Fabric Content Detection Using Textural Properties

Catalina Murray
Department of *Electrical and Computer Engineering*
University of Florida
Gainesville, Florida

Abstract— *Fabric properties are a key measurement for any designer or manufacturer in the garment industry. One of the main properties of a textile is its composition. The content of a piece of fabric can usually only be determined through extensive testing which is both costly and time-consuming. In this paper, a method is proposed that uses computer vision techniques to determine fabric composition with images alone. An evaluation was done on 15 different fabric blends in six different fabric groups, the method achieved accuracy between 89-100%.*

I. Introduction

In garment manufacturing and design the properties of each textile have a significant impact on the construction, wearability, durability, and price of each piece of clothing. Often, manufacturers receive hundreds of fabric samples each week. Some of the key pieces of information every manufacturer and designer needs to know about each fabric include its thread count, whether it is a weave or a knit, the color, and its content. To get this information, they must send the fabric to a lab to do testing which is expensive and timely. This process can take anywhere from a few weeks to a few months.

In order to automate this process computer vision methods have been evaluated. While researchers have been successful at determining the weave type of different textiles, little research has been completed on finding a way to determine the composition of each piece of fabric. Some of the proposed methods for finding the fabric's content use complicated testing or lack the ability to evaluate blended fabrics, a fabric that is comprised of multiple materials (e.g., cotton and polyester). In this paper, some popular computer vision and pattern recognition techniques are used to classify fabrics based on their content. The method proposed utilizes Local Binary Patterns and Gray Level Covariance Matrices to extract the textural features of each image. K-nearest neighbors, a non-parametric supervised learning method is then used to classify the images.

II. Description

There are many key features that can distinguish fabric composition, including but not limited to their elasticity, durability, and roughness. However, one of the main problems with classifying fabric content solely based on images is that you can't extract certain features like durability. Another problem researchers have experienced is the large inter-class variance present within each fabric type. For example, faux leather, faux fur, and fleece can all be made from 100% polyester, but all of which have very different visual properties.

To overcome this, in the proposed method each fabric was grouped based on their type (faux leather, outdoor fabric, denim) such that the small textural differences between each fabric's content were easier to extract. The texture was evaluated to be one of the key features to classify each fabric content, both Local Binary Patterns and Gray Level Co-Occurrence matrices were utilized to extract these features. To enhance the image's properties adaptive histogram equalization was used which helped to create greater local contrast in the images. Augmentation was also used to both increase the number of images in each dataset as well as the variance of image type. Once the images were preprocessed the LBP and GCLM were extracted from each image. To do this the image was split into several sub-images and features were extracted from each sub-image. This was shown to increase the accuracy of the classification models. However, since features were taken of each sub-image the number of features for each image increased significantly. In order to decrease the dimensionality and to avoid over-fitting, PCA was performed. Various classifiers were evaluated and tested, and K-nearest Neighbors was shown to have the most success.

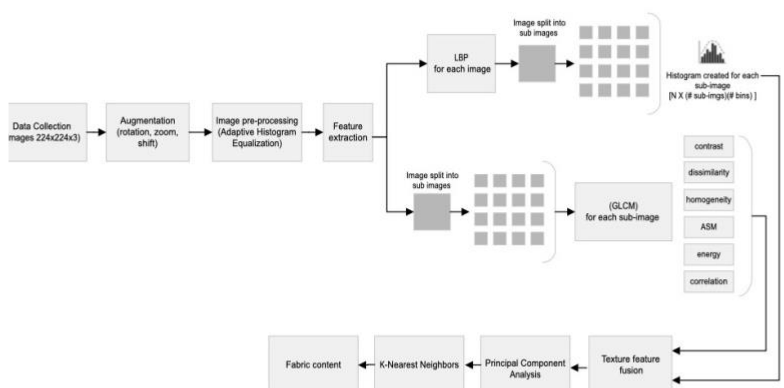

Fig 1. Outline of full proposed method

III. Related Work

The research surrounding fabric classification can largely be divided into three main categories: fabric deformation, weave type, and content extraction. While the category most applicable to this project is content extraction, there is limited research available. The methods used for detecting fabric deformities as well as the weave type were also considered.

A. Content classification

The two main methods that have been researched to determine fabric makeup have utilized spectroscopy [5] and microgeometry structures [6]. The latter is a computer vision method while the former utilizes the electromagnetic spectra which comes from the interaction between electromagnetic radiation and matter. Utilizing NIR spectroscopy and an Extreme learning machine was successful in classifying fabric content with accuracy as high as 100%. However, the data collection process for this method requires careful consideration and special equipment. The focus of this paper, instead, was to eliminate the need for tedious laboratory procedures to determine fabric content.

The second method utilized micro geometry and reflectance to determine fabric content. They used a photometric stereo to take images under different illumination conditions. This helped them to estimate the normal vector and albedo at each point. Utilizing these features as well as fully connected deep convolutional features from the VGG-M model as well as FV pooling they were able to achieve an accuracy of 79.6% on average. However, they only considered fabric compositions that had a content 95% or higher of any one type of material and did not consider blended fabrics.

B. *Fabric Defects*

Many pattern recognition techniques have been utilized in automating the process of finding fabric deformations. The fabric deformations usually come in the form of an oil stain, hole, or broken weft or warp [3]. In these methods the goal usually is to extract the area of the image that is unlike the rest of the image. This is different from the process of discovering fabric content, because we are looking for the underlying textural pattern of the image rather than a hole. Therefore, for these methods, special attention was paid to the ways they were able to extract the larger patterns, or reference patterns, in the images. Some of the main methods utilized in this area of research include histogram-based approaches, color-based segmentation, texture-based approaches, fusion-feature-based operations, and deep-learning-based approaches [7]. Some of the most applicable approaches include the histogram and texture-based approaches. The most common methods used for the histogram-based approaches include gray ratio methods, saliency-based approaches, and fuzzy inductive reasoning.

One histogram method proposed was a multi-window gray ratio that analyzed the change in gray ratio in each sub-image which helped to find the area of the image that had a deformity [8]. Another method that was proposed extracted the saliency maps of an image to create a distinction between the defective regions of the image and defect-free regions [9]. Both methods produced adequate results however as with many histogram-based approaches, they are sensitive to noise and irregular textures. The advantage of these methods is how low-cost they are computationally. Although it is this same simplicity makes them not the best choice for complex data sets.

The texture approaches are largely driven by wavelet-based filters and GLCM features. Some of the limitations of the wavelet methods Some of the major limitations of these methods were that they only looked at finding deformities on one type of fabric, for example jersey knit. One method proposed was to use wavelet analysis [10]. The advantage of this method is that it detects line features efficiently with low complexity. The method is composed of a series of steps including converting the images to grayscale, histogram equalization, and applying a median filter to reduce noise. The wavelet decomposition is then applied at different scales to extract significant features. Thresholding is then applied to find the region of the image with the defect. This method achieved adequate results however the main limitation is that it was only performed on the type of fabric deformity. The generalization of this method is therefore unclear. A more complex method proposed utilized a generalized learning-based approach that used statistical signatures to find the defective regions of the fabric [11]. This method also achieved adequate results, especially with more complex fabric samples. They used the TILDA database which includes a number of images of textiles with deformities. In this dataset there are only 8 different types of textiles. It is unclear whether or not these techniques could be applied to determine other fabric properties.

C. *Weave Type*

The third main category of fabric detection includes detecting the weave type of fabric. The weave type has to do with the way the threads of the fabric are intertwined [14]. Many of these methods were interested in finding the regions of the image where the threads crossed in order to extract a pattern of the thread structure. Some of these methods were useful in this paper because they helped to identify ways to extract the underlying structure of the fabric which can often be unique to each fabric composition. One method that was suggested was to use PCA, GLCM and Gabor Wavelet and a probabilistic neural network classifier. This method achieved an accuracy of 95%. Another method proposed utilized GLCM to find the texture features which were then optimized using PCA [15]. The features were analyzed by a fuzzy c-means clustering algorithm. The main limitation of both of these methods was their difficulties with fluffs and folds in the fabric. Folds are common in fabric and are bound to be present in images however they make it difficult to find the underlying texture. A third method similar to the other two methods mentioned combined the Gabor filter with the Fourier transform [16]. In order to extract the texture features of the image, Fourier transform technology was used to create a spectrum map. A Gabor filter was applied to make a convolution with the origin image and therefore generate sub-images. The parameters were then extracted from the sub-images.

Some researchers have also evaluated the use of deep convolutional networks in weave-type recognition [17]. The proposed method utilized a residual network architecture (ResNet-50). Some of the preprocessing techniques utilized augmentation to increase the size of the dataset. The model proved to be robust to many different fabric variations including color, yarn thickness, rotation and uneven light. However, some of the major limitations is only a few woven types were considered and no non-woven classes were used. Additionally, the data collection process required careful lighting equipment which if this method was applied to real-life applications may cause problems because the equipment may not be available to manufacturers in the field.

IV. Experiments and Evaluation

The experiments and evaluations performed to achieve the final model can be broken up into five different sections: data collection, preprocessing, feature extraction, hyperparameter tuning, and model selection.

A. Data collection

The data utilized are in the form of images. The goal of this project was to detect the presence of different contents in textiles. Initial analysis was done on 500 images of varying types of fabric with contents of 100\% polyester, 100\% cotton, and a blend consisting of polyester, PVC, and polyurethane. However, there was little success with this initial investigation primarily due to the large inter-class variance in each content group.

In the next experiment, the fabrics were grouped into their perspective fabric type; for example, denim, knit, faux leather, etc. From there the images were classified by their content makeup. This initial separation of fabric type admittedly makes data collection a tedious process but improves the performance of the models immensely. Data was collected from two different fabric stores as well as from various clothing items. An iPhone camera was used to collect all the images. To achieve an adequately trained model images could only be taken of textile types that had multiple content blends as well as multiple fabric samples. Images were collected from 6 different categories of fabric: Denim, Faux Leather, Fleece, Knit, linen, and Outdoor fabric. Each category had 2-3 different fabric content make-ups.

The fabric types and their respective content blends are listed in the table below. The content is listed from left to right from the largest percentage to the smallest percentage, respectively. If a percentage is not listed, there were not enough fabric samples of that specific content blend, and instead, images were taken of any fabric that included the same materials at differing percentages. The table also lists the number of images taken which was limited to the number of fabric types available in the stores.

TABLE I. DATA COLLECTION: FABRIC GROUPS

Fabric group	**Data Collection**		
	Fabric Type	***Content***	***Number of images***
1	Denim	100% Cotton	28
	Denim blend	Rayon, other	24
2	Faux Leather 1	100 % Polyester	36
	Faux Leather 2	100% Polyurethane	31
	Faux Leather 3	PVC, Poly, PU	59
3	Fleece	100% Polyester	505
	Fleece blend 1	94% Polyester, 6% Spandex	69
	Fleece blend 2	67% Modal, 28% Polyester, 5%Spandex	38
4	Knit	100% Polyester	68
	Knit blend	Polyester, Rayon, Spandex	75
5	Linen	100% Linen	31
	Linen blend 1	Linen, Rayon	50
	Linen blend 2	Viscose, Linen	64
6	Outdoor Poly	100% Polyester	55
	Outdoor Cotton	100% Cotton	409

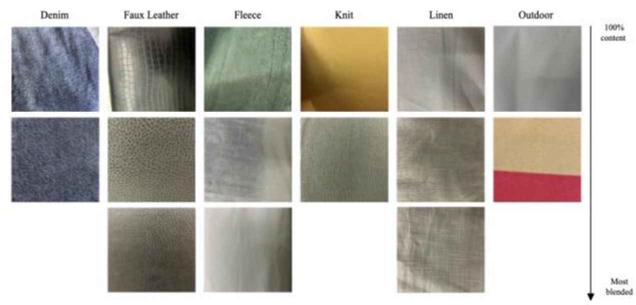

Fig. 2 Example images of each fabric group and content

B. Preprocessing

The preprocessing step in any computer vision model is critical to performance. There were several preprocessing methods compared

1) Image Size and Color

Initially, the images were imported as 300x300 grayscale images to reduce the complexity of the data. They were then binarized utilizing Otsu's thresholding. The binarization process proved to be varying in success since each of the images had extremely different lighting conditions. This was improved by dividing the image into various sub-images. Each sub-image had a different threshold value based on the number of dark values and light values present. However, when utilizing some later methods, including transfer learning with CNN's the expected images were RGB images. In order to make the 300x300x1 images into 300x300x3 images each image was duplicated 3 times which proved to be far less successful than using RGB images.

To compare methods later on RBG images were ultimately used and transformed to grayscale when necessary. Additionally, for similar reasons as listed above images were imported as 224x224 images. Images of any size smaller than 224x224 lost too much detail.

2) Segmentation

Segmentation was evaluated as a pre-processing technique, but it did not improve any of the results largely because the images had a consistent pattern throughout the whole image. Only a few images contained anything besides the fabric.

Another tested method was cropping each image to only contain the part of the image where the texture pattern of the image was strongest. To do this the image was transformed to greyscale and then it was split into 16 quadrants and binarized. For each quadrant, the number of white pixels was compared to the number of black pixels. The difference in the number of black and white pixels was compared over each quadrant. The quadrant with the most similar difference in pixel values to all the other quadrants was kept and any quadrant that had a large difference in white or black pixels was left out. The underlying theory behind this method was that any region of the image that suffered poor lighting, which would either have almost all black pixels or white pixels, thus compromising the true structure of the fabric in any histogram-based approach, would be removed. While the method at first looked promising some of the more complex textures lost a lot of information in this process. It proved to be successful with some of the microscopic images that were tested from the i-bug data set but were ultimately left out of the final dataset. With higher quality or more microscopic images, this method could be more useful.

3) Histogram Equalization

To enhance the texture of each image images were processed using histogram equalization. Histogram equalization and adaptive histogram equalization were compared. The first method spreads out the most frequent intensity values which help to increase the global contrast of the image. Adaptive histogram equalization is like histogram equalization however it computes multiple histograms over varying parts of the image which helps to improve the local contrast. This method was chosen because it helped to enhance the textural features of the image more.

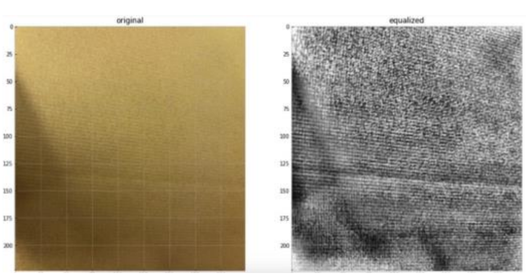

Fig 3. Example of Adaptive histogram Equalization

4) Augmentation

Due to the limited number of samples of each fabric type as well as to increase the generalization of the model each fabric type was augmented. The augmentations produced included rotations, zoom, shift, and flip [18]. Images were chosen randomly from each data set and a total of five images of each augmentation type were added to each data set.

C. Feature Extraction and Selection

There are some key features that can define each material type, namely elasticity, reflectance, and density. A key observation made during the data collection stage was the distinct change in fabric texture in relation to every material type. For example, the presence of spandex in the fleece blends made the fabric much less fuzzy. For this reason, texture was the main feature extracted from the images. Three main methods were evaluated to extract the texture from the images: Local Binary Pattern, Gray level Co-Occurrence Matrix, and Gabor wavelet filters.

1) Local Binary Patterns

Local Binary Pattern (LBP) is a simple yet effective method to extract the texture from an image [12]. Like the name suggests it computes a local representation of texture features. To do this the images first need to be converted to grayscale, next a pixel is compared to the surrounding pixels. If the pixel next to it has a greater value than the original pixel then the value is set to zero but if the original pixel has a greater intensity, then the value is set to one. This is done for each surrounding pixel. This method has two key parameters, the radius, and the number of neighboring points. If the radius is set to one and the number of neighboring points is set to 8 then one point is compared to the 8 points surrounding it. This means for these parameters the LBP for the pixel is an 8-bit value that can be converted to decimal. The amount of pixel intensity variation can help us to infer the textural properties of the image, for example, it can help us find edges or corners. An example of the LBP is shown in fig. 4, it is compared here to the original image. To utilize these LBP codes, we can create a histogram of the different values.

At first, this method was unsuccessful in helping to classify the fabric images. However, instead of creating a histogram of the entire image the image was split into 16 sub-images and a histogram was created of the LBP codes of each sub-image. Utilizing only the LBP patterns we were able to classify the images with a testing accuracy of as high as 97% for some of the fabric groups. Table 2 shows the evaluation of three classifiers using only the sub-image histograms of the extracted LBP.

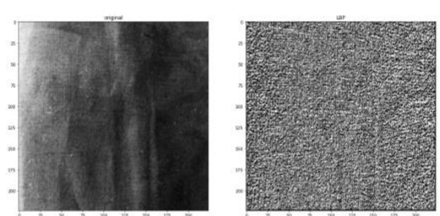

Fig. 4 Example of LBP (original image, LBP)

TABLE III. LBP FEATURE ANLAYSIS

Fabric Type	*SVM Test accuracy*	*Logistic Regression Test accuracy*	*KNN Test Accuracy*
Denim	92	93	96
Faux Leather	51	95	89

Fabric Type	SVM Test accuracy	Logistic Regression Test accuracy	KNN Test Accuracy
Fleece	53	85	80
Knit	100	100	100
Linen	48	82	81
Outdoor	95	99	96

An evaluation was also performed to determine the best number of points for the radius and the neighboring points. On average setting the radius to 1 and the number of neighboring points to 8 had the best results. Other combinations that were tested are listed in the table below. It should be noted the table is a very simplified version of the testing performed. In order to limit space, the individual fabrics and classifiers were not listed, however, in choosing the best parameters all classifiers and fabric groups were considered. While all the average accuracy scores are close to one another the parameters with the most consistency were chosen. In the latter combinations, there was great variation in the accuracy scores amongst fabric types.

TABLE IV. LBP PARAMETERS

Radius, Number of neighbors	Avg accuracy for all fabrics and classifiers
1, 8	85.3
2, 8	85
1, 16	79
2, 16	81

Additionally, the number of bins to include in each histogram was evaluated. Since a histogram is taken for 16 sub-images the number of features increased greatly with the number of bins used. However, using fewer bins also reduces the precision of the features. Various bin sizes were analyzed including 10, 30, 40, 64, and 100. However, a bin size smaller than 64 greatly reduced the accuracy of the model and a bin size much larger than 64 did not improve the accuracy of the model. In an effort to reduce the number of total features, a bin size of 64 was chosen.

2) Gray Level Co-occurrence Matrix

The Gray Level Co-occurrence Matrix (GLCM) is another popular method used to help extract the textural features of an image [13]. To compute the GLCM of an image the image must once again be converted to grayscale. Next, a spatial relationship is inputted which includes the reference pixel and the neighbor pixel. The spatial relationship is how the neighbor pixel is defined, how far it is from the reference pixel, and in what direction, for example, 2 above the reference pixel or 3 pixels diagonally from the reference. The GLCM contains the number of times each pair of intensities is seen in the image with the specified spatial relationship. Next, several properties can be calculated from the GLCM. These include the Mean, Variance, Correlation (1), Contrast (2), angular second moment (3), energy (4), and dissimilarity (5).

$$\sum_{i,j=0}^{levles-1} P_{i,j}[\frac{(i-\mu_i)(j-\mu_i)}{\sqrt{(\sigma_i^2)(\sigma_i^2)}} \quad (1)$$

$$\sum_{i,j=0}^{levles-1} P_{i,j}(i-j)^2 \quad (2)$$

$$\sum_{i,j=0}^{levles-1} P_{i,j}^2 \quad (3)$$

$$\sqrt{ASM} \quad (4)$$

$$\sum_{i,j=0}^{levles-1} P_{i,j}|i-j| \quad (5)$$

In the equations listed above P refers to the GLCM histogram where Pij is the number of times that the gray-level j occurs at the specified distance and angle from the gray-level i. Once again, the image was split into 16 sub-images, and the GLCM was calculated for each sub-image. Alone this method was not very useful in classifying the fabric content. However, combined with the LBP method the classifiers were able to classify the fabric contents with an accuracy of as much as 100% for some fabric groups. Table 4 lists the test accuracies for 3 classifiers using only the GLCM features.

TABLE V. GLCM FEATURE ANALYSIS

Fabric Type	SVM Test accuracy	Logistic Regression Test accuracy	KNN Test Accuracy
Denim	63	64	81
Faux Leather	48	70	57
Fleece	62	74	77
Knit	97	93	93
Linen	45	48	59
Outdoor	97	97	95

The two main parameters in GLCM are the distances and angles which refer to how we define the neighbor pixel. You can use multiple distances however this greatly increases the number of features. At this stage, it was obvious there were an excessive number of features, so I only considered using one distance and one angle. A distance of 2 and an angle set to 0 had the best results. Some of the other combinations of distances and angles that were tested are shown in Table 5.

TABLE VI. GLCM PARAMETERS

Distance, angle	Avg accuracy for all fabrics and classifiers
1, 0	73
2, 0	74

Distance, angle	Avg accuracy for all fabrics and classifiers
1, 20	73
2, 20	68

3) Saliency Maps and Gabor filter

Saliency Maps and Gabor filters were also evaluated as potential feature extraction methods. However, they were ultimately not used because they didn't provide any greater accuracy than LBP and GLCM. They actually were less successful than the two previously mentioned methods and since there were a great number of features already compared to the number of images these methods seemed unnecessary to add.

4) Dimensionality Reduction

In an effort to reduce overfitting, dimensionality reduction was done mainly with principal component analysis (PCA). Utilizing the sub-image method greatly increases the number of features. LBP utilizes a histogram with 64 bins for 16 sub-images which gives us 1024 features. GLCM has 6 properties for 16 sub-images which gives us an additional 96 features. In total, we have 1120 features. With this large number of features and the small amount of data samples in each group, there was a chance of overfitting.

PCA helps us to determine which features are highly correlated to one another and therefore the key features that should be kept and the ones that can be left out. For each fabric group, PCA was performed on the feature set, and only the features necessary to explain 92% of the variance were kept. This threshold was found through an iterative search. A value of less than 92% led to less accurate results and a value greater than 92% led to overfitting. The number of features kept for every fabric group is listed in Table 6.

TABLE VII. KEY FEATURES AFTER PCA

	Denim	Faux Leather	Fleece	Knit	Linen	Outdoor
Number Of Features	32	76	124	62	86	132

D. Classifier hyperparameter tuning:

Three main classifiers were evaluated: Support Vector Machines (SVM), Logistic Regression, and K Nearest Neighbors (KNN). The classifiers were evaluated using training accuracy, testing accuracy, F1 score, and the Area Under the Receiver Operating Characteristic Curve (AUC ROC). An accuracy score is based on the number of correct predictions compared to the total number of predictions; this can be misleading in the case of unbalanced data. This is the main reason the F1 score was included since many of the fabric groups are unbalanced. The F1 score is a combination of precision and recall as shown in (6).

$$F1\ score = 2 * \frac{Precision * Recall}{Precision + Recall} \qquad (6)$$

$$precision = \frac{\#\ of\ true\ positives}{\#\ of\ true\ positive + \#\ of\ flase\ positives} \qquad (7)$$

$$Recall = \frac{\#\ of\ True\ Positives}{\#of\ True\ Positives + \#\ of\ False\ Negatives} \qquad (8)$$

An ROC curve is a curve of probabilities that plots the False positive rate (FPR) against the True positive rate (TPR). The area under the ROC curve helps to determine how well the model is classifying the correct classes. A higher ROC AUC means the model is more accurately classifying the correct classes compared to a smaller ROC AUC. For multi-class problems, the ROC AUC was set to a one-versus-one scheme which means the average of the pairwise ROC AUC scores was computed. The parameters for each classifier were tuned to improve the performance metrics described above.

1) SVM

For the SVM classifier, a grid search was done to find the best parameters for each fabric group [20]. The parameter C refers to the regularization parameter. The value of C is inversely proportional to the amount of penalty that is applied. The penalty is the squared L2 penalty. C was evaluated from a value of 1 to 10. The kernel is the type of kernel that is used in the algorithm, the ones tested include linear, polynomial, radial basis function (RBF), and sigmoid. A linear kernel should be used when the data is linearly separable. The polynomial kernel represents utilizes polynomials of the original variables to represent the similarity of vectors in the data as shown in (9). The RBF kernel is a stationary kernel that is parameterized by a length scale parameter 'L' as shown in (10). The sigmoid kernel is the same as a two-layer perceptron model of a neural network, its equation is shown in (11). The best parameters found for each fabric group are shown in table 8.

$$k(x_i, y) = \tanh^d(\gamma x^T y + r)^d, \gamma > 0 \qquad (9)$$

$$k(x_i, y) = \exp\left(-\frac{d(x_i, y)^2}{2l^2}\right) \qquad (10)$$

$$k(x, y) = \tanh(\gamma x^T y + r) \qquad (11)$$

TABLE VIII. SVM PARAMETERS

	Denim	Faux Leather	Fleece	Knit	Linen	Outdoor
C	1	3	10	1	10	1
kernel	rbf	rbf	rbf	sigmoid	rbf	sigmoid

2) Logistic Regression

For the Logistic Regression function, the parameters that were evaluated were the solver and the penalty term [19]. The solvers that were evaluated were Limited-memory Broyden-Fletcher_Goldfarb_Shanno(L-BFGS), liblinear, newton-cg, and saga. The penalty terms that were evaluated depended on which solver was used, they included L1, L2, and no penalty term. Each solver has a different benefit, for example, liblinear is usually the best choice for small datasets while saga is much faster for larger data sets. L-BFGS approximates the second derivative matrix updates with gradient evaluations. The liblinear or Library for Large Linear Classification solver uses a coordinate descent algorithm. It is based on solving univariate optimization problems in a loop in order to minimize a multivariate function. The Newton-cg solver calculates the hessian explicitly; however, this can be computationally expensive especially in high-dimensions, therefore most often used for problems where there is a greater number of samples than features, unlike in this dataset. The saga solver is a variation of the Sag solver meaning it supports the L1 penalty option. It is usually the best choice for cases of sparse multinomial logistic regression. It should also be noted for multi-class problems newton-cg, saga, or lbfgs handle multinomial loss while liblinear will only use the one-versus-rest schemes. The best parameters for each fabric set are listed below.

TABLE IX. LOGISTIC REGRESSION PARAMETERS

	Denim	***Faux Leather***	***Fleece***	***Knit***	***Linen***	***Outdoor***
penalty	L2	L2	L2	L2	L2	L2
solver	lbfgs	liblinear	lbfgs	lbfgs	lbfgs	lbfgs

3) KNN

For the KNN classifier, the main parameter that was evaluated was the number of k neighbors [21]. A number of 1 to 4 neighbors were evaluated. For any number greater than 4 all of the training and testing accuracies went down. The best parameters for KNN for each fabric group are listed in the table below.

TABLE X. KNN PARAMETERS

	Denim	***Faux Leather***	***Fleece***	***Knit***	***Linen***	***Outdoor***
k	3	2	1	1	3	1

E. Evaluation

The data was first split into training and testing sets utilizing stratification since many of the classes have uneven amounts of data. For each fabric group, 20% of the data was left for testing. After tunning all of the parameters listed above, I found the best model for each fabric group. The best model and their performance metrics are listed below.

TABLE XI. FINAL RESULTS

	Denim	***Faux Leather***	***Fleece***	***Knit***	***Linen***	***Outdoor***
model	LR	KNN	KNN	KNN	SVM	LR
Test acc	100%	97%	94%	100%	100%	98%
F1	[1. 1.]	[1. 0.95 0.96]	[0.94, 0.97, 0.84]	[1. 1.]	[1. 1.]	[0.98 0.94]
Roc auc	1	1	0.93	1	1	1

KNN proved to have the most accurate results for the majority of the fabric groups however for each of the classifiers the performance metrics were very close to one another.

1) Multi-class

I was also interested to see whether putting all the classes together in one classification problem would achieve similar performance. I utilized the same image processing and feature extraction techniques however now there were 15 classes. I got an accuracy score in the test using KNN of 86%. The confusion matrix of the predicted labels and true labels for all the classes is below.

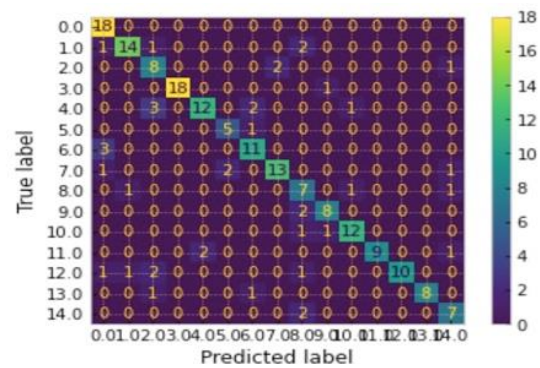

Fig. 5 Confusion Matrix for all classes

2) Comparing to CNN

Since many of the methods that have been used utilized Convolutional Neural Networks to classify images, I wanted to compare the proposed method to a CNN that utilized pre-trained weights, like had been done in other papers. I tried multiple CNN's including Resnet 18, Resnet 50, and Mobile net V2. The pretrained weights for each of these models used the ImageNet data set. I added a layer on top of all of these CNNs to do the classification. The highest accuracy score I received was around 80% in test utilizing the Mobile net V2 with a . I evaluated different parameters to tune including the learning rate, loss function, and number of epochs. I believe that with larger data sets I could have improved this model however seeing that a simpler method achieved high-accuracy results I chose to spend more time evaluating the other method.

3) Comparing to clustering methods

Admittedly this method requires a tedious labeling and data collection process. For this reason, I wanted to see how clustering methods would compare. Many clustering methods have been utilized in the weave-type research. Utilizing the same features, LBP and GLCM, I used KMeans and GMM to compare my results. To evaluate these methods, I used the rand index.

TABLE XII. CLUSTERING RESULTS

	Denim	*Faux Leather*	*Fleece*	*Knit*	*Linen*	*Outdoor*
KMeans	0.5	0.47	0.58	0.57	0.61	0.50
GMM	0.47	0.44	0.48	0.49	0.36	0.50

While the rand index scores are not high, with a very simple evaluation these results make me believe that with future analysis, higher level feature extraction, more model tuning, and more data in an unsupervised method may be able to determine the content of each fabric type.

V. SUMMARY AND CONCLUSIONS

Pattern recognition and computer vision techniques are important tools in textile analysis. Fabric composition is one of the main identifying factors of any textile. Textural properties such as the LBP and GLCM as seen in the proposed method were able to detect fabric composition in various fabric groups. The proposed method utilizes preprocessing techniques including adaptive histogram equalization and data augmentation. It uses LBP and GLCM to extract features from each image and it then uses PCA for dimensionality reduction. The final step is to use a KNN classifier to determine the fabric content. Utilizing this method, accuracy scores between 94-100% were achieved. Admittedly, performance decreased when all the classes were put together. In a future model, there may have to be multiple layers: one that detects the fabric group and a second layer that can detect the fabric content. Another hypothesis I had was that combing other work like microgeometry and reflectance along with the textural properties could help to improve the accuracy. It is also noted that the data collection process is extremely tedious making this method difficult to scale. One of the main limitations of this project was the available data. With more data and more fabric groups, this method could be refined further.

A. Limitations and future work

As mentioned previously, some preliminary evaluations were done using clustering methods and convolutional neural networks. In future work, a clustering method might be a better tool for larger amounts of data with more complex fabric types. One may be able to gather enough data of each material type such that you can extract features of each material. Then comparison of each fabric sample to the dictionary of set material features can be performed. yThis would allow for the classification of many materials present in the fabric.

REFERENCES

[1] X. Zhang, W. Gao and J. Liu, "Automatic Recognition of Yarn Count in Fabric Based on Digital Image Processing," *2008 Congress on Image and Signal Processing*, Sanya, China, 2008, pp. 100-103, doi: 10.1109/CISP.2008.435.

[2] Iqbal Hussain, Muhammad Ather, Babar Khan, Zhijie Wang, and Shenyi Ding. 2020. "Woven Fabric Pattern Recognition and Classification Based on Deep Convolutional Neural Networks" *Electronics* 9, no. 6: 1048. https://doi.org/10.3390/electronics9061048K. Elissa, "Title of paper if known," unpublished.

[3] Tsai, I-Shou, Chung-Hua Lin, and Jeng-Jong Lin. "Applying an artificial neural network to pattern recognition in fabric defects." *Textile Research Journal* 65, no. 3 (1995): 123-130.

[4] Zhang, Jie, Binjie Xin, and Xiangji Wu. "A review of fabric identification based on image analysis technology." *Textiles and Light Industrial Science and Technology* 2, no. 3 (2013): 120-130.

[5] Sun, Xudong, Mingxing Zhou, and Yize Sun. "Classification of textile fabrics by use of spectroscopy-based pattern recognition methods." *Spectroscopy Letters* 49, no. 2 (2016): 96-102.

[6] Kampouris, Christos, et al. "Fine-grained material classification using micro-geometry and reflectance." Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part V 14. Springer International Publishing, 2016.

[7] Rasheed, Aqsa, et al. "Fabric defect detection using computer vision techniques: a comprehensive review." *Mathematical Problems in Engineering* 2020 (2020): 1-24.

[8] Zhang, Wuyi, et al. "MWGR: A new method for real-time detection of cord fabric defects." *The 2012 International Conference on Advanced Mechatronic Systems*. IEEE, 2012.

[9] Li, Min, et al. "Fabric defect detection based on saliency histogram features." *Computational Intelligence* 35.3 (2019): 517-534.

[10] Karlekar, Vaibhav V., M. S. Biradar, and K. B. Bhangale. "Fabric defect detection using wavelet filter." *2015 International Conference on Computing Communication Control and Automation*. IEEE, 2015.

[11] Yapi, Daniel, Mohand Saïd Allili, and Nadia Baaziz. "Automatic fabric defect detection using learning-based local textural distributions in the contourlet domain." *IEEE Transactions on Automation Science and Engineering* 15.3 (2017): 1014-1026.

[12] Xie, Xianghua. "A review of recent advances in surface defect detection using texture analysis techniques." *ELCVIA: electronic letters on computer vision and image analysis* (2008): 1-22.

[13] Dollár, Piotr, et al. "Feature mining for image classification." *2007 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2007.

[14] Zhang, R., & Xin, B. (2016). A review of woven fabric pattern recognition based on image processing technology. *Research Journal of Textile and Apparel, 20*(1), 37-47. doi:https://doi.org/10.1108/RJTA-08-2015-0022

[15] Wang, Xin, Nicolas D. Georganas, and Emil M. Petriu. "Automatic woven fabric structure identification by using principal component analysis and fuzzy clustering." *2010 IEEE Instrumentation & measurement technology conference proceedings*. IEEE, 2010.

[16] Zhang, Rui, and Binjie Xin. "A review of woven fabric pattern recognition based on image processing technology." *Research Journal of Textile and Apparel* 20.1 (2016): 37-47.

[17] Iqbal Hussain, Muhammad Ather, et al. "Woven fabric pattern recognition and classification based on deep convolutional neural networks." *Electronics* 9.6 (2020): 1048.

[18] "Tf.keras.preprocessing.image.imagedatagenerator : Tensorflow V2.12.0." *TensorFlow*, https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator.

[19] "Sklearn.linear_model.Logisticregression." *Scikit*, https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html.

[20] "Sklearn.linear_model.Logisticregression." *Scikit*, https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html.

[21] "Sklearn.neighbors.kneighborsclassifier." *Scikit*, https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html.