Uluslararası İleri Doğa Bilimleri ve Mühendislik Araştırmaları Dergisi
Sayı 8, S. 263-272, 6, 2024

Araştırma Makalesi

IJANSER

International Journal of Advanced Natural Sciences and Engineering Researches
Volume 8, pp. 263-272, 6, 2024

Research Article

https://as-proceeding.com/index.php/ijanser
ISSN: 2980-0811

Kernel Matrix Based Textile Image Categorization Using Machine Learning

Rıfat Aşlıyan[*,1], Ömer Kalfa[2]

[1]Department of Mathematics, Faculty of Sciences, Aydın Adnan Menderes University, Türkiye
[2]The Graduate School of Natural and Applied Sciences, Aydın Adnan Menderes University, Türkiye

[*]rasliyan@adu.edu.tr Email of the corresponding author

(Received: 20 July 2024, Accepted: 24 July 2024)

(4th International Conference on Scientific and Academic Research ICSAR 2024, July 19 - 20, 2024)

ATIF/REFERENCE: Aşlıyan, R. & Kalfa, Ö. (2024). Kernel Matrix Based Textile Image Categorization Using Machine Learning. *International Journal of Advanced Natural Sciences and Engineering Researches*, 8(6), 263-272.

Abstract – Today, just as text and audio data are rapidly increasing, visual data is also growing at a fast pace. In textile technology, fabric patterns, and images have encountered a similar expansion. Consequently, resolving this large volume of data becomes a significant problem. We need efficient classification to quickly access this data, and this categorization process should be automated using machine learning techniques. In our study, we have constructed the systems for classifying textile images using machine learning methods such as Multilayer Perceptron, Support Vector Machines, and K-Nearest Neighbors. The textile dataset consists of color and grayscale images, divided into training and test sets. Models are trained using the training data, and their decision-making performance is evaluated using the test data. During model generation, preprocessing is performed first. All images are converted to black and white. Edge detection filters like Sobel and Prewitt are applied to find the edges in the images. Optionally, thinning can also be applied before this step. After preprocessing, feature extraction is carried out. For each image, the frequency of matrices called kernel matrices, which slide over the image, is calculated and normalized. This representation allows images to be transformed into vectors, which are then used to train machine learning models. In the testing phase, commonly used metrics such as F-score and Accuracy are employed to evaluate the performance of these systems. The developed models are compared to each other, and the most successful methods are determined.

Keywords – Textile Images, Support Vector Machines, Multi-Layer Perceptron, K-Nearest Neighbor, Kernel Matrix.

I. INTRODUCTION

In both our daily lives and on the internet, we encounter a vast number of images. While the number of images around us continues to grow rapidly, the human workforce available to process them remains limited. The task of categorizing, classifying, and listing images by their types incurs high costs and time loss. Moreover, due to human nature, this process often involves significant errors. Additionally, accessing a specific image from an already complex database can be cumbersome. This is where an automation system that accurately identifies, classifies, and compiles images with minimal errors becomes essential. Such an automation system would reduce costs and save time.

In this study, we focus on the classification of textile patterns (Aşlıyan 2002, Aşlıyan and Alpkoçak, 2002; Aşlıyan, 2010; Ulvi et al., 2013). Textile pattern classification is a subset of image classification.

Image classification can be based on either the content of the images or the textual descriptions associated with them. However, the most common approach is content-based image classification (Geyers et al., 2000). Textile pattern classification works as follows: By analyzing the features within a textile image, we determine its specific pattern type. For example, a floral image is assigned to the floral category, while a polka-dotted image is assigned to the polka-dotted category. The ultimate goal of textile pattern classification is to ensure that all textile patterns are accurately categorized, benefiting both customers and designers by allowing them to quickly find the desired textile products.

When classifying textile patterns, the most prevalent and up-to-date classification methods in image classification are Support Vector Machines (SVM) (Burges, 1998; Sezer et al., 2005), K-Nearest Neighbors (K-NN) (Akkuş and Güvenir, 1996; Özkan, 2013), and Multilayer Perceptron Artificial Neural Networks (MLP-ANN) (Haykin, 1994; Efe and Kaynak, 2000; Öztemel, 2006).

In the evaluation of this study, two datasets were created for training and testing purposes. For both the training and test datasets, images containing textile patterns for classes such as ‘Flowery,’ ‘Polka Dot,’ ‘Horizontal Striped,’ ‘Vertical Striped,’ ‘Plaid,’ ‘45-Degree Striped,’ and ‘135-Degree Striped’ were obtained. The training dataset was used to train the classification system, while the test dataset was created to evaluate how well the trained systems performed in classification and for system comparison. In image classification (Yılmaz, 2013), the number of images provided to the system plays a crucial role. Having too many images can make learning difficult, while having too few can increase the system’s error rate. The general structure of textile pattern classification is shown in Figure 1. Initially, textile images undergo preprocessing (Gonzalez and Woods, 2008; Timur and Sarı, 2010). During preprocessing, color or grayscale images are converted to black and white. Next, edge detection filters such as Sobel (Sobel and Feldman, 1968; Boyle and Thomas, 1988), Prewitt (Prewitt, 1970; Konishi et al., 2003), LoG (Özkul, 1995; Bilgi, 2012), and Zero-Cross (Clark, 1989) are applied to detect the edges of textile patterns. Additionally, skeletonization (Blum, 1967) is performed to thin the image. Finally, frequency calculations and normalization using 2x2, 3x3, and 4x4 kernel matrices are used to create image feature vectors.

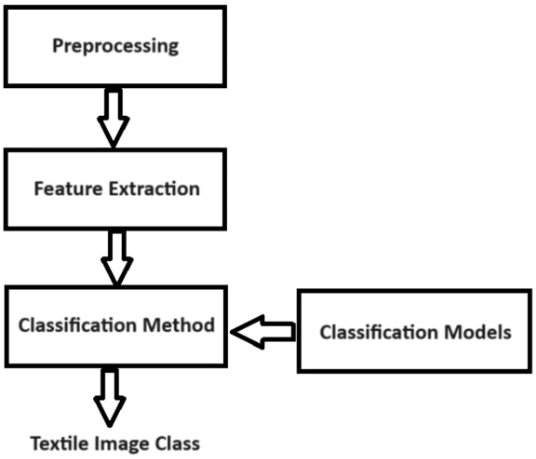

Fig. 1 General structure of textile categorization

As shown in Fig. 1, after calculating the kernel matrix frequencies of images in the training dataset, feature vectors are created for each class. Subsequently, unidentified textile images are classified using classification methods to determine their class. Over time, numerous projects and studies have been conducted in the field of image classification. Generally, these projects are related to computer vision technology. The systems developed as a result of image classification studies have been used in various domains, including health (such as cancer detection), military (such as weapon location detection), and automotive (to assist drivers). Our work will benefit both textile producers and consumers. Through this study, textile engineers and designers will have easier access to the patterns they need, work more efficiently, and create databases by performing faster classification. Consumers will also find it much easier to access products with their desired patterns.

The remaining sections of this study are organized as follows. In the second section, we explain the classification methods used along with digital image processing and edge extraction techniques for objects in images. The third section presents accuracy and F-Measure values resulting from system implementation, displayed in tabular form, and identifies the most successful methods. Finally, the last section compares the methods used based on overall test results.

II. MATERIALS AND METHOD

A. *Multi-Layer Perceptron (MLP)*

One of the classification models that has achieved significant success in classifying large and numerous data sets is the Multilayer Perceptron (MLP). Multilayer Perceptrons are developed from Single-Layer Perceptrons. The shortcomings of linear classification by single-layer perceptrons are addressed in Multilayer Perceptrons, allowing for accurate classification of non-linear data groups.

The working principle of Multilayer Perceptrons is as follows (Öztemel, 2012). Training the Network: Like all Artificial Neural Networks (ANN), the network needs to be trained. Since ANN learns from examples, examples are selected and shown to the network. The network will train itself with these examples.

Processing Input Information: Information taken from the input layer is processed with the learning and momentum coefficients, activation functions, weights, threshold values, and process elements of the Multilayer Perceptron.

Delta Learning Rule: Multilayer Perceptrons use the “Delta Learning” rule, also known as “supervised learning.” This rule is based on minimizing the difference between the expected output and the actual output. Both inputs and outputs are given to the system. The system finds the error and tries to reduce it to an acceptable level. The network continuously adjusts the weights to minimize the error. When the difference between the expected output and the actual output reaches an acceptable level, the network is considered trained.

Random Initial Weights: At the start of training, weights are assigned random values. The network reaches the optimal value during training.

Testing Performance: After training, the network’s performance must be tested. Known examples without outputs are shown to the network. The network classifies based on new input information. There is no fixed standard for acceptable performance; it is up to the user. The user decides if the network works as desired based on error tolerance.

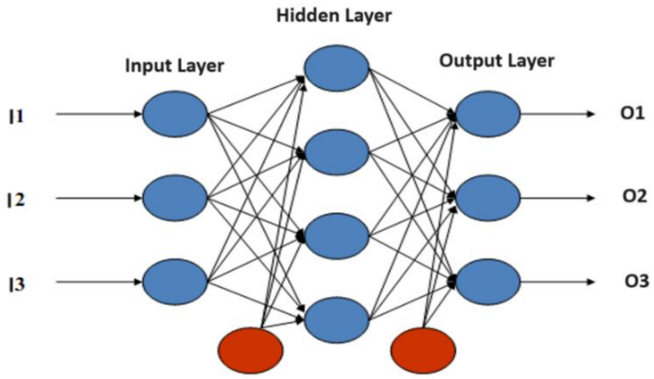

Fig. 2 Multi Layer Perceptron Model (MLP)

B. K-Nearest Neighbor (K-NN)

The K-Nearest Neighbors (K-NN) algorithm, widely used in classification methods, helps determine the classes of new items by using observation values from a sample set with known classes. In this method, the distances of each known class example to a newly determined, class-unknown observation are calculated. Then, the *k* closest examples form a group with the new item. The predominant class in this group is assigned as the class of the new observation (Özkan, 2013).

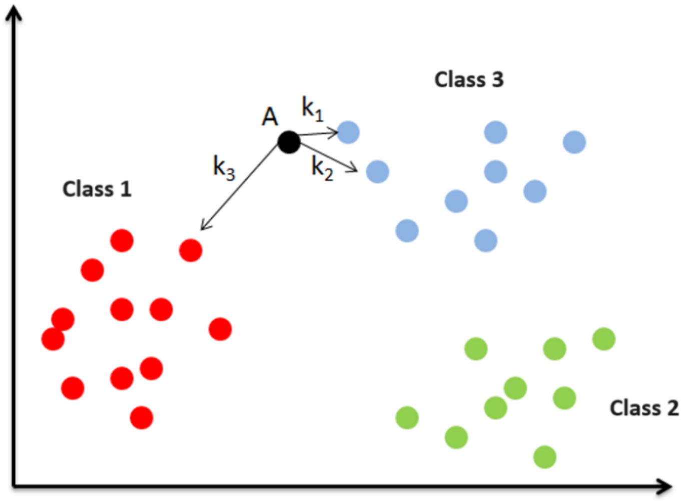

Fig. 3 Detection of point A for k=3

There are various formulas for calculating distances. However, the Euclidean distance formula is the most commonly used. In an N-dimensional set in Euclidean space, the Euclidean distance between points $\boldsymbol{P} = (p_1, p_2, p_3, \dots, p_n)$ and $\boldsymbol{Q} = (q_1, q_2, q_3, \dots, q_n)$.

$$\sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} \tag{1}$$

To briefly summarize the method's application, first, the *k* parameter must be determined. This *k* parameter is the number of nearest examples to the new observation whose class is to be found. In this case, the distances of all examples are calculated. Then, the weighted class of the set consisting of the selected nearest *k* points and the new item is found. This class is assigned as the class of our new observation.

Two methods are used to determine the weighted class. The first is to select the most frequently repeated class as the weighted class. The second method is the weighted voting method.

$$d'(\boldsymbol{P}, \boldsymbol{Q}) = \frac{1}{[d(\boldsymbol{P}, \boldsymbol{Q})]^2} \tag{2}$$

In the weighted voting method, a transformation of each example's Euclidean distances to the new observation is obtained using Equation 2. Then, the sums of these transformations for each class are taken to reach the weighted voting values of the classes in the set. The class with the highest value is assigned as the weighted class. However, the weighted voting method may yield more accurate results when used for the entire sample.

C. Support Vector Machines (SVM)

Support Vector Machines (SVM) are classification models based on statistical learning theory and the principle of minimizing structural risk (Ateş, 2014). In supervised learning, data with known classes are divided into two class labels. Considering the characteristics of known classes, the class of newly added data is determined. Classification with Support Vector Machines can be grouped into two categories: classification for linearly separable spaces and classification for non-linearly separable spaces.

In linear classification, the main goal is to select the optimal separating hyperplane among the hyperplanes that separate the two given classes in an n-dimensional space.

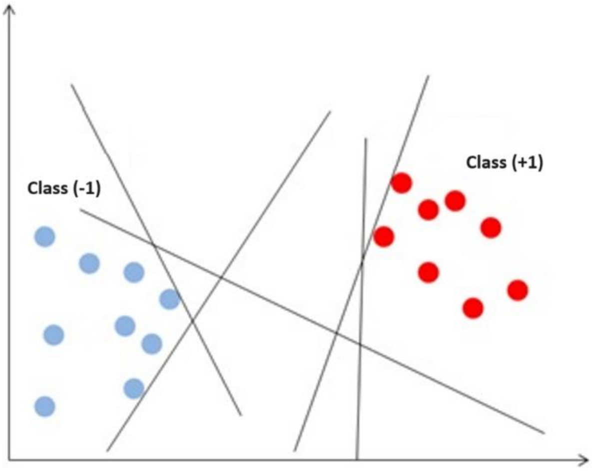

Fig. 4 İki sınıf için doğrusal ayrılabilen verilerin hiper düzlemleri

In this context, let S be a training set with n samples. For all $x_i \in S$, each sample has k attributes, with class labels $\{-1,+1\}$. The output $y_i \in \{-1,+1\}$ represents which class the samples belong to. Let $\boldsymbol{w} = \{w_1, w_2, w_3, \ldots, w_n\}$ be the normal of the hyperplane and also the weight vector, and let b be a constant. If we take each $\boldsymbol{x_i}$ sample as a k-dimensional vector, the equations of the hyperplanes are given in Equation 3.

$$H = <\boldsymbol{w}, \boldsymbol{x_i}> +b \quad (3)$$

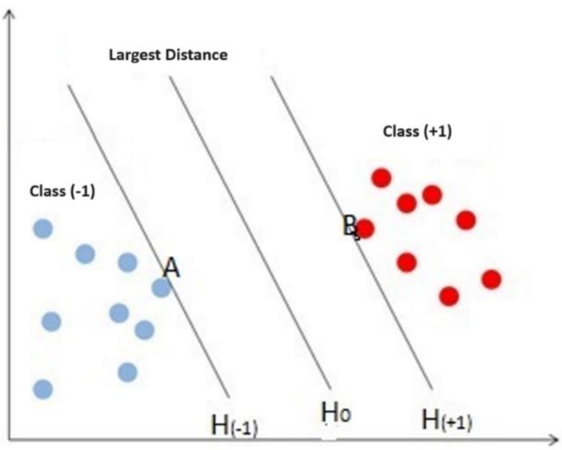

Fig. 5 İki sınıfı ayırmak için kullanılan hiper düzlemler

Optimal ayırma düzlemini bulmak için her iki sınıf arasındaki hiper düzlemler arasından birbirine en uzak iki hiper düzlem $H_{(-1)}$ ve $H_{(+1)}$ hiperdüzlemleri seçilir. Bu her iki hiperdüzlemin tam ortasından geçen H_0 hiper düzlemi ise aranan optimal ayırma hiper düzlemidir (Ayhan ve Erdoğmuş, 2014).

To find the optimal separating plane, the two hyperplanes $H_{(-1)}$ and $H_{(+1)}$, which are the farthest apart between the two classes, are selected. The hyperplane H_0 which passes exactly in the middle of these two hyperplanes, is the desired optimal separating hyperplane (Ayhan and Erdoğmuş, 2014).

III. RESULTS AND DISCUSSION

The first step in implementing the system is creating the dataset. The textile dataset consists of seven classes: “Flovery” “Polka-dotted,” “Plaid,” “Horizontal striped,” “Vertical striped,” “45-degree striped,” and “135-degree striped.” Each class contains 60 textile images. In total, our dataset comprises 420 images, amounting to 20 MB. Sample textile images for each class are provided in Appendix 1. These images are in color or grayscale and are in JPEG format.

Recall, Precision, Accuracy, F-Measure, and K-Fold Cross Validation Criteria are well-known metrics used to evaluate the system and are employed in this thesis. We can briefly explain what these terms mean (Aslıyan, 2002):

Precision: The ratio of correctly classified images to the total number of images in the same class.

Recall: The ratio of correctly classified images to the total number of images assigned to that class by the system.

F-Measure: A metric that combines precision and recall into a single measure to evaluate performance.

Accuracy: The ratio of correctly classified images to the total number of images across all classes.

The system’s results can be shown with an Error Matrix as in Table 1.

Table 1. Error Matrix

	“Pozitive” Class	“Negative” Class
Test Result “Pozitive”	TP	FP
Test Result “Negative”	FN	TN

In the error matrix in Table 1, two classes are defined as “Positive” and “Negative.” TP represents the number of images in the positive class that the system classified as positive; FP represents the number of negative images that the system classified as positive; FN represents the number of positive images that the system classified as negative; and TN represents the number of negative images that the system classified as negative. As can be understood, correct classifications by the system are represented by the letter T, and incorrect classifications by the letter F. In this case, Equation 4 gives Precision, Equation 5 gives Recall, and the Accuracy and F-Measure values for evaluating the system can be obtained from the equations in Equations 6 and 7, respectively.

$$Precision = \frac{TP}{TP+FP} \tag{4}$$

$$Recall = \frac{TP}{TP+FN} \tag{5}$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{6}$$

$$F - \text{Measure} = \frac{2 \times Precision \times Recall}{Precision+Recall} \tag{7}$$

The system was evaluated using the K-Fold Cross Validation criterion, as shown in Fig. 5. This method divides the total dataset into k folds and helps us understand the system’s performance by changing the training and test samples in each iteration. Each iteration has its own result. The average of the results obtained after all experiments gives the method’s success. In the K-Fold Cross Validation criterion, each image is used in both the training and test phases. This minimizes errors caused by unknown data.

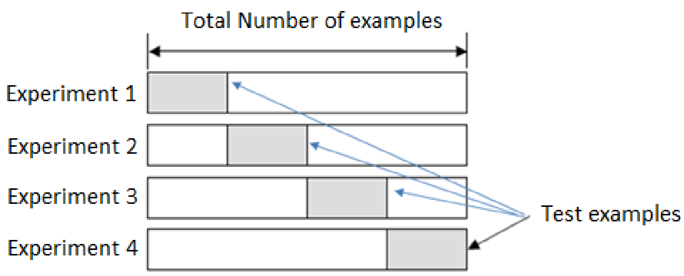

Fig. 6 K-Fold Cross Validation model

The applications in this study were performed on an Intel Core i7-2600 3.4 GHz processor, 16 GB RAM, and Windows 7 64-bit operating system. Codes were written using MATLAB R2012a to preprocess images and calculate frequencies, creating dataset files in arff format. The arff format datasets were taken and classified using WEKA 3.7 with training and testing processes. The 10-Fold Cross-Validation method was applied for this. In the WEKA program, K=1, 3, and 5 for the K-NN method; OldFormat=True and Discretization=True for the Naive Bayes method; Number of Neurons in Hidden Layers=17, Learning Rate=0.1, and Momentum Rate=0.8 for ANN; and BuildLogisticModel=True, FilterType=Standardize Training Data for SVM were selected.

As the size of the kernel matrices increases, the total number of features reaches very large numbers. Therefore, Information Gain was used as a feature dimension reduction method. Additionally, the average Accuracy and F-Measure results for different numbers of features were compared.

Table 2. The best results of the K-NN method according to Sobel, Prewitt, LoG, and Zero-Cross filters

Kernel Matrix: 2x2					
Filter	**Feature Number**	**K**	**Skeletonization**	**Accuracy**	**F-Measure**
Sobel	15	3	Yes	89.8%	89.6%
Prewitt	15	5	Yes	89.3%	89.3%
LoG	**15**	**1 ve 3**	**Yes**	**92.6%**	**92.6%**
Zero-Cross	**15**	**1 ve3**	**Yes**	**92.6%**	**92.6%**
Kernel Matrix: 3x3					
Filter	**Feature Number**	**K**	**Skeletonization**	**Accuracy**	**F-Measure**
Sobel	100	1	No	90.7%	90.6%
Prewitt	50	1	Yes / No	91.7%	91.6%
LoG	250	1	Yes	92.1%	91.9%
Zero-Cross	250	1	Yes	92.1%	91.9%
Kernel Matrix: 4x4					
Filter	**Feature Number**	**K**	**Skeletonization**	**Accuracy**	**F-Measure**
Sobel	250	1	Yes / No	92.4%	92.1%
Prewitt	250	1	No	92.3%	92.2%
LoG	500	1	No	91.4%	91.5%
Zero-Cross	250	1	No	91.7%	91.7%

Table 3. The best results of the PCA method according to Sobel, Prewitt, LoG, and Zero-Cross filters

Kernel Matrix: 2x2				
Filter	**Feature Number**	**Skeletonization**	**Accuracy**	**F-Measure**
Sobel	15	No	87.9%	87.8%
Prewitt	10	Yes	87.6%	87%
LoG	10	No	90.5%	90.4%
Zero-Cross	10	No	90.5%	90.4%
Kernel Matrix: 3x3				
Filter	**Feature Number**	**Skeletonization**	**Accuracy**	**F-Measure**
Sobel	50	No	91.2%	91%
Prewitt	50	No	89.8%	89.7%
LoG	50	No	93.6%	93.6%
Zero-Cross	50	No	93.6%	93.6%
Kernel Matrix: 4x4				
Filter	**Feature Number**	**Skeletonization**	**Accuracy**	**F-Measure**
Sobel	250	Yes	93.8%	93.7%
Prewitt	250	Yes	93.3%	93.3%
LoG	50	No	93.3%	93.3%
Zero-Cross	**250**	**No**	**94%**	**94%**

Table 4. The best results of the DVM method according to Sobel, Prewitt, LoG, and Zero-Cross filters

Kernel Matrix: 2x2				
Filter	**Feature Number**	**Skeletonization**	**Accuracy**	**F-Measure**
Sobel	15	No	88%	87.9%
Prewitt	15	Yes/ No	88.3%	88%
LoG	15	Yes	89.5%	89.2%
Zero-Cross	15	Yes	89.5%	89.2%
Kernel Matrix: 3x3				
Filter	**Feature Number**	**Skeletonization**	**Accuracy**	**F-Measure**
Sobel	50	No	93.6%	93.5%
Prewitt	50	No	92.9%	92.9%
LoG	250	Yes	94%	94%
Zero-Cross	50	No	94.4%	94.4%
Kernel Matrix: 4x4				
Filter	**Feature Number**	**Skeletonization**	**Accuracy**	**F-Measure**
Sobel	**500**	**No**	**95%**	**95%**
Prewitt	250	No	94.8%	94.8%
LoG	250	Yes	93.6%	93.6%
Zero-Cross	250	Yes	93.6%	93.6%

According to the data in Table 2, the best result for Accuracy and F-Measure with the K-NN method is 92.6%. This result was obtained using the LoG and Zero-Cross filters, with 15 features, K=1, skeletonization, and a 2x2 kernel matrix. According to the results in Table 3, the best Accuracy and F-Measure value with the PCA method is 94%. This result was obtained using the Zero-Cross filter, with 250 features, a 4x4 kernel matrix, and without skeletonization. As seen in Table 4, the best Accuracy and F-Measure results with the SVM method are 95%. This value was achieved using the Sobel filter, with 500 features, a 4x4 kernel matrix, and without skeletonization. When using a 2x2 kernel matrix, the best method is the K-NN method (Accuracy and F-Measure: 92.6%, Number of Features: 15, Filter: LoG and Zero-Cross, K:1 and 3, Skeletonization: Yes). The best result with a 3x3 kernel matrix was obtained using the SVM method (Accuracy and F-Measure: 94.4%, Number of Features: 50, Filter: Zero-Cross, Skeletonization: No). The SVM method (Accuracy and F-Measure: 0.950, Number of Features: 500, Filter: Sobel, Skeletonization: No) is also the best method with a 4x4 kernel matrix, as it was with the 3x3 matrix.

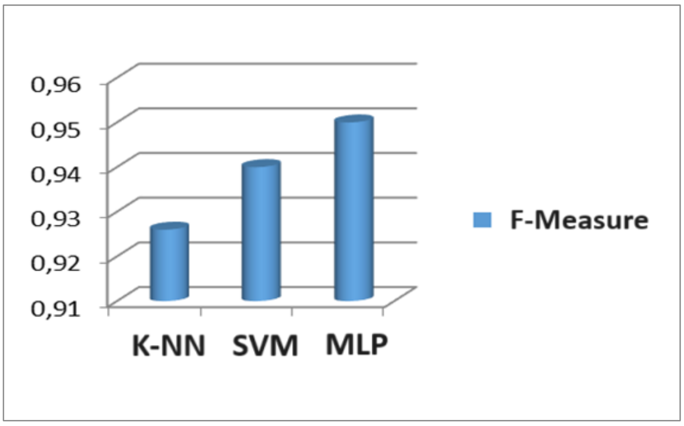

Fig. 7 The success of methods

As a result of the experimental study, the most successful core matrices are generally 4x4, 3x3 and 2x2 respectively. As seen in Fig. 7, the best results methods are SVM, MLP and K-NN.

IV. CONCLUSION

In this study, we have conducted the research to decide the classes of textile images such as “Flowery”, “Polka Dot”, “Plaid”, “Horizontal Striped”, “Vertical Striped”, “45-Degree Striped”, and “135-Degree Striped” using K-NN, MLP, and SVM methods. For this aim, the preprocessing involved edge detection using Sobel, Prewitt, LoG, and Zero-Cross filters. Subsequently, feature vectors were constructed by calculating and normalizing the frequencies of 2x2, 3x3, and 4x4 kernel matrices within the images.

The performance of the systems has been compared based on Accuracy and F-score using 10-fold cross-validation. The most successful methods were SVM, MLP, and K-NN, respectively. SVM, utilizing a 4x4 kernel matrix with the Sobel filter, achieved the highest accuracy with 95% and F-score. Among the kernel matrices, the order of success was 4x4, 3x3, and 2x2.

Information Gain has been employed to reduce the dimensionality of feature vectors, enhancing system performance while reducing execution time. Future work will include comparing different categorization techniques, using different edge detection filters, and developing systems with various kernel matrices.

ACKNOWLEDGMENT

The BAP project entitled “Güncel Metotlarla Resim Sınıflandırma” and with FEF 14030 number has been supported by Aydın Adnan Menderes University. The authors thank the BAP Agency.

REFERENCES

[1] R. Aşlıyan, “Classification of Textile Images”, Graduate School of Natural and Applied Sciences, Computer Engineering, Dokuz Eylül University, M.Sc. Thesis, İzmir, 2002.

[2] R. Aşlıyan, and A. Alpkoçak, Tekstil Desenlerinin Otomatik Olarak Sınıflandırılması Üzerine Bir Çalışma. SİU2002. 10. Sinyal İşleme ve İletişim Uygulamaları Kurultayı. Cilt I s. 123-128, Pamukkale, Denizli, 2002.

[3] R. Aşlıyan, “Textile Image Classification: Categorizing huge amout of textile images efficiently”. Lambert Academic Publishing AG& CO. KG. Saarbrücken, Germany. ISBN: 978-3-8383-5732-4. 2010.

[4] İ. Ulvi, R. Aşlıyan and K. Günel “Textile Image Classification Using Artificial Neural Networks”, 3rd World Conference on Innovation and Computer Science (INSODE - 2013), Antalya, Turkey, 2013.

[5] T. Geyers, F. Aldershoff and A. W. M. Smeulders, "Classification of Images on Internet by Visual and Textual Information", In SPIE Vol: 3964: doi: 10.1117/12.373453, San Jose, 2000.
[6] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition", Knowledge Discovery and Data Mining: 2(2) , 1998.
[7] O. G. Sezer, A. Erçil, M. Keskinöz, "Destek Vektör Makinesi Kullanarak Bağımsız Bileşen Tabanlı 3B Nesne Tanıma", *SUI 2005*, Sabancı Üniversitesi Mühendislik ve Doğa Bilimleri Fakültesi, 2005.
[8] A . Akkuş and H. A. Güvenir, "K-Nearest Neighbor Classification on Feature Projections". In Proc. *ICMI'96,* Lorenzo Saitta (Ed.), Morgan Kaufmann, Bari, Italy, pp. 12-19, 1996.
[9] Y. Özkan, *Veri Madenciliği Yöntemleri*, Dr. Rifat Çölkesen Dr. Cengiz Uğurkaya Papatya Yayıncılık, İstanbul, 2013.
[10] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Macmillan College Publishing Company, New York, 1994.
[11] M. Ö. Efe and O. *Kaynak, Artificial Neural Networks*, Boğaziçi University Publishing, İstanbul, 2000.
[12] E. Öztemel, *Yapay Sinir Ağları*, Papatya Yayıncılık Eğitim, Türkiye, 2012.
[13] R. C. Gonzalez, R. Woods, *Introduction. Digital Image Processing*, 3rd Edition, Prentice-Hall, New Jersey, 2008.
[14] E. Timur and C. Sarı, "Agora (Magnesia/Aydın) Manyetik Verilerinin Kenar Belirleme İşleçleri ve 3-Boyutlu Ters Çözümle Modellenmesi", Hacettepe Üniversitesi Yerbilimleri Uygulama ve Araştırma Merkezi Dergisi, vol. 31(2), pp. 67-82, 2010.
[15] I. Sobel and G. Feldman, *A 3x3 Isotropic Gradient Operator For Image Processing*, John Wiley and Sons, New York, US, 1968.
[16] R. Boyle and R. *Thomas, Computer Vision: A First Course,* Blackwell Scientific Publications, pp. 48-50, 1988.
[17] J. M. S. Prewitt, *Object Enhancement and Extraction, Picture Processing and Psychopictorics*, Editors Lipkin, B., Rosenfeld, A., Academic Press, New York, 1970.
[18] S. Konishi, A. L. Yuille, J. M. Coughlan, S. C. Zhu, "Statistical Edge Detection: Learning and Evaluating Edge Cues", *IEEE Transactions on, Pattern Analysis and Machine Intelligence*, vol. 25(1), pp. 57-74, 2003.
[19] S. M. Özkul, "Tek Kamera ile Görüntüde Derinliğin Hesaplanması", Osmangazi Üniversitesi Fen Bilimleri Enstitüsü Elektrik Elektronik Mühendisliği, Yüksek Lisans Tezi, Eskişehir, 1995.
[20] S. Bilgi, "Çok Ölçekli Kartografik Gösterimlerde Mekansal Bilginin Nicelik Analizi", İstanbul Teknik Üniversitesi Fen Bilimleri Enstitüsü Geomatik Mühendisliği Programı Gemomatik Mühendisliği Anabilimdalı, Doktora Tezi, sf. pp. 13-15, İstanbul, 2012.
[21] J. J. Clark, "Authenticating Edges Produced by Zero-Crossing Algorithms", *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol 2(1), pp. 43-57, 1989
[22] H. Blum, "A Transformation for Extracting New Descriptors of Shape, Models for the Perception of Speech and Visual Form", MIT Press, Cambridge, pp. 362–380, 1967.
[23] N. Ateş, "Destek Vektör Makineleri ve Gauss Karışım Modeli ile İstenmeyen E-postaların Tespiti", Süleyman Demirel Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı, Yüksek Lisans Tezi, Isparta, 2014.
[24] S. Ayhan and Ş. Erdoğmuş, "Destek Vektör Makineleriyle Sınıflandırma Probleminin Çözümü İçin Çekirdek Fonksiyonu Seçimi", Eskişehir Osmangazi Üniversitesi İİBF Dergisi, vol. 9(1), pp.175-198, 2014.