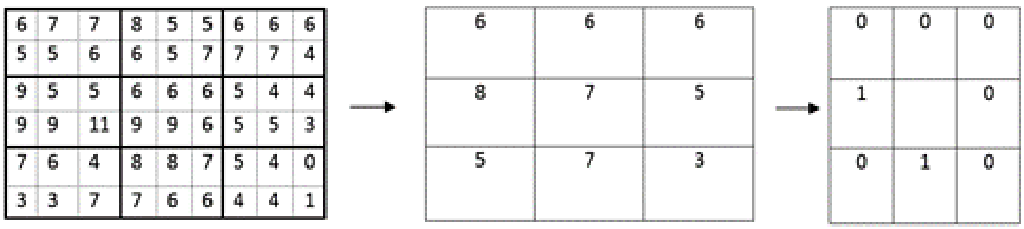

Figure 2.6: Multi-Block LBP.

LBP cannot represent the velocity of local variations. To add this information with LBP Huang *et al.* [27] proposed to use gradient magnitude information alongside basic LBP. As shown in Figure 2.7, the first layer is actually the original LBP code and the following layers encode the binary representation of absolute gray-level value differences (GD). If the first layer is not discriminative enough, the information encoded in additional layers can be utilized to distinguish them.

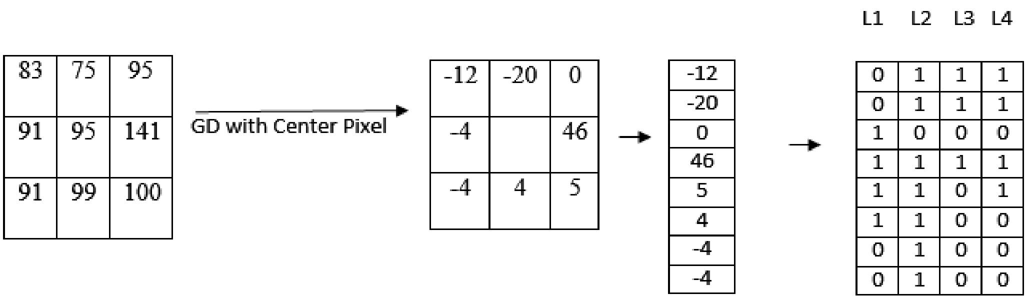

Figure 2.7: GD LBP. L1 signifies the basic LBP code where L2, L3 and L4 is the additional layers that are generated from binary representation of GD

Recently, in 2010 a similar approach called Completed LBP (CLBP) is proposed by Guo *et al.* [28]. Here, the LBP codes are computed in three dimensions – Sign components, magnitude components and center pixel differences. Sign components are actually the basic LBP codes. Unlike the binary bit coding strategy used by [28], CLBP compares GD with the mean GD to calculate magnitude components. For example in Fig. 2.8 the left side 3x3 matrix represents the exact value of GD and the magnitude component is in right side.

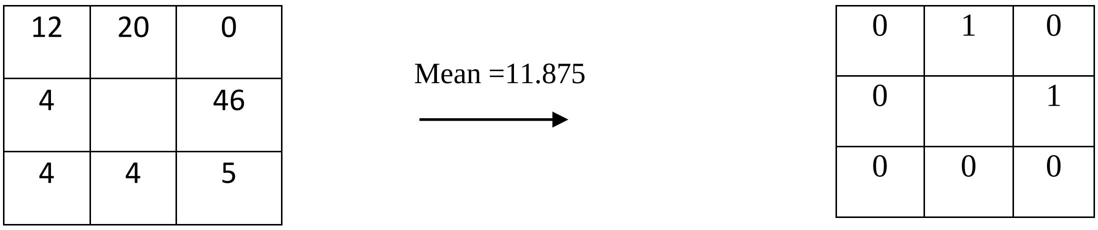

Figure 2.8: Completed LBP. Generated pattern from magnitude component

LBP thresholds exactly at the value of center pixel which makes it sensitive to noise. To address this problem, first initiative was made by Tan *et al.* [29]. They proposed 3-value codes named as Local Ternary Patterns (LTPs). LTP replaced eqn. (2) as follows:

$$s(x) = \begin{cases} 1, & i_n \geq i_c + t \\ 0, & |i_n - i_c| < t \\ -1, & i_n \geq i_c + t \end{cases} \tag{3}$$

Here, t is a user-specified threshold. A coding scheme is used to split each ternary pattern into two parts: the positive one and the negative one, as illustrated in Figure 2.9. One problem of LTP is to find a suitable t, however, Tan *et al.* [29] used t = 5.

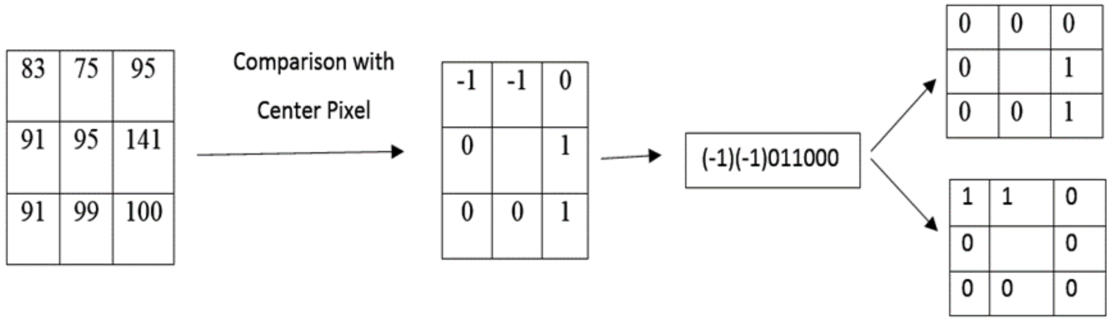

Figure 2.9: LTP

Nanni *et al*. [30] suggest to use a five-value codes and named it as quinary pattern. These five values are encoded using two thresholds (t1, t2). They replaced eqn. (2) as follows:

$$s(x) = \begin{cases} 2, & u \geq x + \tau_2 \\ 1, & x + \tau_1 \leq u < x + \tau_2 \\ 0, & x - \tau_1 \leq u < x + \tau_1 \\ -1, & x - \tau_2 \leq u < x - \tau_1 \\ -2, & otherwise \end{cases} \tag{4}$$

Another significant approach to improve the threshold function of basic LBP is the soft LBP (SLBP) [31] which proposed two fuzzy membership functions instead of eqn. (2).

$$s_{1,d}(x) = \begin{cases} 0, & x < -d \\ 0.5 + 0.5\frac{x}{d}, & -d \leq x \leq d \\ 1, & x > d \end{cases} \tag{5}$$

$$s_{0,d}(x) = 1 - s_{1,d}(x) \tag{6}$$

Parameter d controls the amount of fuzzification. In SLBP, one pixel contributes to more than one bin, but the sum of the contributions to all bins is always 1. As a small change in the input image causes only a small change in output, SLBP provides robustness. However, same as LTP, a proper value of d should be set.

LBP is sensitive to noise and small pixel difference due to noise may affect LBP a lot. Moreover, LBP treat noise-affected image patterns as they are. Hamming LBP [32] proposed to ignore the effect of small pixel difference by distributing them into the uniform patterns. They reclassified the non-uniform patterns into the uniform patterns based on their minimum Hamming distance instead of collecting them into a single bin as [22] does. If several uniform patterns have same hamming distance with a non-uniform pattern, the uniform pattern with minimum Euclidian distance is selected.

Very recently Ren *et al.* [33] proposed a mechanism to recover the corrupted image patterns and named as Noise-Resistant LBP (NRLBP). They encode small pixel difference as an uncertain bit first and then determine the value of uncertain bits based on the values of the other certain bits to form one or more codes. Since uniform patterns occur more likely than non-uniform ones, they assign the values of uncertain bits so as to form possible uniform codes. A non-uniform pattern is generated only if no uniform pattern can be formed. Fig. 14 shows an example of NRLBP. Bins of all the patterns are updated instead of a single bin. For instance, the example used in Figure 2.10 generates 4 patterns. So, ¼ will be added to all of the four bins instead of 1 into a single bin.

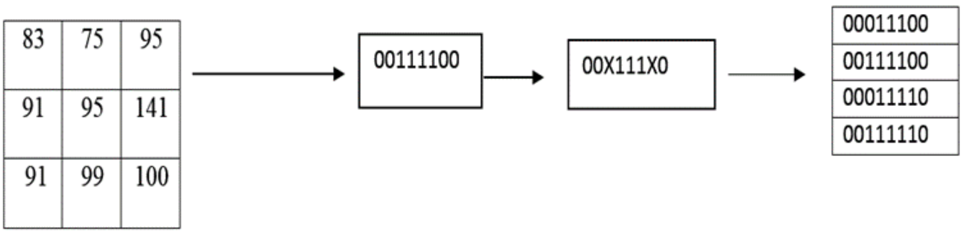

Figure 2.10: Noise-Resistant LBP. X defines uncertain code.

2.3 Background Extraction Method

Background extraction can be done by choosing the appropriate background color for a certain object, or performing further analysis on the object of interest. Background removal can be easily done on photo retouched images, where shadows and minor objects are removed, providing a uniform background of a known color. However, all these methods depend on some assumptions. On the other hand, GrabCut algorithm is a generic background extraction method. So in this thesis, GrabCut algorithm is used to separate our interested garment segment. The algorithm was originally designed by Carsten Rother, Vladimir Kolmogorov & Andrew Blake from Microsoft Research Cambridge, UK in [2]. It uses a Gaussian Mixture Model (GMM) [34] to model the foreground and background. GMM learns and create new pixel distribution by labeling unknown pixels either probable foreground or probable background depending on its relation with the other hard-labelled pixels in terms of color statistics. A graph is built from this pixel distribution where pixels are used as nodes. Additional two nodes are added, Source node and Sink node. Every foreground pixel is connected to the Source node and every background pixel is connected to the Sink node. The weights of the edges are defined by the probability of a pixel being foreground or background. If there is a large difference in pixel color, the edge between them will get a low weight. Then a 'min-cut' algorithm is used to segment the graph. It cuts the graph into two separating source node and sink node with minimum cost function. After the cut, all the pixels connected to the Source node become foreground and those connected to the Sink node become background. The process is continued until the classification converges.

2.4 Classifiers

Besides feature selection, appropriate classifier selection is also an important task of any image classification system. For ‘Garment Texture Classification’ Random Forest classifier is used, because it does not overfit and it is very fast. Alongside, we can run as many trees as we want. A Random Forest consists of a collection of simple decision trees, each capable of producing a classification and "votes" for a specific class. The forest chooses the classification having the most votes over all the trees in the forest. Each tree construction follows a common procedure. If the number of cases in the training set is N, each tree takes N sample cases at random but with replacement from the original data. If there are M input variables, a number $m \ll M$ is specified such that at each node, m variables are selected at random out of the M and the best split is taken. The value of m is held constant during the forest growing. There is no pruning so each tree grows to the largest extent possible.

This chapter reviews all of the preliminaries that were studied for the thesis. Moreover, some points are included that justifies why we use LBP features and Random Forest classifier. In following chapters, we will focus on how we fit these image processing techniques to ‘Garment Texture Classification’ system.

Chapter 3

Literature Review

Though a plenty of research has been done on different types of texture classification, 'Garment Texture Classification' is relatively new area of research. Even though quite a few related works can be found. However, all of these approaches focus on special garment classes and applications. In this chapter, those approaches will be discussed in details.

3.1 Clothing Recognition and Segmentation

Kennedy *et al.* [35] proposed a framework to provide automatic suggestion of clothes from online shopping catalogs. They divided their approach into two stages. First, they detect the classes present in the query image by classification of promising image regions and then, they use image retrieval techniques to retrieve visually similar products belonging to each class. Their main contribution is to propose a simple and effective segment refinement method and similar garment product recognition system. For segmenting they used segmentation method of Felzenszwalb and Huttenlocher [36]. It is a graph-based approach. Low weight of two edges signifies two nodes of same cluster whereas high weight signifies different clusters. Figure 3.1 shows a segment result of this method.

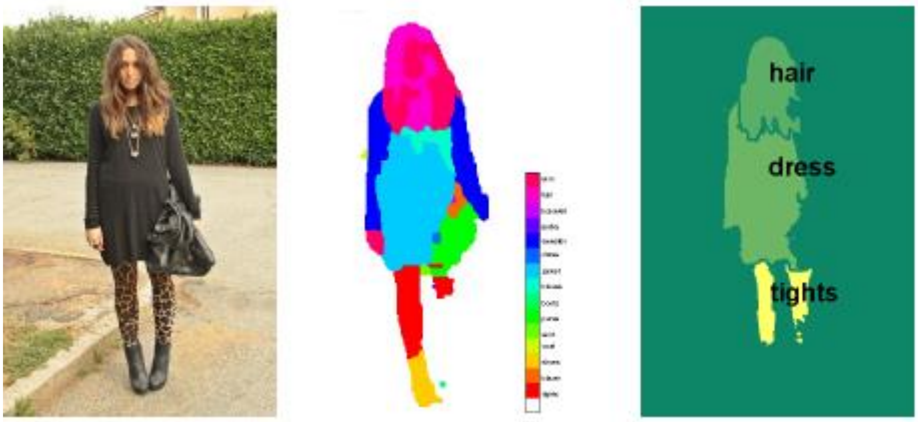

Figure 3.1: Garment Segmentation [35]

To recognize the similar garment, they used human pose estimation in which the whole body is assumed as a graph and different parts of the body assumed as node.

3.2 Clothes Matching

Tian *et al*. [38] proposed an automated cloth matching system for blind and color blind people. They argued their proposed method can handle clothes in uniform color without any texture, as well as clothes with multiple colors and complex textures patterns. Their whole method is divided into two steps – color classification and texture detection.

Their color classification system acquires a normalized color histogram for each image of the clothes in HSI (Hue, Saturation, and Intensity) space. For this reason, each image is first converted from RGB to HSI color space. In particular, for each image of the clothes, the color classifier creates a histogram of the following colors: red, orange, yellow, green, cyan, blue, purple, pink, black, grey and white. Next, HSI space is quantized into a small number of colors. To detect the texture, first they identify whether the color is uniform or not. If the color is uniform, it is detected as no texture in the cloth otherwise it is sure that the cloth contains texture, so further processing is required. Next, Gaussian Smoothing [39] is done to reduce the noise. Then, they apply canny edge detection which can identify the texture pattern easily. Some morphological operation also be conducted to remove the small edges. An example of this method is illustrated in Figure 3.2.

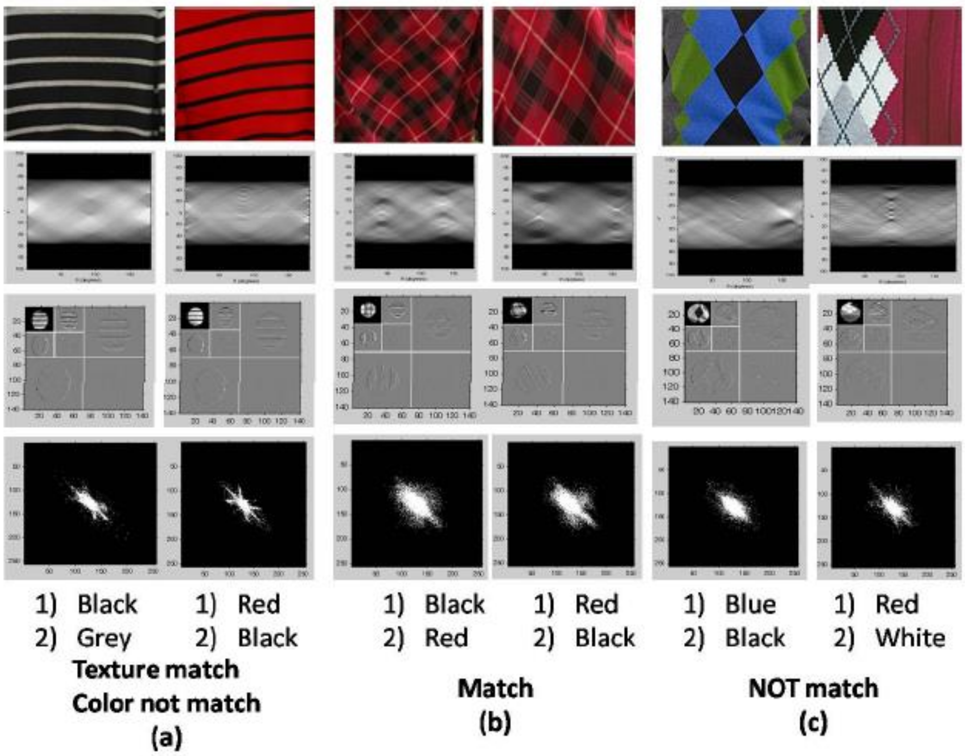

Figure 3.2: Examples of results for clothes matching. (a) The clothes image are texture match, but color doesn't match; (b) the clothes images are match for both texture and color; (c) the clothes images are NOT match for both texture and color.

3.3 Rotation and Illumination invariant Clothes Texture Analysis

Tian *et al*. [40] proposed another complete method for clothes texture analysis by combining Random transform, wavelet features and co-occurrence matrix. The input of this system is a pair of images of two clothes. At first, some preprocessing steps including conversion of color image to gray and histogram equalization are done to remove the effect of illumination changes. Then, Radom transform is used to obtain the dominant orientation information. Next, Haar wavelet transform [15] is employed to extract features on 3 directions (horizontal, vertical and diagonal) and co-occurrence matrix (See Figure 3.3) for each wavelet sub images is calculated. Finally, the matching of clothes patterns is performed based on six statistical features (mean, variance, smoothness, energy, homogeneity, and entropy).

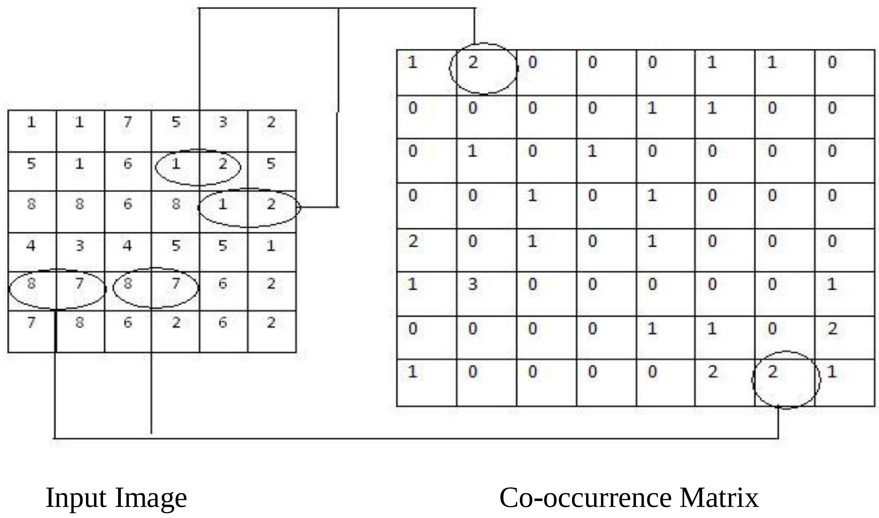

Figure 3.3: Example of Co-occurrence Matrix taken from [40]

3.4 Garment segmentation and color classification

Grana *et al.* [37] proposed a method for automatic segmentation, color based retrieval and classification of garment. For background removal they used Grab-cut algorithm. They extract the region of interest (ROI) by removing the skins from the image. To classify garment using their size, horizontal and vertical projection histogram is used. Color histogram is used to identify color features while HOG descriptor [17] is used to extract texture information. Finally, random forest is used to classify the garment types. Their workflow is very similar to us, though their goal is to identify the garments type. Figure 3.4 provides an example result of this method:

Figure 3.4: Results of garment classification on three categories: skirts, dresses and short pants. In the first column a training image for each class is presented. Second, third and fourth column are correctly classified garments.

3.5 Summary

In this chapter, the existing works regarding 'Garment Texture Classification' are reviewed. Some steps of the first three works are similar to us, though none of them are close to our objectives. The fourth had a very similar workflow to us, though their goal is to classify different types of garment products while our objective is to divide a specific type of garment into some classes according to their design. These literature review help us to identify our scope of work and help us to propose a complete framework which will be discussed in next chapter.

Chapter 4

Garment Texture Classification System Description

Studying existing frameworks, it can be easily identified that none of the approaches directly tackled generic garment classification problem. None of them provide a complete framework for classifying garments using their texture design. Thus a new framework is required to classify a garment product into some classes depending on their designs. In this chapter, a complete method is proposed to classify the garment textures.

4.1 Architecture of the Proposed Method

The main feature of the proposed system is to classify a garment according to its design. The proposed solution classifies the garment products into three classes – Uniform color (No texture), Stripe and Print as shown in figure 4.1.

Uniform Color Stripe Print

Figure 4.1: Garment Classes

The proposed solution is composed of following modules:

i. Background Removal
ii. Segmentation of Garments of Interest
iii. Color Signature Definition and Extraction
iv. Identify Texture Description
v. Garment Classification

Roughly, given an image, background removal is performed in order to obtain a binary mask. Consequently, both skin and additional garments and accessories are removed to obtain a clear picture of the object of interest. Finally, a garment color descriptor and LBP based descriptors are computed to identify the color and texture patterns. Every single module will be detailed in the following sections. The overall schema of the system is provided in Figure 4.2.

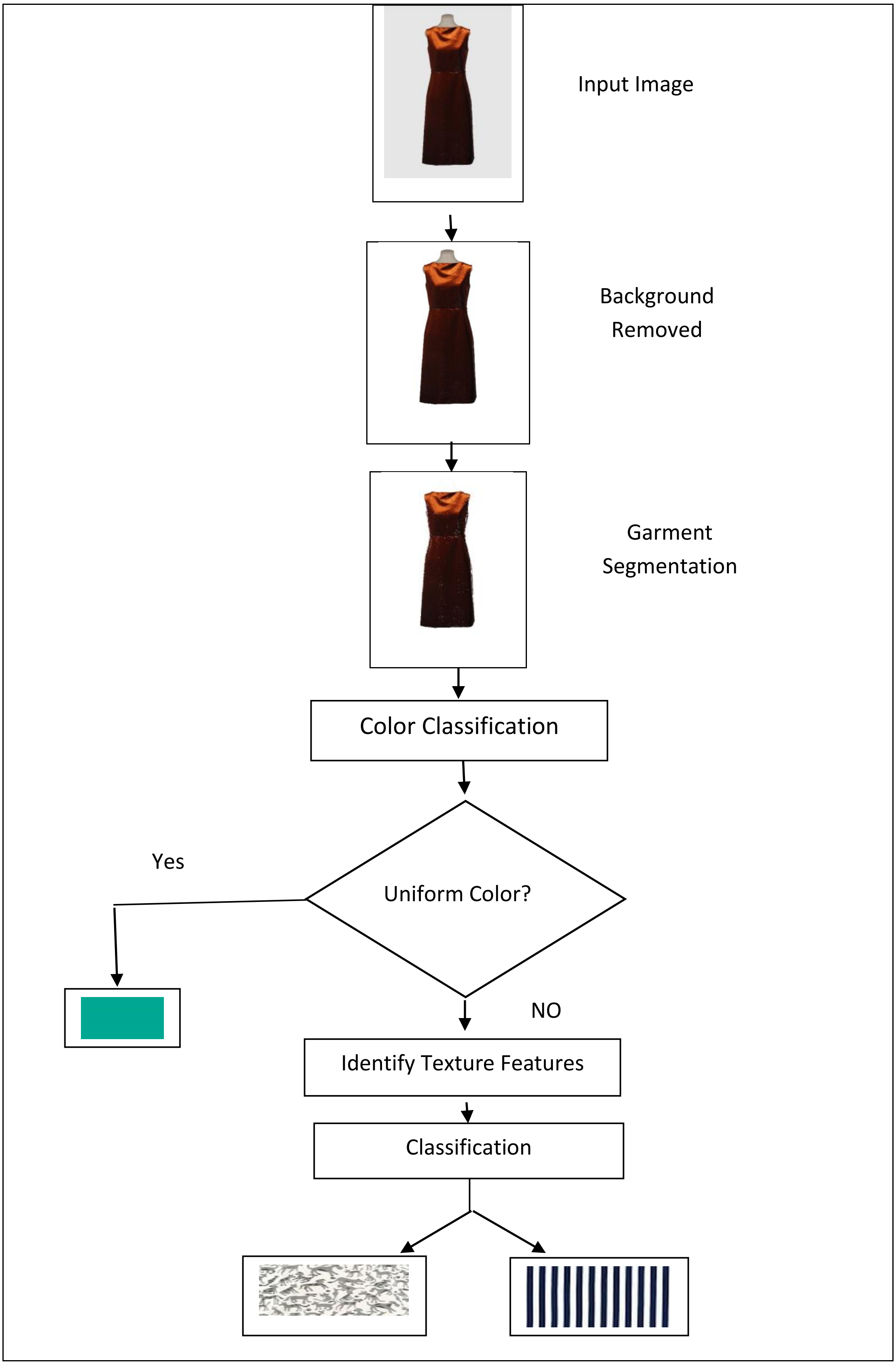

Figure 4.2: Overall Schema of the System

4.2 Workflow in details

As mentioned in the previous section, the proposed system consists of five modules. In this section each module will be discussed in details.

4.2.1 Background Removal

Background removal is the procedure of separating the interested object of an image from the background. It is also called as foreground extraction. We have used the background removal method of [37]. The method starts with a gradient map computation using Sobel operator to highlight the uniform and low-textured areas. Then, an initial background model is generated using the RGB histogram. A background probability map B_p is generated, where the probability of each pixel is represented by the corresponding histogram value. These values are linearly scaled in the range [0 1]. If a pixel x having a color that is never found on the selected background, then $B_p(x) = 0$, on the other hand, when $B_p(x) = 1$, the pixel x belongs to the set of colors which is most likely to be background. After that, the GrabCut algorithm is used (described in section 2.3) to separate the background and the foreground finally. An example of the background extraction procedure is provided in Figure 4.3.

	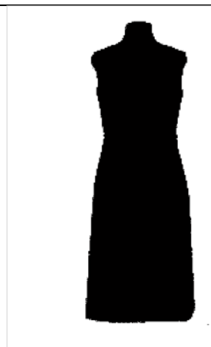	
Input Image	Segmented Foreground	Output Image

Figure 4.3: The background removal procedure.

4.2.2 Segmentation of garments of interest

This step is only needed if the garment products are worn by a model or a mannequin. Skin represents one of the most valuable indicator of people presence. So, skin detection and removal is adopted for this step. The adaptive skin detection approach of [37] is used for this system. Instead of using Gaussian Mixture Models training, [37] used energy minimization approach of Grab-Cut algorithm because it is computationally less expensive. An example of this garment segmentation is provided in Figure 4.4.

	
Input Image	Output Image

Figure 4.4: Skin removal procedure.

4.2.3 Color Signature Definition and Extraction

The main goal of this step is to identify the garments with uniform color. For color extraction, we follow the texture features with color method reported by Kyllonen and Pietikainen [41]. In this method, they used the concept of color centiles. Centiles are color histogram features introduced for wood inspection by Silven and Kauppinen [42]. The centiles can be calculated from normalized cumulative channel histograms $C_k(x)$ by finding the intensity value x that divides the cumulative channel histogram vertically into desired parts, thus it is finding the x when $C_k(x)$ is given. By calculating, color centiles we get a value for each RGB channel. This value ranges [0, 1]. When the color is uniform, all the three values become 1.

4.2.4 Identify Texture Description

Our research question was to identify the suited texture descriptors for garment texture classification. To identify this we have made a comprehensive survey on LBP variants (described in section 2.3). From the survey, we have finalized completed LBP as the texture descriptor of garment classification. The rotation invariant uniform LBP is used in this context. After calculating LBP codes a LBP histogram is generated for each image.

4.2.5 Classification

This is the final module of the system. To classify the garments into three predefined class, Random Forest Classifier (described in section 2.5) is used. In particular Random Forest classifiers have been chosen because they can handle multiclass problems easily providing an inherent feature selection mechanism. The random forest is trained using the LBP histograms and color centiles.

The overall ‘Garment Classification System’ is detailed in this chapter. The core modules of the system is explained one by one. Evaluation of this proposed system will be provided in the next chapter.

Chapter 5

Experimental Results

This chapter verifies the correctness of the proposed system. First part of the chapter focus on the experimental setup and dataset description and the next portion visualizes the efficiency of the system.

5.1 Experimental Setup and Data Description

Total experiment of the thesis was done in 'MATLAB R2012a'. Feature selection and classification works were done separately. The efficiency of the texture descriptor was evaluated under 'Outex' dataset which is a State-of-the-Art dataset for texture classification and can be found in web at www.outex.oulu.fi. We use 13 test suites of Outex database which contain 320 surface textures. For Garments Classification evaluation, a publicly available dataset was used that is available at http://imagelab.ing.unimore.it/fashion_dataset.asp. As this dataset consists of various kinds of garment products, only the shirts and skirts are separated. Then the images were manually categorized into three classes including uniform color, stripe and print. The final experimental dataset contains following images of different classes.

Class	**No. of Images**		
	Skirts	**Shirts**	**Total**
Uniform Color	2441	200	2641
Stripe	173	200	373
Print	1142	200	1342

Table 5.1: Dataset

Each class was divided into five sub-classes. Four sub-classes of each class were used to train the classifier and the fifth one was used to test.

5.2 Result

As there are no complete system in the literature to compare with our system, the efficiency of each module is compared separately.

5.2.1 Garment Segmentation

After running the first two modules of the system segmentation of interested garment region is achieved. In order to quantify the effectiveness of the garment segmentation algorithm, we do not have any ground truth. For this reason, we randomly picked 500 images from the dataset and ran proposed segmentation method. To quantify the efficiency of the garment segmentation strategy, we manually check each of the images and found most of the images were segmented as expected, some were segmented partially and very few were segmented wrongly. Table 5.2 provides the segmentation result:

Segmented Successfully	Partially Segmented	Wrongly Segmented
481	17	2

Table 5.2: Accuracy of the Garment Segmentation Method

So, the accuracy of the segmentation algorithm is reported as 96.20% while wrongly segmented 0.04% and partially segmented 3.4%.

5.2.2 Texture Descriptor

To test the effectiveness of the texture descriptor, we test our texture descriptor under 'Outex' dataset and compared with some State-of-the-Art LBP variants. Table 5.3 provides a comparison among the descriptors.

Texture Descriptor	Accuracy in Outex
LBP	84.82
Mean LBP	79.22
Humming LBP	82.03
LTP	76.06
Fuzzy LBP	87.43
Noise Resistance LBP	92.10
Completed LBP (Used in this thesis)	93.87

Table 5.3: Accuracy of the descriptor

5.2.3 Garment classification

The garment classification algorithm was tested on a selected dataset of 4556 images belonging to 2 categories (Shirts and Skirts). Initially, we generate result for skirts. The result is reported in table 5.4 and 5.5.

Classes	Training Image	Test Image	Correctly Detected	False Detection	Proportion of Correct and False detection
Uniform Color	1941	500	488	12	122 : 3
Print	892	250	192	58	96 : 29
Stripe	138	35	21	13	21 : 13

Table 5.4: Classification Rate for Skirts

Category	Precision	Recall
Uniform Color	0.89	0.98
Print	0.93	0.77
Stripe	0.72	0.60

Table 5.5: Precision and Recall for Skirts

The table 5.6 and 5.7 are generated using 200 images of shirt for each class.

Classes	Training Image	Test Image	Correctly Detected	False Detection	Proportion of Correct and False detection
Uniform Color	160	40	31	9	31:9
Print	160	40	36	4	9:1
Stripe	160	40	37	3	37:3

Table 5.6: Classification Rate for Shirts

Category	Precision	Recall
Uniform Color	0.94	0.85
Print	0.86	0.90
Stripe	0.86	0.93

Table 5.7: Precision and Recall for Shirts

5.3 Discussion

By analyzing the above results, it can be identified that the segmentation tool and feature descriptor we used really perform well enough to meet the expectation. Though, table 5.2 signifies that the accuracy of the classification much depends on the training sets. As the number of training images of stripe class was very low, it had low recall. To identify this issue more precisely, we can observe table 5.2, here all of the classes had more accurate recall where the classifier is trained with equal number of images of each class. Uniform color detection sometimes does not result as expected because some wrinkles presented in the clothes are detected as textures. Without this issue, the classification rate is close enough to accept it as a good classifier.

Chapter 6

Conclusion

In this thesis, a complete method for garment texture classification has been proposed. The proposed method has great potential of being efficient in terms of adaptable to different fashion rules and accurate enough to compete with human operators' performance on the same data. There are some limitations of the method such as it cannot identify wrinkles successfully and not adaptable for all kinds of garments accessories. Our future plan is to reduce the error rate and enhance the method for more garment accessories such as bags and shoes to identify the current fashion trend more precisely.

Bibliography

[1] M Crosier, L D Griffin, "Using Basic Image Features for Texture Classification," International Journal of Computer Vision, July 2010, Volume 88, Issue 3, pp 447-460

[2] C. Rother, V. Kolmogorov, A. Blake, GrabCut: Interactive foreground extraction using iterated graph cuts, ACM Trans. Graph., vol. 23, pp. 309–314, 2004

[3] T. Ojala, M. Pietikainen, D. Harwood, "A comparative study of texture measures with classification based on featured distribution," Pattern Recognition, vol. 29, no.1, pp. 51–59, 1996.

[4] M. Niskanen, O. Silvén, H. Kauppinen, "Color and texture based wood inspection with non-supervised clustering", Scandinavian Conference on Image Analysis , 2001

[5] Ho, T. Kam, "A Data Complexity Analysis of Comparative Advantages of Decision Forest Constructors," Pattern Analysis and Applications, vol. 5, p. 102-112, 2002

[6] P. Brodatz, Textures: A Photographic Album for Artists and Designers.Dover, 1966.

[7] R. Haralick, "Statistical and structural approaches to texture," Proc. IEEE, vol. 67, no. 5, pp. 786–804, 1979.

[8] J. Chen, S. Shan, C. He, G. Zhao, M. Pietikainen, X. Chen, and W. Gao, "WLD: A robust local image descriptor," IEEE Trans. *Pattern Anal.Mach. Intell.*, vol. 32, no. 9, pp. 1705–1720, 2010

[9] X. Hong, Member, G. Zhao, M. Pietikainen, and X. Chen, "Combining LBP Difference and Feature Correlation for Texture Description," IEEE trans. on image proc., 2014

[10] J. Chen et al., "WLD: A robust local image descriptor," IEEE Trans. *Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1705–1720, Sep. 2010.

[11] C. Geng and X. Jiang, "Face recognition based on the multi-scale local image structures," *Pattern Recognition.*, vol. 44, nos. 10–11, pp. 2565–2575, 2011.

[12] H. Bay, A. Ess, T. Tuytelaars, and L. J. V. Gool, "Speeded-up robust features (surf)," Comput. Vis. Image Understand., vol. 110, no. 3, pp. 346–359, 2008.

[13] H. J. Seo and P. Milanfar, "Training-free, generic object detection using locally adaptive regression kernels," IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 9, pp. 1688–1704, Sep. 2010.

[14] H. Deng, W. Zhang, E. Mortensen, T. Dietterich, and L. Shapiro, "Principal curvature-based region detector for object recognition," in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit., Jun. 2007, pp. 1–8.

[15] J. Maver, "Self-similarity and points of interest," IEEE Trans. Pattern Anal. Mach. Intell., vol. 32, no. 7, pp. 1211–1226, Jul. 2010.

[16] J. Stottinger, A. Hanbury, N. Sebe, and T. Gevers, “Sparse color interest points for image retrieval and object categorization,” *IEEE Trans. Image Process.*, vol. 21, no. 5, pp. 2681–2692, May 2012.

[17] C. Papageorgiou and T. Poggio, “A trainable system for object detection,” Int. J. Comput. Vis., vol. 38, no. 1, pp. 15–33, Jun. 2000.

[18] P. Viola, M. J. Jones, and D. Snow, “Detecting pedestrians using patterns of motion and appearance,” Int. J. Comput. Vis., vol. 63, no. 2, pp. 153–161, 2005

[19] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit., Jun. 2005, pp. 886–893

[20] A. Satpathy, X. Jiang, and H.-L. Eng, “Human detection by quadraticclassification on subspace of extended histogram of gradients,” IEEE Trans. Image Process., vol. 23, no. 1, pp. 287–297, Jan. 2014.

[21] X. Wang, X. Bai, W. Liu, and L. Latecki, “Feature context for image classification and object detection,” in Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit., Jun. 2011, pp. 961–968

[22] T. Ojala, M. Pietikainen, and T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” IEEE Trans. Pattern Anal. Mach. Intelligence, vol. 24, no. 7, pp. 971–987, July, 2002

[23] M. Pietikainen, T. Ojala, and Z. Xu, “Rotation invariant texture classification using feature distributions,” Pattern Recognition, vol. 33, pp. 43–52, 2000.

[24] H. Jin, Q. Liu, H. Lu, and X. Tong, “Face detection using improved LBP under Bayesian framework,” in Proc Int. Conf. Image and Graphics (ICIG), 2004, pp. 306–309.

[25] T. Ahonen, A. Hadid, and M. Pietikainen, “Face recognition with local binary patterns,” in Proc. Euro. Conf. Comput. Vis., 2004, pp. 469–481.

[26] T. Maenpaa, J. Viertola, and M. Pietik¨ainen, “Optimising colour andtexture features for real-time visual inspection,” Pattern Anal. Appl.,vol. 6, no. 3, pp. 169–175, 2003.

[27] D. Huang, Y. Wang, and Y. Wang, “A robust method for near infrared face recognition based on extended local binary pattern,” in Proc. Int. Symp. Vis. Comput., 2007, pp. 437–446

[28] Z. Guo, L. Zhang, and D. Zhang, “A completed modeling of local binary pattern operator for texture classification,” IEEE Trans. Image Process., vol. 19, no. 6, pp. 1657–1663, Jun. 2010

[29] X. Tan and B. Triggs, “Enhanced local texture feature sets for face recognition under

difficult lighting conditions," in Proc. Analysis and Modeling of Faces and Gestures (AMFG), 2007, pp. 168-182.

[30] L. Nanni, A. Lumini, and S. Brahnam, "Local binary patterns variants as texture descriptors for medical image analysis," Artificial intelligence in medicine, vol. 49, no. 2, pp. 117–125, 2010

[31] T. Ahonen and M. Pietikainen, "Soft histograms for local binary patterns," in Proc. Fin. Signal Process. Symp. , Oulu, Finland, 2007

[32] H. Yang and Y. Wang, "A LBP-based face recognition method with Hamming distance constraint," in Proc. Int. Conf. Image Graph., Aug., 2007, pp. 645–649.

[33] J. Ren, X. Jiang, and J. Yuan, "Noise-Resistant Local Binary Pattern with an Embedded Error-Correction Mechanism," IEEE trans. on image proc. , 2014

[34] Yu, Guoshen, "Solving Inverse Problems with Piecewise Linear Estimators: From Gaussian Mixture Models to Structured Sparsity". IEEE Transactions on Image Processing 21 (5): 2481–2499, 2012

[35] Y. Kalantidis, L. Kennedy, L. Li, "Getting the Look: Clothing Recognition and Segmentation for Automatic Product Suggestions in Everyday Photos", 3rd ACM conference on International conference on multimedia retrieval, 2013, pp 105-112

[36] P.F. Felzenszwalb and D.P. Huttenlocher. Efficient graph-based image segmentation. IJCV, 59(2):167{181, 2004.

[37] M. Manfredi. C. Grana, S. Calderara, R. Cucchiara, "A complete system for garment segmentation and color classification", Springer Machine Vision and Applications May 2014, Volume 25, Issue 4, pp 955-969

[38] Y. Tian, S. Yuan, "Clothes Matching for Blind and Color Blind People", Springer, Computers Helping People with Special Needs, Lecture Notes in Computer Science, Volume 6180, 2010, pp 324-331

[39] Shapiro, L. G. & Stockman, G. C: "Computer Vision", page 137, 150. Prentice Hall, 2001

[40] S. Yaun, Y. Tian, "Rotation and illumination invariant texture analysis: Matching clothes with complex patterns for blind people", IEEE 3rd International Congress on Image and Signal Processing, 2010, 2643 – 2647

[41] M. Pietikainen , T. Maenpaa , J. Viertola, "Color texture classification with color histograms and local binary patterns", Pattern Recognition, vol. 27, no.1, pp. 81–89, 2002

[42] M. Niskanen, O. Silven, and H. Kauppinen, "Color and texture based wood inspection with non-supervised clustering" , Pattern Recognition, vol. 23, no.1, pp. 62–67, 2000